User's Manual

32

RH850/D1L/D1M Group

User's Manual: Hardware

Renesas microcontroller
RH850 Family

Renesas Electronics
www.renesas.com

Rev.2.20 Jan 2018

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Handling of Unused Pins

   Handle unused pins in accordance with the directions given under Handling of Unused Pins in the manual.

   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment when power is supplied.

   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.

   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

Readers This manual is intended for users who wish to understand the functions of the RH850/D1L/D1M and design application systems using the following RH850/D1L/D1M microcontrollers:

Purpose This manual is intended to give users an understanding of the hardware functions of the RH850/D1L/D1M shown in the *Organization* below.

Organization This manual is divided into two parts: Hardware (this manual) and Architecture (RH850 Family User's Manual: Software).

Hardware	Software
Pin functions	Overview
CPU function	Processor Model
On-chip peripheral functions	Register Reference
Flash memory programming	Exceptions and Interrupts
	Memory Management
	Instruction Reference
	Reset
	Appendix

How to read this manual It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

To understand the overall functions of the RH850/D1L/D1M.

→ Read this manual according to the Contents.

To understand the details of an instruction function

→ See RH850 G3M User's Manual: Software (R01US0123EJ) available separately.

Product Naming To indicate the dedicated product in the "RH850/D1L/D1M Group", each product name is described as below:

D1L1, D1L2, D1L2H, D1L2(H), D1Lx

D1M1, D1M1H, D1M1(H), D1M2, D1M2H, D1M2(H), D1Mx, D1M1A, D1M1-V2

Conventions Data significance: Higher digits on the left and lower digits on the right

Active low representation: $\overline{xxx}$ (overscore over pin or signal name)

Memory map address: Higher addresses on the top and lower addresses on the bottom

Note: Footnote for item marked with Note in the text

Caution: Information requiring particular attention

Remark: Supplementary information

Numeric representation: Binary ... xxxx or $xxxx_B$

Decimal ... xxxx

Hexadecimal ... $xxxx_H$

Prefix indicating power of 2 (address space, memory capacity):

K (kilo): $2^{10} = 1,024$

M (mega): $2^{20} = 1,024^2$

G (giga): $2^{30} = 1,024^3$

Description of Registers

Each register description includes register access, register address, and register value after a reset, a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meaning of the bit settings.

The standard format for bit charts and tables are described below.

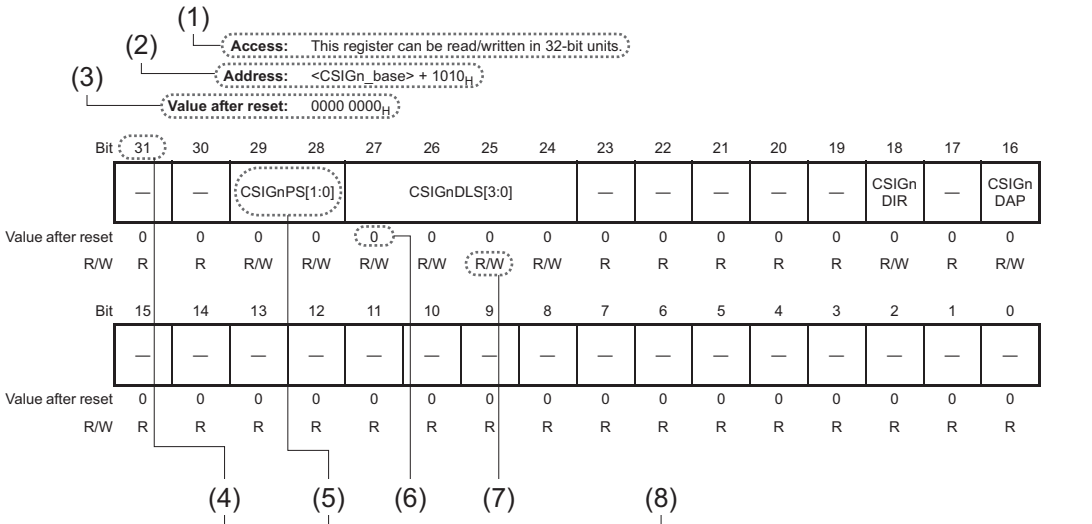


Table 14.19 CSIGnCFG0 register contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	The write value should always be the value after reset.
29, 28	CSIGnPS[1:0]	Specifies parity.
27 to 24	CSIGnDLS [3:0]	Specifies data length. 0: Data length is 16 bits 1: Data length is 1 bit 2: Data length is 2 bits ... 15: Data length is 15 bits **CAUTION** For a data length of less than 7 bits, do not set bits CSIGnCFG0.CSIGnDLS[3:0] for a value 1 to 6 when the extended data length function is disabled with bit CSIGnCTL1.CSIGnEDLE set to 0. It is forbidden to transmit two consecutive data with a data length of less than 7 bits.
23 to 19	Reserved	The write value should always be the value after reset.

CSIGn PS1	CSIGn PS0	Transmission	Reception
0	0	No parity transmitted	No parity is waited for.
0	1	Add parity bit fixed at 0	Parity bit is waited for but not judged.
1	0	Add odd parity	Odd parity bit is waited for.
1	1	Add even parity	Even parity bit is waited for.

(1) Access

The register can be accessed in the bit unit indicated here.

(2) Address

This is the register address.
For base address, see description of base address in each section.

(3) Value after a reset (in hexadecimal notation)

This is the value of all bits of the register after a reset. Values for bytes are given as numbers in the range from 0 to 9 and letters from A to F or as X where they are undefined.

(4) Bit position

This is the bit number.
The bits are numbered from 31 to 0 for 32-bit registers, 15 to 0 for 16-bit registers, and 7 to 0 for 8-bit registers.

(5) Bit name

Bit name or field name is indicated.
When clearly identifying the digits of a bit field is required, do so by using a form such as CSIGnDLS[3:0] above.
Indicate reserved bits by using a dash (—).

(6) Value after a reset (in binary notation)

This is the bit values after a reset.

0	:	The value after a reset is 0.
1	:	The value after a reset is 1.
—	:	The value after a reset is undefined.

(7) R/W

This is the bit attribute of all bits of the register.

R/W	:	The bit or field is readable and writable.
R	:	The bit or field is readable. Note that all reserved bits are indicated as R. When written, the value specified in the bit chart or the value after a reset should be written.
W	:	This bit or field is writable.When read, the value is undefined. If a value is indicated in the bit chart, the value is returned.

(8) Function

This is function of the bit.

Table of Contents

Section 1 Overview

1.1 Former products

CAUTION

For the new development project, please consider using "Replacement product" in the following table instead of "Former product".

Table 1.1 RH850/D1L/D1M former products

Former product		Replacement product	
Name	**Part number**	**Name**	**Part number**
D1M1_3.75M	R7F701404	D1M1-V2	R7F701442, R7F701462
D1M1_5M	R7F701405		
D1M1H_3.75M	R7F701406	D1M1A	R7F701441, R7F701461
D1M1H_5M	R7F701407		

1.2 RH850/D1L products overview

Table 1.2 RH850/D1L products overview (1/3)

Series name:			D1L1	D1L2	D1L2H
Memory	Code Flash		2 MB	4 MB	
	Local RAM (LRAM)		256 KB	512 KB	
	Retention RAM (RRAM)		16 KB		
	Data Flash		64 KB		
	Video RAM (VRAM) with Video RAM wrapper		–	144 KB	
External memory interfaces	Serial Flash Memory I/F (SFMA)	Bus width	4 bit	4 bit	8 bit
		Mode	SDR	SDR, DDR	SDR, DDR
		Max. clock	40 MHz	SDR: 120 MHz, DDR: 80 MHz	SDR: 120 MHz, DDR: 80 MHz
CPU	CPU System		G3M		
	CPU frequency		120 MHz		
	Floating Point Unit (FPU)		Provided		
	Memory Protection Unit (MPU)		Provided		
	Memory caches	Instruction cache	8 KB/4-way associative		
		Non-CPU system memories	–	16 KB/4-way associative	
DMA			16 channels		

Table 1.2 RH850/D1L products overview (2/3)

Series name:			D1L1	D1L2	D1L2H
Operating clock	Main Oscillator (MainOsc)		8 to 16 MHz		
	Low Speed Internal Oscillator (LS IntOsc)		typ. 240 kHz		
	High Speed Internal Oscillator (HS IntOsc)		typ. 8 MHz		
	Sub Oscillator (SubOsc)		typ. 32.768 kHz		
	Spread-spectrum PLL0		max. 480 MHz		
	PLL1		max. 480 MHz		
I/O port			103	103	126
A/D Converter (ADCE)			16 channels, 12 bit resolution		
Timer	Timer Array Unit B (TAUB)		3 units (16 bit resolution, 16 channels/unit)		
	Timer Array Unit J (TAUJ)		1 unit (32 bit resolution, 4 channels/unit)		
	Operating System Timer (OSTM)		2 units (32 bit resolution, 1 channel/unit)		
	Alway-On-Area Timer (AWOT)		1 unit (32 bit resolution, 1 channel/unit)		
	Real-Time Clock (RTCA)		Provided		
	Window Watchdog Timer A (WDTA)		2 units		
	PWM Generators with Diagnostic		1 unit (12 bit resolution, 24 PWM generators, 12 with diagnostic capability)		
Communi-cation interfaces	Clocked Serial Interface G (CSIG)		4 channels		
	Clocked Serial Interface H (CSIH)		2 channels		
	CAN Interface (RS-CAN)		3 channels (total 192 message buffers)*1		
	CAN Interface (RS-CANFD)		3 channels (total 192 message buffers)*1		
	LIN/UART Interface (RLIN3)		4 channels		
	I^2C Interface (RIIC)		2 channels		
External interrupts	Maskable		11		
	Non-maskable (NMI)		1		
Audio	Sound Generator (SG)		5 units		
	PCM-PWM Converter (PCMP)		1 unit		
	I^2S Interface (SSIF)		2 units (1 channel/unit)		
Video and Graphics	Video Output (VO)	Resolution	–	480 x 320 pixels	
		Color format	–	RGB666	
		Max. pixel clock	–	10 MHz	
		Layers	–	4	
		I/F	–	LVTTL	
		RLE decoding	–	Provided	
		Sprite layer	–	3 x 16 sprites for 3 output layers	
		Timing Controller (TCON)	–	3 programmable signals	7 programmable signals

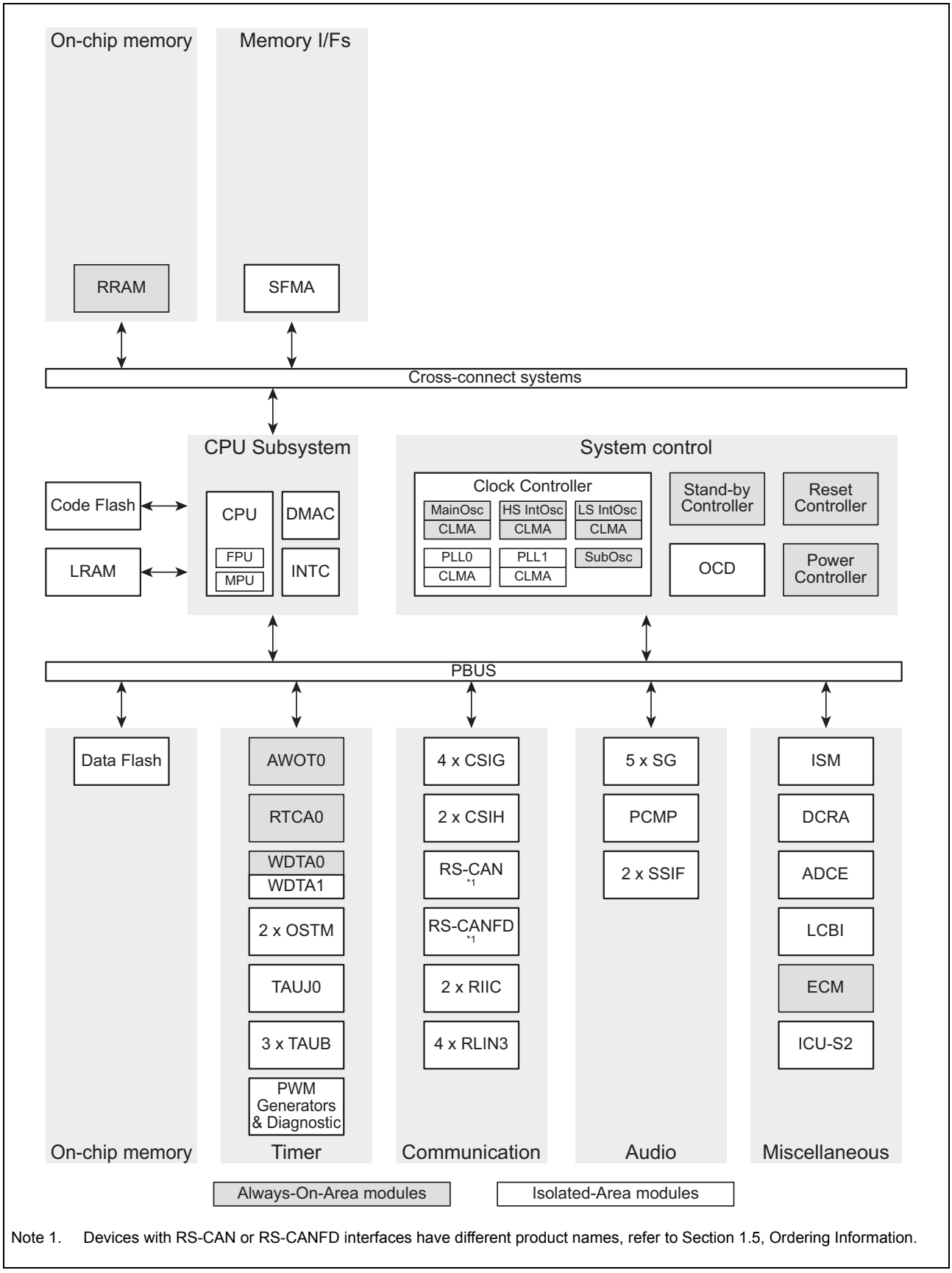
Table 1.2 RH850/D1L products overview (3/3)

	Series name:		D1L1	D1L2	D1L2H
Other functions	LCD Bus I/F (LCBI)		18 bit output, max. 10 MHz		
	Clock Monitors (CLMA)		for MainOsc, LS IntOsc, HS IntOsc, PLL0, PLL1		
	Data CRC (DCRA)		Provided		
	Power-On-Clear (POC)		Provided		
	Intelligent Stepper Motor Driver (ISM), incl. zero point detection for each channel		1 unit, 6 channels		
	Error Correction Coding (ECC)		for Code Flash, Data Flash, Local RAM, Retention RAM, Video RAM, RS-CAN RAM, Caches tag/data RAMs		
	Error Control Module (ECM)		Master/Checker		
	Intelligent Cryptographic Unit (ICU-S2)		Provided		
	On-Chip debug (OCD)		Provided		
	Boundary Scan		Provided		
Voltage supply*2	Internal logic	AWO*3	3.3 V, 5 V via on-chip voltage regulator		
		ISO*3	3.3 V, 5 V via on-chip voltage regulator		
	I/O buffers	GPIO*3	3.3 V, 5 V		
	A/D Converter supplies		nominal 3.3 V, 5 V		
Package	Type		QFP	QFP	QFP
	Pins		144	144	176
	pin/ball pitch		0.5 mm	0.5 mm	0.5 mm

Note 1. Devices with RS-CAN or RS-CANFD interfaces have different product names, refer to **Section 1.5, Ordering Information**.

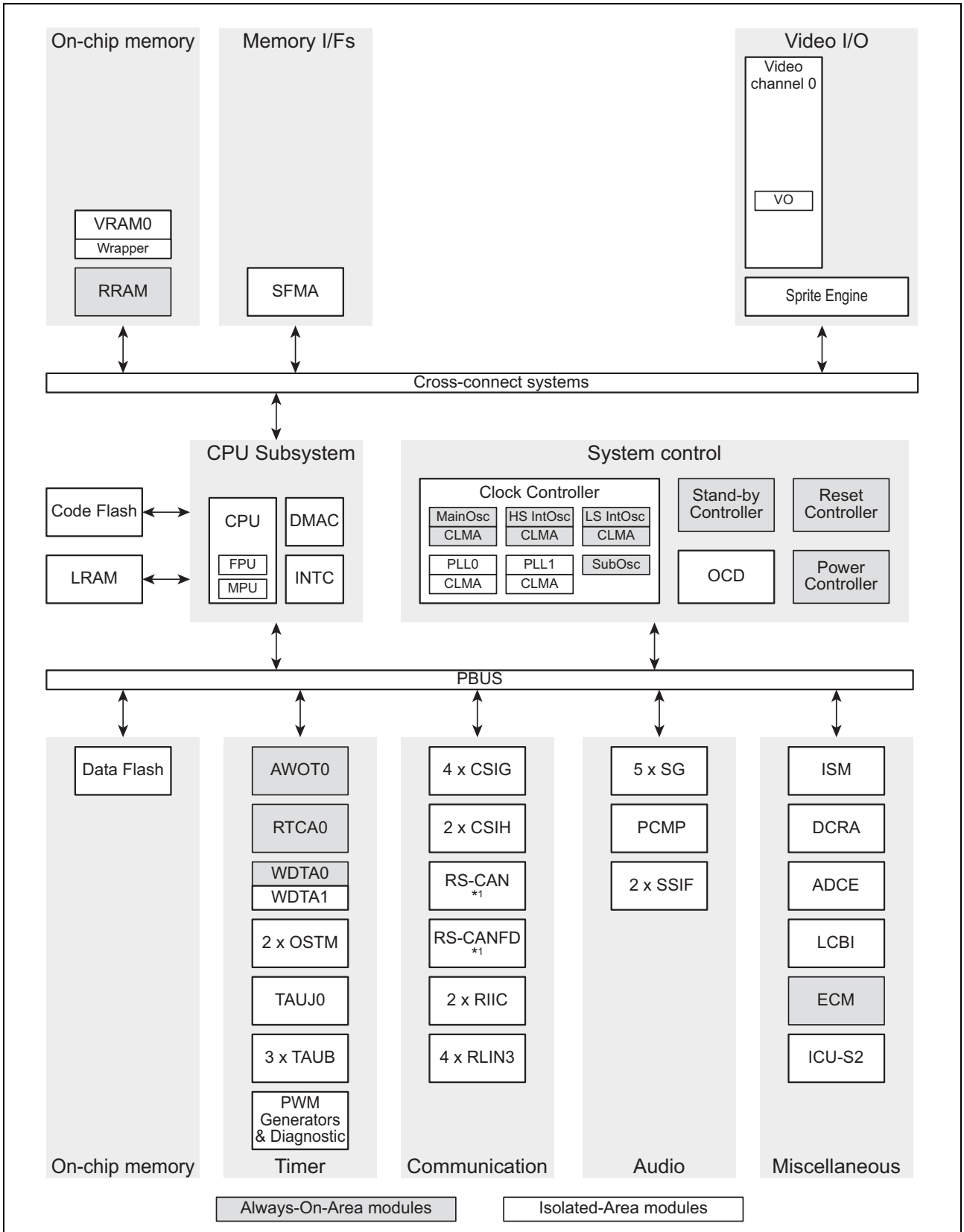
Note 2. The supply voltages are given as nominal values. Refer to the data sheet for detail specification of electrical values.

Note 3. AWO: Always-On-Area
ISO: Isolated-Area
GPIO: General purpose I/O port



Note 1. Devices with RS-CAN or RS-CANFD interfaces have different product names, refer to Section 1.5, Ordering Information.

Figure 1.1 D1L1 block diagram



Note 1. Devices with RS-CAN or RS-CANFD interfaces have different product names, refer to Section 1.5, Ordering Information.

Figure 1.2 D1L2(H) block diagram

1.3 RH850/D1M1 products overview

1.3.1 RH850/D1M1 former products overview

Table 1.3 RH850/D1M1 former products overview (1/3)

Series name:			D1M1_		D1M1H_	
			3.75M	5M	3.75M	5M
Memory	Code Flash		3.75 MB	5 MB	3.75 MB	5 MB
	Local RAM (LRAM)		512 KB			
	Retention RAM (RRAM)		16 KB			
	Data Flash		64 KB			
	Video RAM (VRAM) with Video RAM wrapper		1.55 MB			
External memory interfaces	SDRAM I/F	Bus width	–		32-bit	
		Mode	–		SDR-SDRAM (SDRA)	
		Max. clock	–		100 MHz	
	Serial Flash Memory I/F 0 (SFMA0)	Bus width	8 bit			
		Mode	SDR, DDR			
		Max. clock	SDR: 120 MHz, DDR: 80 MHz			
CPU	CPU System		G3M			
	CPU frequency		160 MHz		200 MHz	
	Floating Point Unit (FPU)		Provided			
	Memory Protection Unit (MPU)		Provided			
	Memory caches	Instruction cache	8 KB/4-way associative			
		Non-CPU system memories	16 KB/4-way associative			
DMA			16 channels			
Operating clock	Main Oscillator (MainOsc)		8 to 16 MHz			
	Low Speed Internal Oscillator (LS IntOsc)		typ. 240 kHz			
	High Speed Internal Oscillator (HS IntOsc)		typ. 8 MHz			
	Sub Oscillator (SubOsc)		typ. 32.768 kHz			
	Spread-spectrum PLL0		max. 480 MHz			
	PLL1		max. 480 MHz			
I/O port			126			
A/D Converter (ADCE)			16 channels, 12 bit resolution			
Timer	Timer Array Unit B (TAUB)		3 units (16 bit resolution, 16 channels/unit)			
	Timer Array Unit J (TAUJ)		1 unit (32 bit resolution, 4 channels/unit)			
	Operating System Timer (OSTM)		2 units (32 bit resolution, 1 channel/unit)			
	Alway-On-Area Timer (AWOT)		1 unit (32 bit resolution, 1 channel/unit)			
	Real-Time Clock (RTCA)		Provided			
	Window Watchdog Timer A (WDTA)		2 units			
	PWM Generators with Diagnostic		1 unit (12 bit resolution, 24 PWM generators,12 with diagnostic capability)			

Table 1.3 RH850/D1M1 former products overview (2/3)

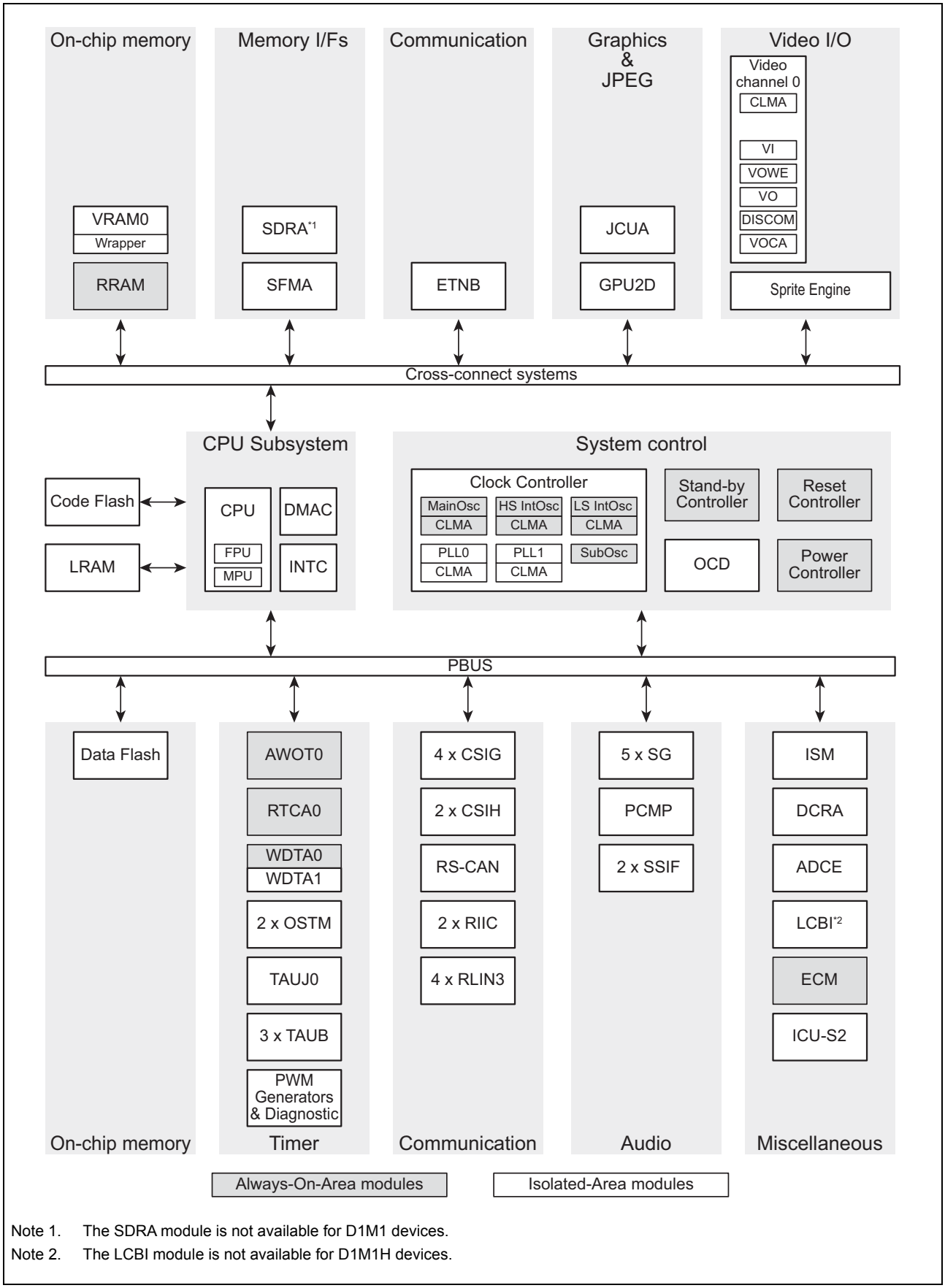
Series name:			D1M1_		D1M1H_	
			3.75M	5M	3.75M	5M
Communi-cation interfaces	Clocked Serial Interface G (CSIG)		4 channels			
	Clocked Serial Interface H (CSIH)		2 channels			
	CAN Interface (RS-CAN)		3 channels (total 192 message buffers)			
	CAN Interface (RS-CANFD)		–			
	LIN/UART Interface (RLIN3)		4 channels			
	I^2C Interface (RIIC)		2 channels			
	Ethernet AVB MAC (ETNB)		1 channel (Media Access Controller for up to 100 Mbps,with Audio Video Bridging)			
External interrupts	Maskable		11			
	Non-maskable (NMI)		1			
Audio	Sound Generator (SG)		5 units			
	PCM-PWM Converter (PCMP)		1 unit			
	I^2S Interface (SSIF)		2 units (1 channel/unit)			
Video and Graphics	Video Output 0 (VO0)	Resolution	1024 x 1024 pixels			
		Color format	RGB888			
		Max. pixel clock	30 MHz			
		Layers	4			
		I/F	LVTTL			
		Predistortion	Warping Engine (VOWE)			
		Timing Controller (TCON)	7 programmable signals			
		Video output data control	Video Output Checker (VOCA), 2 CRC checker (DISCOM)			
	Video Outputs shared features	RLE decoding	Provided for background layers			
		Sprite layer	3 x 16 sprites			
	Video Input (VI)	Channels	1 channel			
		Resolution	1024 x 1024 pixels			
		Pixel clock	30 MHz			
		Color formats	RGB666, ITU656			
		I/F	LVTTL			
	Graphics Processing Unit		2D Graphics Processing Unit (GPU2D), 80 MHz operation clock		2D Graphics Processing Unit (GPU2D), 100 MHz operation clock	
	JPEG Unit (JCUA)		Provided			

Table 1.3 RH850/D1M1 former products overview (3/3)

Series name:			D1M1_		D1M1H_	
			3.75M	5M	3.75M	5M
Other functions	LCD Bus I/F (LCBI)		18 bit output, max. 10 MHz		–	
	Clock Monitors (CLMA)		for MainOsc, LS IntOsc, HS IntOsc, PLL0, PLL1, Video Input pixel clocks			
	Data CRC (DCRA)		Provided			
	Power-On-Clear (POC)		Provided			
	Intelligent Stepper Motor Driver (ISM), incl. zero point detection for each channel		1 unit, 6 channels			
	Error Correction Coding (ECC)		for Code Flash, Data Flash, Local RAM, Retention RAM, Video RAM, RS-CAN RAM, Caches tag/data RAMs			
	Error Control Module (ECM)		Master/Checker			
	Intelligent Cryptographic Unit (ICU-S2)		Provided			
	On-Chip debug (OCD)		Provided			
	Boundary Scan		Provided			
Voltage supply*1	Internal logic	AWO*2	3.3 V, 5 V via on-chip voltage regulator			
		ISO*2	3.3 V via on-chip voltage regulator			
	I/O buffers	GPIO*2	3.3 V, 5 V			
		SDR-SDRAM	–		3.3 V	
	A/D Converter supplies		nominal 3.3 V, 5 V			
Package	Type		HLQFP		BGA	
	Pins		176		272	
	pin/ball pitch		0.5 mm		1.0 mm	

Note 1. The supply voltages are given as nominal values. Refer to the data sheet for detail specification of electrical values.

Note 2. AWO: Always-On-Area
ISO: Isolated-Area
GPIO: General purpose I/O port



Note 1. The SDRA module is not available for D1M1 devices.
Note 2. The LCBI module is not available for D1M1H devices.

Figure 1.3 D1M1(H) block diagram

1.3.2 RH850/D1M1 development products overview

Table 1.4 RH850/D1M1 development products overview (1/4)

Series name:			D1M1-V2	D1M1A
Memory	Code Flash		4 MB	
	Local RAM (LRAM)		512 KB	
	Retention RAM (RRAM)		16 KB	
	Data Flash		64 KB	
	Video RAM (VRAM) with Video RAM wrapper		1.55 MB	2 x 1.2 MB
External memory interfaces	SDRAM I/F	Bus width	–	32-bit
		Mode	–	SDR-SDRAM (SDRA)
		Max. clock	–	120 MHz
	Serial Flash Memory I/F 0 (SFMA0)*4	Bus width	8 bit	
		Mode	SDR, DDR	
		Max. clock	SDR: 120 MHz, DDR: 80 MHz	
	Serial Flash Memory I/F 1 (SFMA1)*4	Bus width	8 bit	
		Mode	SDR, DDR	
		Max. clock	SDR, DDR: 40 MHz	
	Serial Flash Memory I/F 2 (SFMA2)*4	Bus width	–	4 bit
		Mode	–	SDR, DDR
		Max. clock	–	SDR, DDR: 80 MHz
	HyperBus I/F (HYPB)*4	Bus width	8 bit	
		Mode	DDR	
		Max. clock	80 MHz	
	OCTA Flash I/F (OCTA)*4	Bus width	8 bit	
		Mode	DDR	
		Max. clock	80 MHz	
	NAND Flash I/F (NFMA)*4	Bus width	–	8 bit
		Mode	–	ONFi 1.0 (mode 0 and 1)
CPU	CPU System		G3M	
	CPU frequency		160 MHz	240 MHz
	Floating Point Unit (FPU)		Provided	
	Memory Protection Unit (MPU)		Provided	
	Memory caches	Instruction cache	8 KB/4-way associative	
		Non-CPU system memories	16 KB/4-way associative	
DMA			16 channels	

Table 1.4 RH850/D1M1 development products overview (2/4)

	Series name:	D1M1-V2	D1M1A
Operating clock	Main Oscillator (MainOsc)	8 to 16 MHz	
	Low Speed Internal Oscillator (LS IntOsc)	typ. 240 kHz	
	High Speed Internal Oscillator (HS IntOsc)	typ. 8 MHz	
	Sub Oscillator (SubOsc)	typ. 32.768 kHz	
	Spread-spectrum PLL0	max. 960 MHz	
	PLL1	max. 480 MHz	
I/O port		127	
A/D Converter (ADCE)		16 channels, 12 bit resolution	
Timer	Timer Array Unit B (TAUB)	3 units (16 bit resolution, 16 channels/unit)	
	Timer Array Unit J (TAUJ)	1 unit (32 bit resolution, 4 channels/unit)	
	Operating System Timer (OSTM)	2 units (32 bit resolution, 1 channel/unit)	
	Alway-On-Area Timer (AWOT)	1 unit (32 bit resolution, 1 channel/unit)	
	Real-Time Clock (RTCA)	Provided	
	Window Watchdog Timer A (WDTA)	2 units	
	PWM Generators with Diagnostic	1 unit (12 bit resolution, 24 PWM generators, 12 with diagnostic capability)	
Communi-cation interfaces	Clocked Serial Interface G (CSIG)	4 channels	
	Clocked Serial Interface H (CSIH)	2 channels	
	CAN Interface (RS-CAN)	3 channels (total 192 message buffers)*1	
	CAN Interface (RS-CANFD)	3 channels (total 192 message buffers)*1	
	LIN/UART Interface (RLIN3)	4 channels	
	I^2C Interface (RIIC)	2 channels	
	Ethernet AVB MAC (ETNB)	1 channel (Media Access Controller for up to 100 Mbps, with Audio Video Bridging)	
External interrupts	Maskable	11	
	Non-maskable (NMI)	1	
Audio	Sound Generator (SG)	5 units	
	PCM-PWM Converter (PCMP)	1 unit	
	I^2S Interface (SSIF)	2 units (1 channel/unit)	

Table 1.4 RH850/D1M1 development products overview (3/4)

			Series name: D1M1-V2	D1M1A
Video and Graphics	Video Output 0 (VO0)	Resolution	1024 x 1024 pixels	1280 x 1024 pixels
		Color format	RGB888	
		Max. pixel clock	30 MHz LVTTL, 40 MHz SerialRGB	48 MHz LVTTL, 34 MHz OpenLDI, 30 MHz VODDR
		Layers	4	
		I/F	LVTTL,SerialRGB	LVTTL, OpenLDI, VODDR *4
		Predistortion	Warping Engine (VOWE)	
		Timing Controller (TCON)	7 programmable signals	
		Video output data control	Video Output Checker (VOCA), 2 CRC checker (DISCOM)	
	Video Output 1 (VO1)	Resolution	–	1280 x 1024 pixels
		Color format	–	RGB888
		Max. pixel clock	–	40 MHz SerialRGB, 10 MHz VODDR
		Layers	–	4
		I/F	–	SerialRGB, VODDR
		Predistortion	–	–
		Timing Controller (TCON)	–	7 programmable signals
		Video output data control	–	Video Output Checker (VOCA) 2 CRC checker (DISCOM)
	Video Outputs shared features	RLE decoding	Provided for all layers	
		Sprite layer	4 x 16 sprites	
	Video Input (VI)	Channels	1 channel	
		Resolution	1024 x 1024 pixels	
		Pixel clock	30 MHz	
		Color formats	RGB666, ITU656	
		I/F	LVTTL	
	Graphics Processing Unit		2D Graphics Processing Unit (GPU2D), 80 MHz operation clock	2D Graphics Processing Unit (GPU2D), 120 MHz operation clock
	JPEG Unit (JCUA)		Provided	

Table 1.4 RH850/D1M1 development products overview (4/4)

Series name:			D1M1-V2	D1M1A
Other functions	LCD Bus I/F (LCBI)		18 bit output, max. 10 MHz	–
	Clock Monitors (CLMA)		for MainOsc, LS IntOsc, HS IntOsc, PLL0, PLL1, Video Input pixel clocks	
	Data CRC (DCRA)		Provided	
	Power-On-Clear (POC)		Provided	
	Intelligent Stepper Motor Driver (ISM), incl. zero point detection for each channel		1 unit, 6 channels	
	Error Correction Coding (ECC)		for Code Flash, Data Flash, Local RAM, Retention RAM, Video RAM, RS-CAN RAM, Caches tag/data RAMs	
	Error Control Module (ECM)		Master	
	Intelligent Cryptographic Unit (ICU-S2)		Provided	
	On-Chip debug (OCD)		Provided	
	Boundary Scan		Provided	
Voltage supply*2	Internal logic	AWO*3	3.3 V, 5 V via on-chip voltage regulator	
		ISO*3	3.3 V via on-chip voltage regulator	
	I/O buffers	GPIO*3	3.3 V, 5 V	
		SDR-SDRAM	–	3.3 V
	A/D Converter supplies		nominal 3.3 V, 5 V	
Package	Type		LQFP	BGA
	Pins		176	272
	pin/ball pitch		0.5 mm	1.0 mm

Note 1. Devices with RS-CAN or RS-CANFD interfaces have different product names, refer to **Section 1.5, Ordering Information**.

Note 2. The supply voltages are given as nominal values. Refer to the data sheet for detail specification of electrical values.

Note 3. AWO: Always-On-Area
ISO: Isolated-Area
GPIO: General purpose I/O port

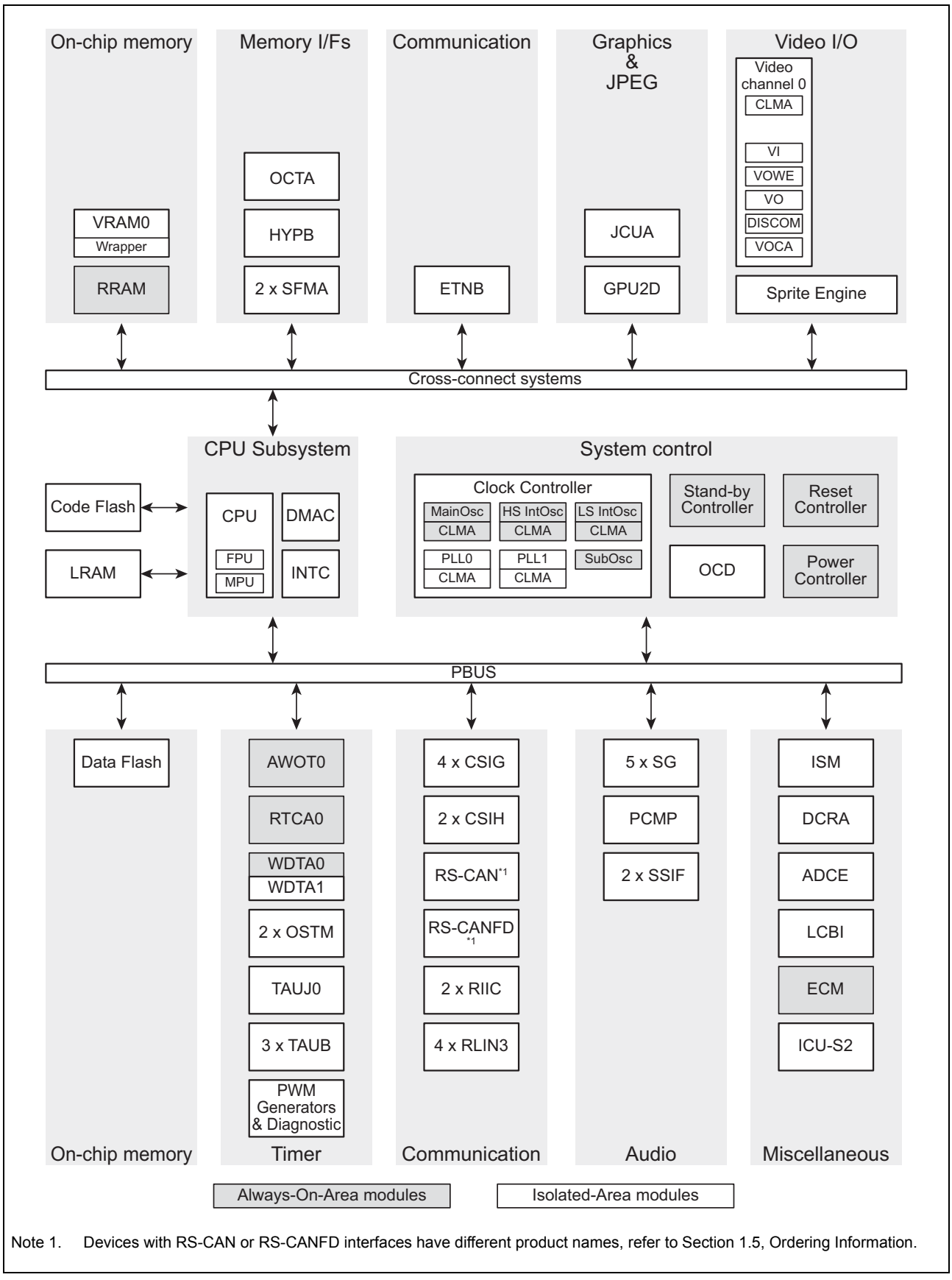
Note 4. Not all flash memory interfaces and all Video Output VO0 interfaces can be used in parallel. Refer to Table 1.5, Flash memory and Video output VO0 connection options for D1M1-V2 and D1M1A devices.

Table 1.5 Flash memory and Video output VO0 connection options for D1M1-V2 and D1M1A devices

Option	HYPB0	OCTA0	SFMA0	SFMA2	SFMA1	NFMA	VO0
	Port group P21			Port groups P21, P22, P45	Port group P42		Port group P45
1	HYPB0 or OCTA0 or SFMA0 (8 bit, 80 MHz)			–	–	8 bit, 20 MHz	All I/Fs *2
2 *1	–	–	4 bit, 80 MHz	4 bit, 80 MHz	–	8 bit, 20 MHz	OpenLDI
3	HYPB0 or OCTA0 or SFMA0 (8 bit, 80 MHz)			–	8 bit, 40 MHz	–	All I/Fs

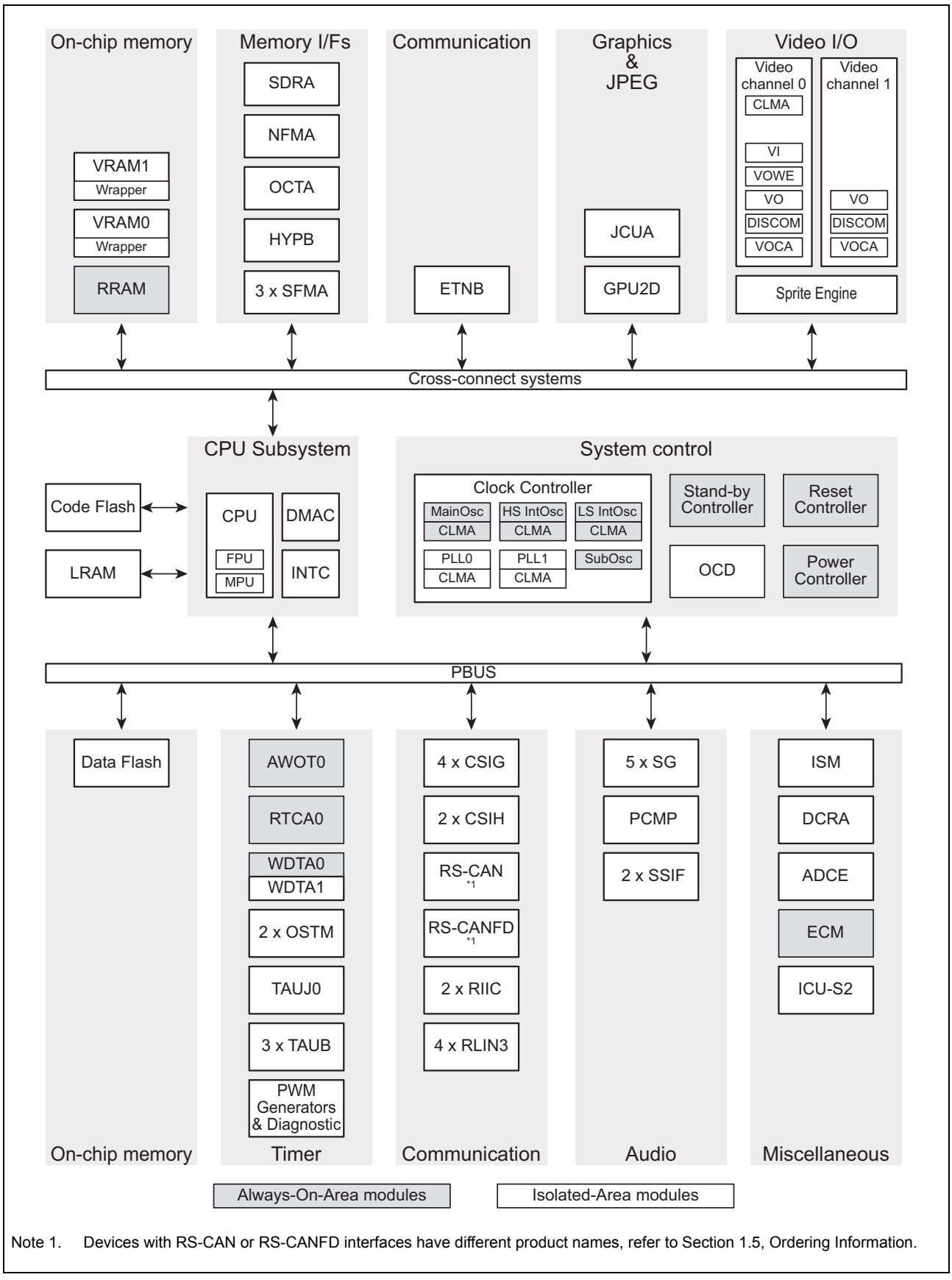
Note 1. This option is not available for D1M1-V2 devices.

Note 2. All interfaces means: LVTTL, OpenLDI, VODDR



Note 1. Devices with RS-CAN or RS-CANFD interfaces have different product names, refer to Section 1.5, Ordering Information.

Figure 1.4 D1M1-V2 block diagram



Note 1. Devices with RS-CAN or RS-CANFD interfaces have different product names, refer to Section 1.5, Ordering Information.

Figure 1.5 D1M1A block diagram

1.4 RH850/D1M2 products overview

Table 1.6 RH850/D1M2 products overview (1/4)

Series name:			D1M2_		D1M2H_	
			3.75M	5M	3.75M	5M
Memory	Code Flash		3.75 MB	5 MB	3.75 MB	5 MB
	Local RAM (LRAM)		512 KB			
	Retention RAM (RRAM)		16 KB			
	Data Flash		64 KB			
	Video RAM (VRAM) with Video RAM wrapper		2 x 1.55 MB			
External memory interfaces	SDRAM I/F	Bus width	16-bit		32-bit	
		Mode	DDR2-SDRAM I/F (SDRB)			
		Max. clock	240 MHz			
	Serial Flash Memory I/F 0 (SFMA0)	Bus width	8 bit			
		Mode	SDR, DDR			
		Max. clock	SDR: 120 MHz, DDR: 80 MHz			
CPU	CPU System		G3M			
	CPU frequency		240 MHz			
	Floating Point Unit (FPU)		Provided			
	Memory Protection Unit (MPU)		Provided			
	Memory caches	Instruction cache	8 KB/4-way associative			
		Non-CPU system memories	32 KB/4-way associative			
DMA			16 channels			
Operating clock	Main Oscillator (MainOsc)		8 to 16 MHz			
	Low Speed Internal Oscillator (LS IntOsc)		typ. 240 kHz			
	High Speed Internal Oscillator (HS IntOsc)		typ. 8 MHz			
	Sub Oscillator (SubOsc)		typ. 32.768 kHz			
	Spread-spectrum PLL0		max. 480 MHz			
	PLL1		max. 960 MHz			
	PLL2		max. 480 MHz			
I/O port			159		199	
A/D Converter (ADCE)			20 channels, 12 bit resolution			
Timer	Timer Array Unit B (TAUB)		3 units (16 bit resolution, 16 channels/unit)			
	Timer Array Unit J (TAUJ)		1 unit (32 bit resolution, 4 channels/unit)			
	Operating System Timer (OSTM)		2 units (32 bit resolution, 1 channel/unit)			
	Alway-On-Area Timer (AWOT)		1 unit (32 bit resolution, 1 channel/unit)			
	Real-Time Clock (RTCA)		Provided			
	Window Watchdog Timer A (WDTA)		2 units			
	PWM Generators with Diagnostic		1 unit (12 bit resolution, 24 PWM generators, 12 with diagnostic capability)			

Table 1.6 RH850/D1M2 products overview (2/4)

Series name:			D1M2_		D1M2H_	
			3.75M	5M	3.75M	5M
Communi-cation interfaces	Clocked Serial Interface G (CSIG)		4 channels			
	Clocked Serial Interface H (CSIH)		2 channels			
	CAN Interface (RS-CAN)		3 channels (total 192 message buffers)*1			
	CAN Interface (RS-CANFD)		3 channels (total 192 message buffers)*1			
	LIN/UART Interface (RLIN3)		4 channels			
	I^2C Interface (RIIC)		2 channels			
	Ethernet AVB MAC (ETNB)		1 channel (Media Access Controller for up to 100 Mbps, with Audio Video Bridging)			
	Media Local Bus (MLBB)		–		1 channel (50 Mbps)	
External interrupts	Maskable		11			
	Non-maskable (NMI)		1			
Audio	Sound Generator (SG)		5 units			
	PCM-PWM Converter (PCMP)		1 unit			
	I^2S Interface (SSIF)		2 units (1 channel/unit)			
Video and Graphics	Video Output 0 (VO0)	Resolution	1280 x 1024 pixels			
		Color format	RGB888			
		Max. pixel clock	48 MHz LVTTL, 48 MHz RSDS			
		Layers	4			
		I/F	LVTTL, single RSDS selectable for channel 0 or 1			
		Predistortion	Warping Engine (VOWE)			
		Timing Controller (TCON)	7 programmable signals			
		Video output data control	Video Output Checker (VOCA) 2 CRC checker (DISCOM)			

Table 1.6 RH850/D1M2 products overview (3/4)

Series name:			D1M2_		D1M2H_	
			3.75M	5M	3.75M	5M
Video and Graphics	Video Output 1 (VO1)	Resolution	1280 x 1024 pixels			
		Color format	RGB888			
		Max. pixel clock	48 MHz LVTTL, 48 MHz RSDS			
		Layers	4			
		I/F	LVTTL, single RSDS selectable for channel 0 or 1			
		Predistortion	–			
		Timing Controller (TCON)	7 programmable signals			
		Video output data control	Video Output Checker (VOCA) 2 CRC checker (DISCOM)			
	Video Outputs shared features	RLE decoding	Provided for background layers			
		Sprite layer	3 x 16 sprites			
	Video Input (VI)	Channels	1 channel		2 channels	
		Resolution	1024 x 1024 pixels			
		Pixel clock	48 MHz			
		Color formats	RGB888, ITU656			
		I/F	LVTTL for both channels, single MIPI CSI-2 for channel 0			
	Graphics Processing Unit		2D Graphics Processing Unit (GPU2D), 240 MHz operation clock			
	JPEG Unit (JCUA)		Provided			
Other functions	Clock Monitors (CLMA)		for MainOsc, LS IntOsc, HS IntOsc, PLL0, PLL1, Video Input pixel clocks			
	Data CRC (DCRA)		Provided			
	Power-On-Clear (POC)		Provided			
	Intelligent Stepper Motor Driver (ISM), incl. zero point detection for each channel		1 unit, 4 channels		1 unit, 6 channels	
	Error Correction Coding (ECC)		for Code Flash, Data Flash, Local RAM, Retention RAM, Video RAM, RS-CAN RAM, Caches tag/data RAMs			
	Error Control Module (ECM)		Master/Checker			
	Intelligent Cryptographic Unit (ICU-S2)		Provided			
	On-Chip debug (OCD)		Provided			
	Boundary Scan		Provided			

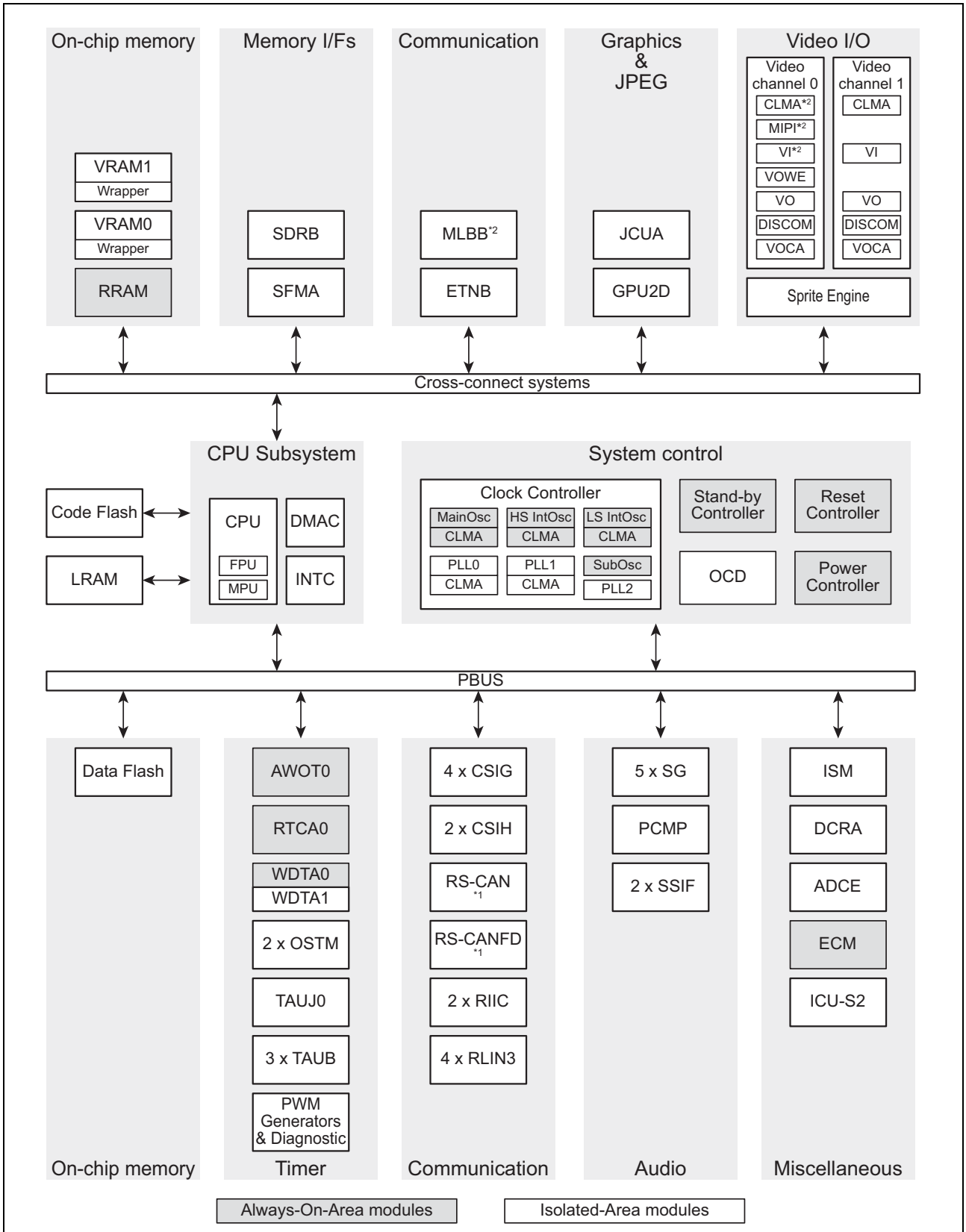
Table 1.6 RH850/D1M2 products overview (4/4)

Series name:			D1M2_		D1M2H_	
			3.75M	5M	3.75M	5M
Voltage supply*2	Internal logic	AWO*3	3.3 V, 5 V via on-chip voltage regulator			
		ISO*3	1.25 V			
	I/O buffers	GPIO*3	3.3 V, 5 V			
		SDR-SDRAM	–			
		DDR2-SDRAM	1.8 V			
	A/D Converter supplies		nominal 3.3 V, 5 V			
Package	Type		BGA			
	Pins		376		484	
	pin/ball pitch		1.0 mm			

Note 1. Devices with RS-CAN or RS-CANFD interfaces have different product names, refer to **Section 1.5, Ordering Information**.

Note 2. The supply voltages are given as nominal values. Refer to the data sheet for detail specification of electrical values.

Note 3. AWO: Always-On-Area
ISO: Isolated-Area
GPIO: General purpose I/O port



Note 1. Devices with RS-CAN or RS-CANFD interfaces have different product names, refer to Section 1.5, Ordering Information.
Note 2. These modules are not available for the D1M2 devices.

Figure 1.6 D1M2(H) block diagram

1.5 Ordering Information

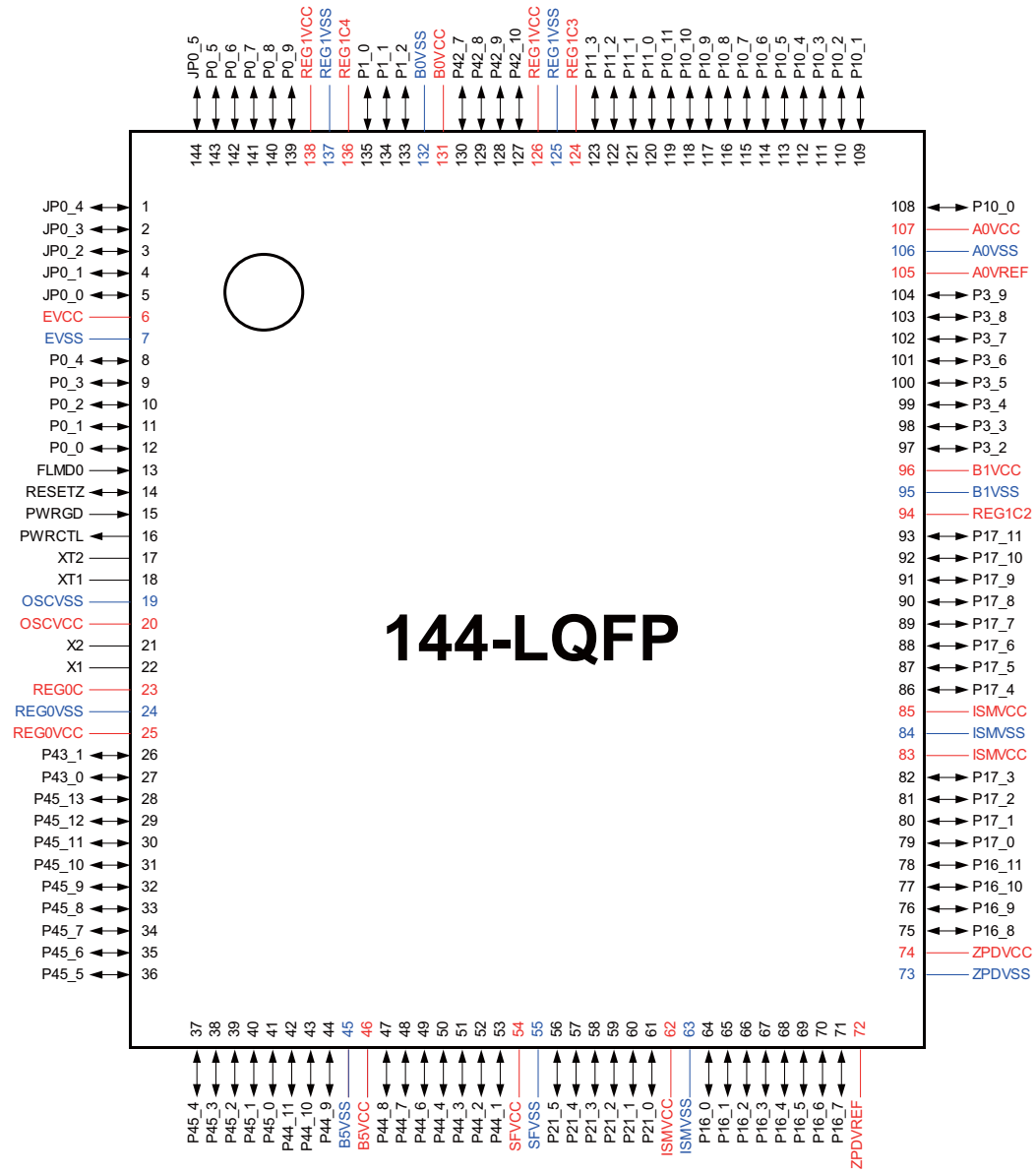
Table 1.7 Ordering information

Series name	Part number	Renesas order code	Remarks
D1L1	R7F701401		D1L1 with RS-CAN I/F
	R7F701421		D1L1 with RS-CANFD I/F
D1L2	R7F701402		D1L2 with RS-CAN I/F
	R7F701422		D1L2 with RS-CANFD I/F
D1L2H	R7F701403		D1L2H with RS-CAN I/F
	R7F701423		D1L2H with RS-CANFD I/F
D1M1_3.75M	R7F701404		D1M1 with 3.75 MB Code Flash and RS-CAN I/F
D1M1_5M	R7F701405		D1M1 with 5 MB Code Flash and RS-CAN I/F
D1M1H_3.75M	R7F701406		D1M1H with 3.75 MB Code Flash and RS-CAN I/F
D1M1H_5M	R7F701407		D1M1H with 5 MB Code Flash and RS-CAN I/F
D1M2_3.75M	R7F701408		D1M2 with 3.75 MB Code Flash and RS-CAN I/F
	R7F701428		D1M2 with 3.75 MB Code Flash and RS-CANFD I/F
D1M2_5M	R7F701410		D1M2 with 5 MB Code Flash and RS-CAN I/F
	R7F701430		D1M2 with 5 MB Code Flash and RS-CANFD I/F
D1M2H_3.75M	R7F701411		D1M2H with 3.75 MB Code Flash and RS-CAN I/F
	R7F701431		D1M2H with 3.75 MB Code Flash and RS-CANFD I/F
D1M2H_5M	R7F701412		D1M2H with 5 MB Code Flash and RS-CAN I/F
	R7F701432		D1M2H with 5 MB Code Flash and RS-CANFD I/F
D1M1-V2	R7F701442		D1M1-V2 with 4 MB Code Flash and RS-CAN I/F
	R7F701462		D1M1-V2 with 4 MB Code Flash and RS-CANFD I/F
D1M1A	R7F701441		D1M1A with 4 MB Code Flash and RS-CAN I/F
	R7F701461		D1M1A with 4 MB Code Flash and RS-CANFD I/F

Section 2 Pins

2.1 Pin Connection Diagrams

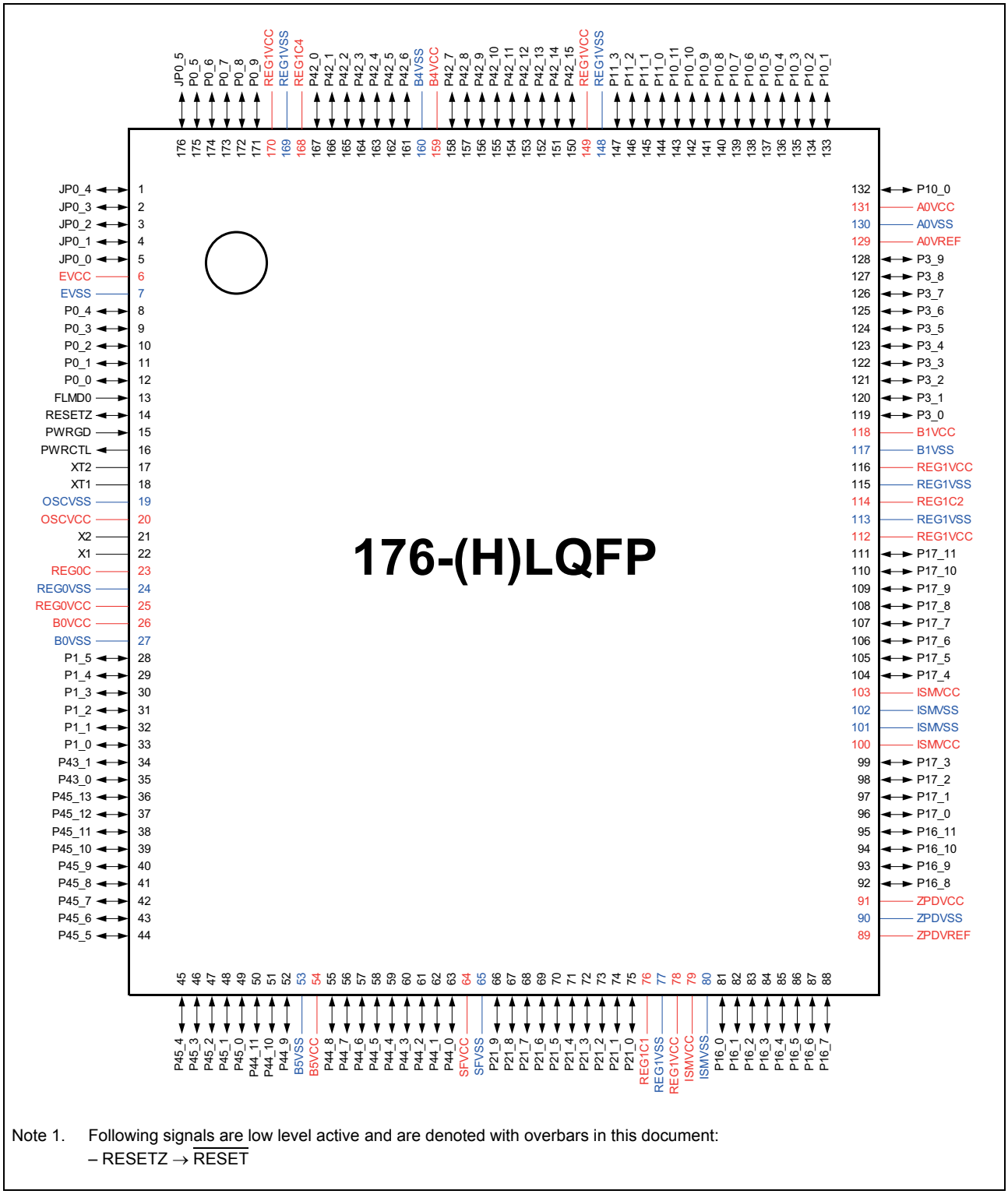
2.1.1 D1L1 and D1L2 pin connection



Note 1. Following signals are low level active and are denoted with overbars in this document:
– RESETZ → $\overline{\text{RESET}}$

Figure 2.1 D1L1 and D1L2 pin connection diagram (144 pins LQFP package)

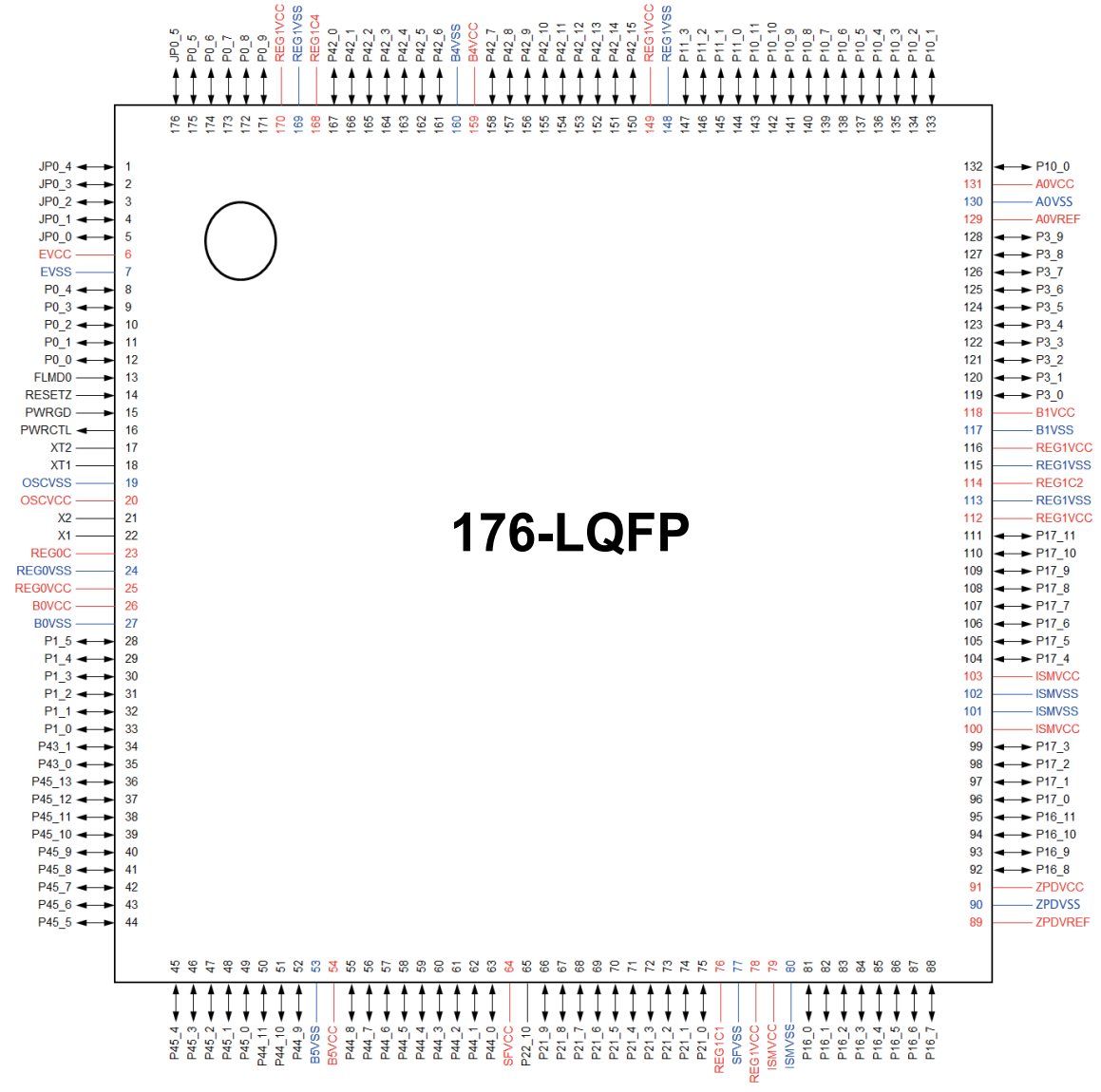
2.1.2 D1L2H and D1M1 pin connection



Note 1. Following signals are low level active and are denoted with overbars in this document:
– RESETZ → $\overline{\text{RESET}}$

Figure 2.2 D1L2H and D1M1 pin connection diagram (176 pins (H)LQFP package)

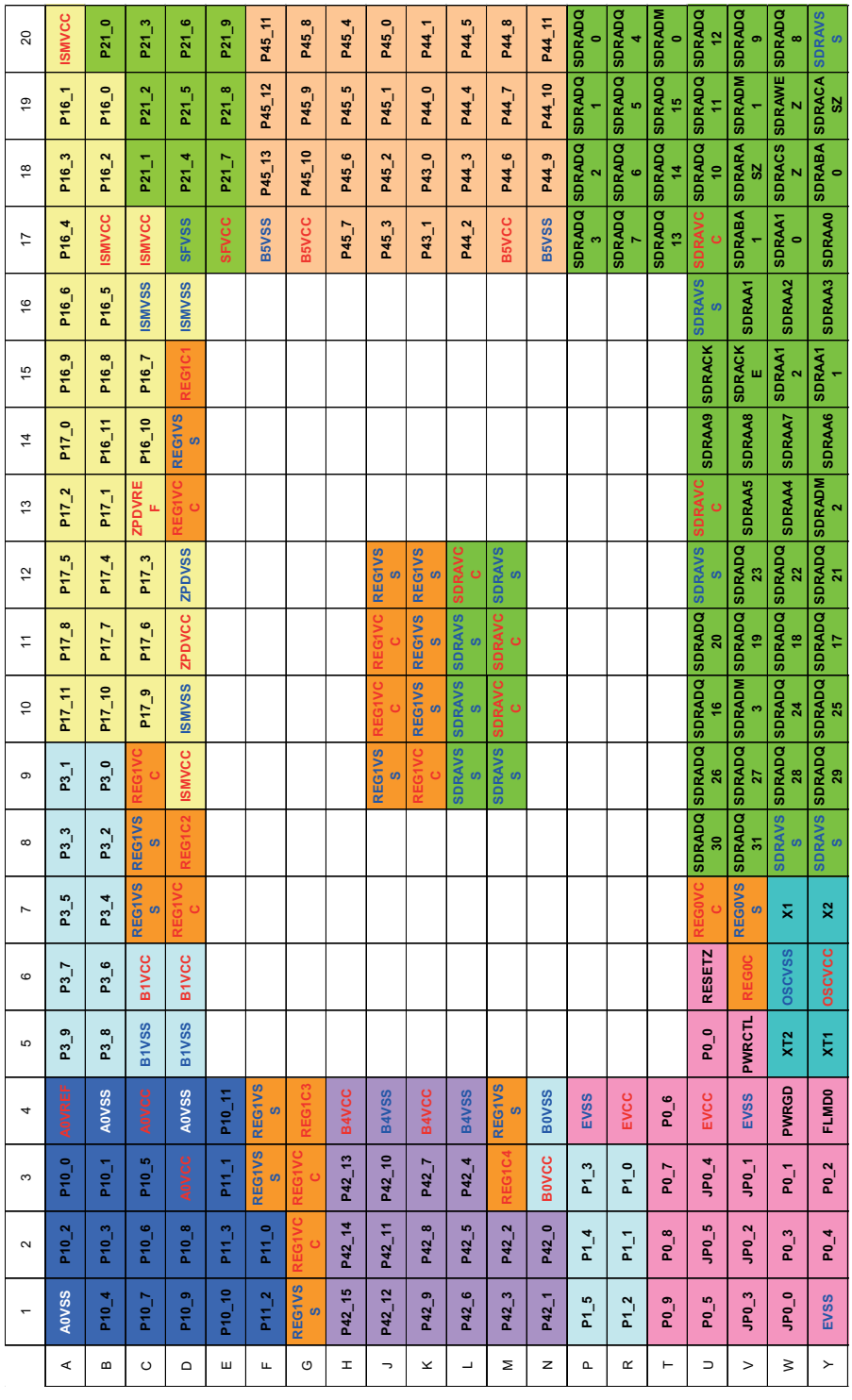
2.1.3 D1M1-V2 pin connection



Note 1. Following signals are low level active and are denoted with overbars in this document:
– RESETZ → $\overline{\text{RESET}}$

Figure 2.3 D1M1-V2 pin connection diagram (176 pins LQFP package)

2.1.4 D1M1H pin connection

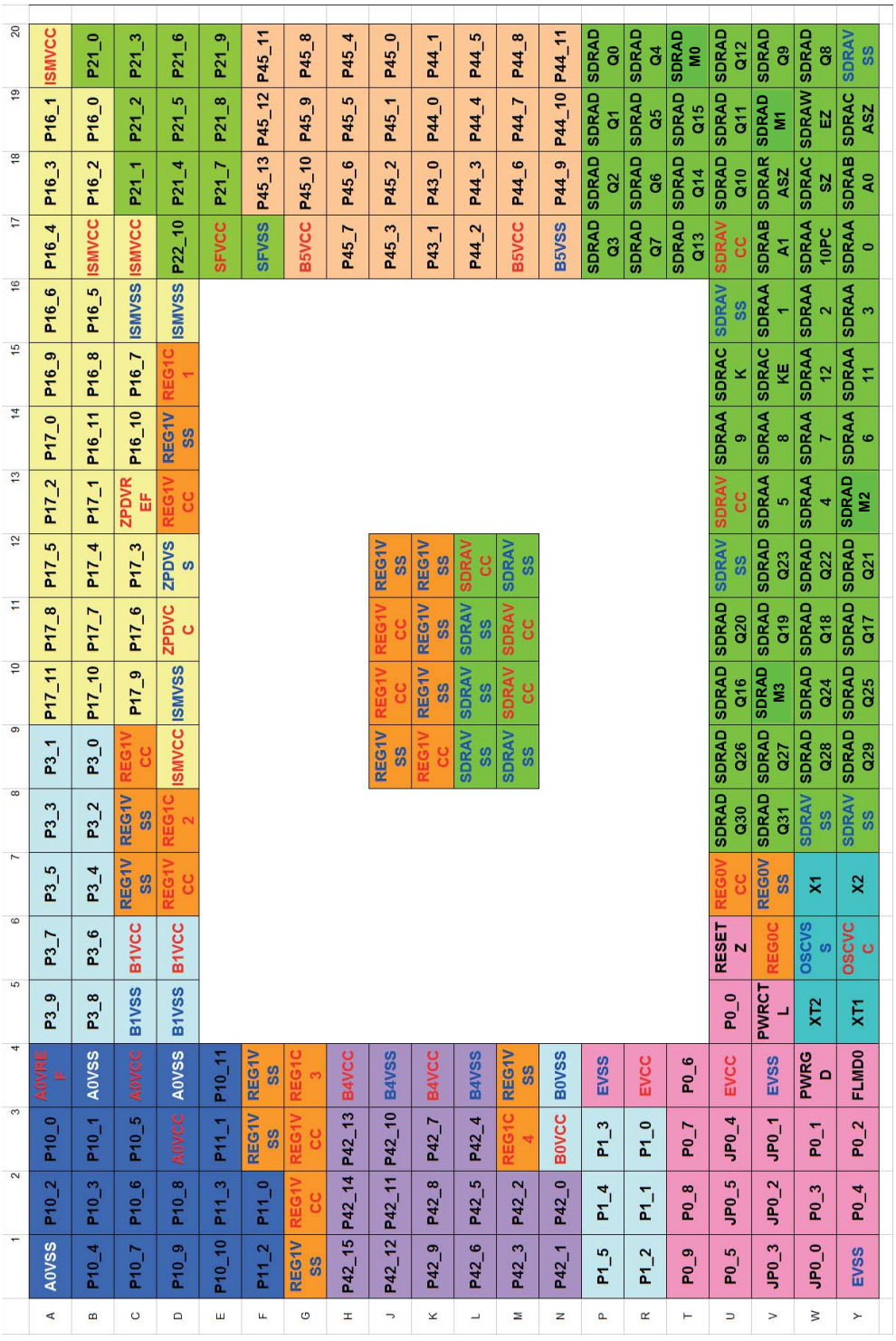


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	A0VSS	P10_2	P10_0	A0VREF	P3_9	P3_7	P3_5	P3_3	P3_1	P17_11	P17_8	P17_5	P17_2	P17_0	P16_9	P16_6	P16_4	P16_3	P16_1	ISMVCC
B	P10_4	P10_3	P10_1	A0VSS	P3_8	P3_6	P3_4	P3_2	P3_0	P17_10	P17_7	P17_4	P17_1	P16_11	P16_8	P16_5	ISMVCC	P16_2	P16_0	P21_0
C	P10_7	P10_6	P10_5	A0VCC	B1VSS	B1VCC	REG1VSS	REG1VSS	REG1VCC	P17_9	P17_6	P17_3	ZPDVREF	P16_10	P16_7	ISMVSS	ISMVCC	P21_1	P21_2	P21_3
D	P10_9	P10_8	A0VCC	A0VSS	B1VSS	B1VCC	REG1VCC	REG1C2	ISMVCC	ISMVSS	ZPDVCC	ZPDVSS	REG1VCC	REG1VSS	REG1C1	ISMVSS	SFVSS	P21_4	P21_5	P21_6
E	P10_10	P11_3	P11_1	P10_11													SFVCC	P21_7	P21_8	P21_9
F	P11_2	P11_0	REG1VSS	REG1VSS													B5VSS	P45_13	P45_12	P45_11
G	REG1VSS	REG1VCC	REG1VCC	REG1C3													B5VCC	P45_10	P45_9	P45_8
H	P42_15	P42_14	P42_13	B4VCC													P45_7	P45_6	P45_5	P45_4
J	P42_12	P42_11	P42_10	B4VSS					REG1VSS	REG1VCC	REG1VCC	REG1VSS					P45_3	P45_2	P45_1	P45_0
K	P42_9	P42_8	P42_7	B4VCC					REG1VCC	REG1VSS	REG1VSS	REG1VSS					P43_1	P43_0	P44_0	P44_1
L	P42_6	P42_5	P42_4	B4VSS					SDRAVSS	SDRAVSS	SDRAVSS	SDRAVCC					P44_2	P44_3	P44_4	P44_5
M	P42_3	P42_2	REG1C4	REG1VSS					SDRAVSS	SDRAVCC	SDRAVCC	SDRAVSS					B5VCC	P44_6	P44_7	P44_8
N	P42_1	P42_0	B0VCC	B0VSS													B5VSS	P44_9	P44_10	P44_11
P	P1_5	P1_4	P1_3	EVSS													SDRADQ3	SDRADQ2	SDRADQ1	SDRADQ0
R	P1_2	P1_1	P1_0	EVCC													SDRADQ7	SDRADQ6	SDRADQ5	SDRADQ4
T	P0_9	P0_8	P0_7	P0_6													SDRADQ13	SDRADQ14	SDRADQ15	SDRADM0
U	P0_5	JP0_5	JP0_4	EVCC	P0_0	RESETZ	REG0VCC	SDRADQ30	SDRADQ26	SDRADQ16	SDRADQ20	SDRAVSS	SDRAVCC	SDRAA9	SDRACKE	SDRAVSS	SDRAVCC	SDRADQ10	SDRADQ11	SDRADQ12
V	JP0_3	JP0_2	JP0_1	EVSS	PWRCTL	REG0C	REG0VSS	SDRADQ31	SDRADQ27	SDRADM3	SDRADQ19	SDRADQ23	SDRAA5	SDRAA8	SDRAA12	SDRAA1	SDRABA1	SDRARASZ	SDRADM1	SDRADQ9
W	JP0_0	P0_3	P0_1	PWRGD	XT2	OSCVSS	X1	SDRAVSS	SDRADQ28	SDRADQ24	SDRADQ18	SDRADQ22	SDRAA4	SDRAA7	SDRAA11	SDRAA2	SDRAA1	SDRACSZ	SDRAWEZ	SDRADQ8
Y	EVSS	P0_4	P0_2	FLMD0	XT1	OSCVCC	X2	SDRAVSS	SDRADQ29	SDRADQ25	SDRADQ17	SDRADQ21	SDRADM2	SDRAA6	SDRAA11	SDRAA3	SDRAA0	SDRABA0	SDRACASZ	SDRAVSS

Note 1. Following signals are low level active and are denoted with overbars in this document:
- RESETZ → $\overline{\text{RESET}}$
- SDRACSZ → $\overline{\text{SDRACS}}$
- SDRACASZ → $\overline{\text{SDRACAS}}$
- SDRARASZ → $\overline{\text{SDRARAS}}$
- SDRAWEZ → $\overline{\text{SDRAWE}}$

Figure 2.4 D1M1H pin connection diagram (272 pins BGA package)

2.1.5 D1M1A pin connection



Note 1. Following signals are low level active and are denoted with overbars in this document:
- RESETB → $\overline{\text{RESET}}$
- SDRACSB → $\overline{\text{SDRACS}}$
- SDRACASB → $\overline{\text{SDRACAS}}$
- SDRARASB → $\overline{\text{SDRARAS}}$
- SDRAWEB → $\overline{\text{SDRAWE}}$

Figure 2.5 D1M1A pin connection diagram (272 pins BGA package)

NOTE

The D1M1A pin connection is backward compatible to the D1M1H pin connection.
The pin connection diagram of D1M1A (Figure 2.5) and D1M1H (Figure 2.4) are identical with the exception of port pin P22_10 (ball D17), that is added in D1M1A.

2.1.6 D1M2 and D1M2H pin connection

	1	2	3	4	5	6	7	8	9	10	11
A	SDRBVSS	SDRBVSS	SDRBVSS	SDRBA10	SDRBDQ14	SDRBDQ9	SDRBVSS	SDRBDQS1B	SDRBDQ5	SDRBVSS	SDRBDQ2
B	SDRBVSS	SDRBVSS	SDRBA12	SDRBA7	SDRBBA2	SDRBDQ13	SDRBDQ10	SDRBDQS1	SDRBDQ3	SDRBDQ0	SDRBVSS
C	SDRBVSS	SDRBCK	SDRBVSS	SDRBA9	SDRBA5	SDRBVSS	SDRBDQ11	SDRBDM1	SDRBVSS	SDRBDQ4	SDRBDQS0B
D	SDRBCKB	SDRBA1	SDRBA8	SDRBVSS	SDRBA3	SDRBDQ15	SDRBDQ12	SDRBDQ8	SDRBDQ7	SDRBDQ6	SDRBDQS0
E	SDRBBA0	SDRBBA1	SDRBCS	SDRBVSS	SDRBCKVSS	SDRBVREF	SDRBVSS	SDRBVCC	SDRBVSS	SDRBVCC	SDRBVSS
F	SDRBVSS	SDRBCKE	SDRBA2	SDRBA13	SDRBCKVCC						
G	SDRBWE	SDRBRAS	SDRBA6	SDRBA4	SDRBVCC						
H	SDRBODT	SDRBCAS	SDRBA0	SDRBA11	SDRBVSS						
J	SDRBVSS	SDRBVSS	SDRBVCC	SDRBVCC	SDRBVCC				ISOVSS	ISOVSS	ISOVSS
K	SDRBVSS	SDRBVSS	SDRBVSS	SDRBVSS	SDRBVSS				ISOVSS	ISOVSS	ISOVSS
L	SDRBVSS	SDRBVSS	SDRBVSS	SDRBVSS	SDRBVCC				ISOVSS	ISOVSS	ISOVSS
M	SDRBVSS	SDRBVSS	SDRBVSS	ISOVSS	ISOVSS				ISOVSS	ISOVSS	ISOVSS
N	SDRBVSS	SDRBVSS	SDRBVSS	ISOVDD	ISOVDD				ISOVSS	ISOVSS	ISOVSS
P	P1_5	P1_4	P1_3	B0VCC	B0VCC				ISOVSS	ISOVSS	ISOVSS
R	P1_2	P1_1	P1_0	B0VSS	B0VSS						
T	P2_11	P2_9	P2_7	ISOVDD	ISOVDD						
U	P2_10	P2_8	P2_6	ISOVSS	ISOVSS						
V	P2_5	P2_3	P2_1	P2_0	B1VCC	B1VSS	REG1VCC	EVCC	REG0VSS	REG0VCC	REG0C
W	P2_4	P2_2	P3_8	B1VCC	B1VSS	REG1VSS	REG1VCC	EVCC	EVSS	JP0_4	JP0_2
Y	P3_9	P3_7	P3_4	P3_0	P0_8	P0_6	P0_3	PWRGD	FLMD0	JP0_3	JP0_0
AA	P3_5	P3_3	P3_1	P0_9	P0_5	P0_2	P0_0	PWRCTL	XT2	JP0_1	X2
AB	ISOVSS	P3_6	P3_2	P0_7	P0_4	P0_1	JP0_5	RESETZ	XT1	OSCVSS	X1

	12	13	14	15	16	17	18	19	20	21	22
A	SDRBVSS	P21_12	P21_9	P21_6	P21_3	P43_11	P43_10	P43_7	P47_9	P47_5	ISOVSS
B	SDRBDM0	P21_11	P21_8	P21_5	P21_2	P43_12	P43_9	P47_10	P47_6	P47_3	P47_4
C	SDRBVSS	P21_10	P21_7	P21_4	P21_1	P21_0	P43_8	P47_8	P47_7	P47_2	P46_15
D	SDRBDQ1	PLLVSS	PLLVCC	SFVCC	SFVSS	B3VCC	B3VSS	ISOVSS	P47_1	P46_13	P46_14
E	SDRBVCC	PLLVSS	PLLVCC	SFVCC	SFVSS	B3VCC	B3VSS	ISOVSS	P47_0	P46_12	P46_9
F							ISOVDD	ISOVDD	P46_11	P46_10	P46_7
G							ISOVSS	ISOVSS	P46_8	P46_6	P46_5
H							B2VCC	B2VCC	P46_4	P46_3	P46_2
J	ISOVSS	ISOVSS	ISOVSS				B2VSS	B2VSS	P43_6	P46_1	P46_0
K	ISOVSS	ISOVSS	ISOVSS				ISMVCC	ISMVCC	P43_3	P43_4	P43_5
L	ISOVSS	ISOVSS	ISOVSS				ISMVSS	ISMVSS	P43_0	P43_1	P43_2
M	ISOVSS	ISOVSS	ISOVSS				ISMVCC	P16_5	P16_2	P16_1	P16_0
N	ISOVSS	ISOVSS	ISOVSS				ZPDVREF	P16_10	P16_6	P16_4	P16_3
P	ISOVSS	ISOVSS	ISOVSS				ZPDVCC	P16_11	P16_9	P16_8	P16_7
R							ZPDVSS	P17_3	P17_2	P17_1	P17_0
T							RVCC	RVCC	P45_7	P45_12	P45_13
U							RVSS	RVSS	P45_6	P45_10	P45_11
V	OSCVSS	OSCVCC	A0VREF	A0VCC	A0VSS	ISOVDD	ISOVSS	RVSS	P45_3	P45_8	P45_9
W	OSCVSS	OSCVCC	P11_1	A0VCC	A0VSS	P10_0	ISOVDD	ISOVSS	P45_2	P45_4	P45_5
Y	P11_7	P11_4	P11_0	P10_9	P10_4	P10_1	P44_2	P44_3	P44_9	P45_0	P45_1
AA	P11_6	P11_3	P10_11	P10_8	P10_5	P10_2	P44_1	P44_4	P44_8	P44_10	P44_11
AB	P11_5	P11_2	P10_10	P10_7	P10_6	P10_3	P44_0	P44_5	P44_6	P44_7	RVSS

Note 1. Following signals are low level active and are denoted with overbars in this document:
- RESETB → $\overline{\text{RESET}}$
- SDRBCKB → $\overline{\text{SDRBCK}}$
- SDRBDQS[1:0]B → $\overline{\text{SDRBDQS}}$[1:0]
- SDRBCSB → $\overline{\text{SDRBCS}}$
- SDRBCASB → $\overline{\text{SDRBCAS}}$
- SDRBRASB → $\overline{\text{SDRBRAS}}$
- SDRBWEB → $\overline{\text{SDRBWE}}$

Figure 2.6 D1M2 pin connection diagram (376 pins BGA package)

	1	2	3	4	5	6	7	8	9	10	11	12	13
A	SDRBVSS	SDRBVSS	SDRBVSS	SDRBA10	SDRBDQ15	SDRBDQ12	SDRBVSS	SDRBDQS1B	SDRBDQ5	SDRBVSS	SDRBDQ2	SDRBVSS	P21_12
B	SDRBVSS	SDRBVSS	SDRBA12	SDRBA7	SDRBBA2	SDRBDQ13	SDRBDQ9	SDRBDQS1	SDRBDQ3	SDRBDQ0	SDRBVSS	SDRBDM0	P21_10
C	SDRBVSS	SDRBCK	SDRBVSS	SDRBA9	SDRBA5	SDRBVSS	SDRBDQ10	SDRBDM1	SDRBVSS	SDRBDQ4	SDRBDQS0B	SDRBVSS	P21_11
D	SDRBCKB	SDRBA1	SDRBA8	SDRBVSS	SDRBA3	SDRBDQ14	SDRBDQ11	SDRBDQ8	SDRBDQ7	SDRBDQ6	SDRBDQS0	SDRBDQ1	SFVSS
E	SDRBBA0	SDRBBA1	SDRBCS	SDRBVSS	SDRBCKVSS	SDRBVREF	SDRBVSS	SDRBVCC	SDRBVSS	SDRBVCC	SDRBVSS	SDRBVCC	SFVSS
F	SDRBVSS	SDRBCKE	SDRBA2	SDRBA13	SDRBCKVCC								
G	SDRBWE	SDRBRAS	SDRBA6	SDRBA4	SDRBVCC								
H	SDRBODT	SDRBCAS	SDRBA0	SDRBA11	SDRBVSS								
J	SDRBDQ17	SDRBDQ18	SDRBVSS	SDRBDQ16	SDRBVCC								
K	SDRBVSS	SDRBDQ23	SDRBDQ20	SDRBDQ21	SDRBVSS					ISOVSS	ISOVSS	ISOVSS	ISOVSS
L	SDRBDQ19	SDRBVSS	SDRBDQS2B	SDRBDQS2	SDRBVCC					ISOVSS	ISOVSS	ISOVSS	ISOVSS
M	SDRBVSS	SDRBDM2	SDRBDQ22	SDRBDQ24	SDRBVSS					ISOVSS	ISOVSS	ISOVSS	ISOVSS
N	SDRBDQ25	SDRBDQ27	SDRBVSS	SDRBDQ28	SDRBVCC					ISOVSS	ISOVSS	ISOVSS	ISOVSS
P	SDRBVSS	SDRBDQ29	SDRBDQ30	SDRBDQ26	SDRBVSS					ISOVSS	ISOVSS	ISOVSS	ISOVSS
R	SDRBDQS3B	SDRBDQS3	SDRBDM3	SDRBDQ31	SDRBVCC					ISOVSS	ISOVSS	ISOVSS	ISOVSS
T	P1_11	P1_10	P1_9	ISOVSS	ISOVSS					ISOVSS	ISOVSS	ISOVSS	ISOVSS
U	P1_8	P1_7	P1_6	ISOVDD	ISOVDD					ISOVSS	ISOVSS	ISOVSS	ISOVSS
V	P1_5	P1_3	P1_4	B0VSS	B0VSS								
W	P1_1	P1_2	P1_0	B0VCC	B0VCC								
Y	P2_11	P2_9	P2_10	B1VSS	B1VSS								
AA	P2_7	P2_8	P2_5	B1VCC	B1VCC								
AB	P2_6	P2_3	P2_4	ISOVSS	ISOVSS	ISOVDD	REG1VCC	REG1VSS	EVCC	EVCC	EVSS	REG0VCC	REG0C
AC	P2_0	P2_2	P2_1	P3_11	ISOVSS	ISOVDD	REG1VCC	REG1VSS	P0_5	JP0_4	REG0VSS	REG0VCC	JP0_2
AD	P3_13	P3_12	P3_9	P3_10	P3_3	P0_8	P0_9	P0_4	P0_1	PWRGD	FLMD0	JP0_3	JP0_0
AE	ISOVSS	P3_8	P3_7	P3_6	P3_1	P3_0	P0_7	P0_3	P0_0	PWRCTL	XT2	JP0_1	X2
AF	ISOVSS	ISOVSS	ISOVSS	P3_5	P3_4	P3_2	P0_6	P0_2	JP0_5	RESETZ	XT1	OSCVSS	X1

	14	15	16	17	18	19	20	21	22	23	24	25	26
A	P21_9	P21_6	P21_3	P42_15	P42_13	P42_10	P42_6	P42_5	P42_1	P43_11	P43_8	ISOVSS	ISOVSS
B	P21_7	P21_5	P21_2	P42_14	P42_12	P42_9	P42_7	P42_2	P42_0	P43_10	P43_7	P47_9	ISOVSS
C	P21_8	P21_4	P21_1	P21_0	P42_11	P42_8	P42_4	P42_3	P43_12	P43_9	P47_10	P47_4	P47_6
D	SFVCC	PLLVCC	PLLVSS	ISOVSS	ISOVDD	B4VSS	B4VCC	B4VCC	B4VSS	ISOVSS	P47_8	P47_3	P47_7
E	SFVCC	PLLVCC	PLLVSS	ISOVSS	ISOVDD	B4VSS	B4VCC	B4VCC	B4VSS	ISOVSS	P47_5	P47_2	P47_1
F									ISOVDD	ISOVDD	P46_15	P46_13	P47_0
G									B2VSS	B2VSS	P46_14	P46_12	P46_11
H									B2VCC	B2VCC	P46_9	P46_7	P46_10
J									ISOVSS	ISOVSS	P46_8	P46_5	P46_6
K	ISOVSS	ISOVSS	ISOVSS	ISOVSS					ISOVDD	ISOVDD	P46_4	P46_2	P46_3
L	ISOVSS	ISOVSS	ISOVSS	ISOVSS					ISMVCC	ISOVSS	P43_6	P46_0	P46_1
M	ISOVSS	ISOVSS	ISOVSS	ISOVSS					ISMVCC	P16_0	P43_3	P43_4	P43_5
N	ISOVSS	ISOVSS	ISOVSS	ISOVSS					ISMVSS	P16_2	P16_1	P43_1	P43_2
P	ISOVSS	ISOVSS	ISOVSS	ISOVSS					ISMVSS	P16_4	P16_7	P16_3	P43_0
R	ISOVSS	ISOVSS	ISOVSS	ISOVSS					ISMVCC	P16_6	P16_9	P16_8	P16_5
T	ISOVSS	ISOVSS	ISOVSS	ISOVSS					ISMVSS	P17_0	P17_3	P16_11	P16_10
U	ISOVSS	ISOVSS	ISOVSS	ISOVSS					ZPDVCC	P17_2	P17_5	P17_4	P17_1
V									ZPDVSS	P17_9	P17_8	P17_7	P17_6
W									ZPDVREF	P17_11	P17_10	P45_12	P45_13
Y									RVSS	RVSS	P45_5	P45_10	P45_11
AA									RVCC	RVCC	P45_4	P45_8	P45_9
AB	OSCVSS	OSCVCC	AVSS	MVCC	A0VREF	A0VCC	A0VSS	ISOVDD	ISOVSS	ISOVSS	P44_11	P45_6	P45_7
AC	OSCVSS	OSCVCC	AVSS	MVCC	P11_2	A0VCC	A0VSS	P10_1	ISOVDD	RVSS	P44_10	P45_2	P45_3
AD	P40_0	P40_1	P11_6	P11_3	P10_10	P10_7	P10_4	P10_0	P44_4	P44_5	P44_9	P45_0	P45_1
AE	P40_2	P40_4	P11_5	P11_1	P10_11	P10_8	P10_5	P10_2	P44_1	P44_3	P44_7	P44_8	RVSS
AF	P40_3	P40_5	P11_7	P11_4	P11_0	P10_9	P10_6	P10_3	P44_0	P44_2	P44_6	RVSS	RVSS

Note 1. Following signals are low level active and are denoted with overbars in this document:
- RESETB → $\overline{\text{RESET}}$
- SDRBCKB → $\overline{\text{SDRBCK}}$
- SDRBDQS[3:0]B → $\overline{\text{SDRBDQS}}$[3:0]
- SDRBCSB → $\overline{\text{SDRBCS}}$
- SDRBCASB → $\overline{\text{SDRBCAS}}$
- SDRBRASB → $\overline{\text{SDRBRAS}}$
- SDRBWEB → $\overline{\text{SDRBWE}}$

Figure 2.7 D1M2H pin connection diagram (484 pins BGA package)

2.2 Port Functions

2.2.1 Features

2.2.1.1 Port Group

This product has the following numbers of port groups:

Table 2.1 RH850/D1L/D1M Port Group

Product	Number of Group	Name of Group
D1L1	13	P0, P1, P3, P10, P11, P16, P17, P21, P42 to P45, JP0
D1L2	13	P0, P1, P3, P10, P11, P16, P17, P21, P42 to P45, JP0
D1L2H	13	P0, P1, P3, P10, P11, P16, P17, P21, P42 to P45, JP0
D1M1	13	P0, P1, P3, P10, P11, P16, P17, P21, P42 to P45, JP0
D1M1H	13	P0, P1, P3, P10, P11, P16, P17, P21, P42 to P45, JP0
D1M2	15	P0 to P3, P10, P11, P16, P17, P21, P43 to P47, JP0
D1M2H	17	P0 to P3, P10, P11, P16, P17, P21, P40, P42 to P47, JP0
D1M1A	14	P0, P1, P3, P10, P11, P16, P17, P21, P22, P42 to P45, JP0
D1M1-V2	14	P0, P1, P3, P10, P11, P16, P17, P21, P22, P42 to P45, JP0

Port Group Index n

Each port group is identified by its own index “n” (n = 0 to 3, 10, 11, 16, 17, 21, 22, 40, 42 to 47, JP0) throughout this section; e.g. PMCn for the port mode control register of the Pn pin.

2.2.1.2 Register Address

All port addresses are given as an offset from the individual base addresses, <PORTUn_Base>, <PORTOn_Base>, <JPORTUn_Base> and <JPORTOn_Base>.
These base addresses are listed in the following table.

Table 2.2 Port Base Address

Base address	Port group number	Address
<PORTUn_Base>	n = 0	FF61 8000$_H$
	n > 0	FF61 0000$_H$
<PORTOn_Base>	n = 0	FFC1 8000$_H$
	n > 0	FFC1 0000$_H$
<JPORTUn_Base>	n = 0	FF62 0000$_H$
<JPORTOn_Base>	n = 0	FFC2 0000$_H$

2.2.2 Overview

This product has various pins for input/output (I/O) functions, known as ports. The ports are organized in port groups.

This product also has several control registers to allocate the functions other than general I/O purpose to the corresponding pins.

For definitions of pin, port, and port group, see Section 2.2.2.1, Terms.

2.2.2.1 Terms

The terms described in this section are defined as follows.

- Pin:
  Denotes a physical pin. Every pin is denoted by a unique pin number. Most of pins can be used in multiple modes. Thus the pin name indicating its own function is assigned to each pin depending on the selected mode.
- Port group:
  Denotes a group of ports. The ports in the same port group have a common set of port mode control registers.
- Port mode / Port:
  A pin in port mode functions as a general purpose I/O pin. It is then called port.
  The corresponding name is Pn_m. For example, P0_7 denotes the port 7 of port group 0 and it is referred to as port P0_7.
- Alternative mode:
  In alternative mode, a pin can be used for various non-general purpose I/O functions; e.g. as the I/O pin of on-chip peripherals. Therefore, the corresponding pin name depends on the selected function. For example, the INTP0 pin denotes the pin for one of the external interrupt inputs.
  Note that two or more different names can refer to the same physical pin, e.g. for P0_2 and INTP0. Those different names indicate the function in which the pin is being operated.
- Port type:
  A control circuit depends on the assignment of a register to be set and a port type depends on the sort of the control circuit.

2.2.2.2 Overview of Pin Functions

The pin can operate in the following three different modes:

- Port mode (PMCn.PMCn_m = 0)
  The pin operates as a general purpose I/O port in port mode.
  PMn.PMn_m selects input or output.
- S/W I/O control alternative mode (PMCn.PMCn_m = 1, PIPCn.PIPCn_m = 0)
  The pin is operated by an alternative function in S/W I/O control alternative mode.
  The selection between input and output is made by S/W via the PMn.PMn_m control bits.
- Direct I/O control alternative mode (PMCn.PMCn_m = 1, PIPCn.PIPCn_m = 1)
  The pin is operated by an alternative function in direct I/O control alternative mode.
  In contrast to S/W I/O control alternative mode, input/out is directly controlled by the alternative function in this mode.

Table 2.3 shows the outline of the register settings.

Table 2.3 Pin Function Configuration (Outline)

Mode	Bit			I/O
	PMCn_m	PMn_m	PIPCn_m	
Port mode	0	0	X	O
		1*1		I
S/W I/O control alternative mode	1	0	0	O
		1*2	0	I
Direct I/O control alternative mode		X	1	Controlled by the alternative function

Note 1. The input buffer should be enabled (PIBCn_m = 1).
Note 2. The input buffer should be disabled (PIBCn_m = 0).

When a pin is operated in alternative mode (PMCn.PMCn_m = 1), one of six alternative functions can be selected by the PFCn, PFCEn and PFCAEn registers.

- S/W I/O control alternative functions (PIPCn.PIPCn_m = 0):
  - Outputs (PMn_m = 0): ALT-OUT1 to ALT-OUT6
  - Inputs (PMn_m = 1): ALT-IN1 to ALT-IN6
- Direct I/O control alternative functions (PIPCn.PIPCn_m = 1):
  - Input/Output of ALT-OUT1 to ALT-OUT6 and ALT-IN1 to ALT-IN6 is directly controlled by the alternative function.

Table 2.4 Outline of Alternative Mode Selection (PMCn.PMCn_m = 1)

Function	Register				I/O
	PFCAE	PFCE	PFC	PM*1	
Alternative output mode 1 (ALT-OUT1)	0	0	0	0	O
Alternative input mode 1 (ALT-IN1)	0	0	0	1	I
Alternative output mode 2 (ALT-OUT2)	0	0	1	0	O
Alternative input mode 2 (ALT-IN2)	0	0	1	1	I
Alternative output mode 3 (ALT-OUT3)	0	1	0	0	O
Alternative input mode 3 (ALT-IN3)	0	1	0	1	I
Alternative output mode 4 (ALT-OUT4)	0	1	1	0	O
Alternative input mode 4 (ALT-IN4)	0	1	1	1	I
Alternative output mode 5 (ALT-OUT5)	1	0	0	0	O
Alternative input mode 5 (ALT-IN5)	1	0	0	1	I
Alternative output mode 6 (ALT-OUT6)	1	0	1	0	O
Alternative input mode 6 (ALT-IN6)	1	0	1	1	I

Note 1. When PIPCn.PIPCn_m = 1, the I/O direction is directly controlled by the peripheral (alternative) function and PM is ignored.

When a pin is operated in alternative mode (PMCn.PMCn_m = 1), one of several alternative functions can be selected by the PFCn, PFCEn and PFCAEn registers.

2.2.2.3 Pin Data Input/Output

The registers used for data input and output are described below.

The source of the data to be read via the PPRn register depends on pin mode.

Output data

In port mode (PMCn.PMCn_m = 0), the value of Pn.Pn_m is output from the Pn_m pin.

Input data

A read operation of the PPRn register returns either the value of the Pn_m pin, the associated bit of the port register Pn.Pn_m, or the data output by an alternative function.

The source of the data read via PPRn depends on pin mode and setting of several control bits.

Table 2.5 summarizes the differences of PPRn read modes.

Table 2.5 PPRn_m Read Values

PMC n_m	PM n_m	PIBC n_m	PIPC n_m	PODC n_m	Mode	PPRn_m read value
0	1	0	X	X	Port input, input buffer disabled	Pn.Pn_m register
		1		X	Port input, input buffer enabled	Pn_m pin
	0	X		0	Port push-pull output	Pn.Pn_m register*1
				1	Port N-ch open-drain output	
1	1	0	0	X	S/W I/O control alternative input	Pn_m pin
	0	X		0	S/W I/O control alternative push-pull output	Alternative function internal output signal*1
				1	S/W I/O control alternative N-ch open-drain output	
	X		1	0	Direct I/O control alternative push-pull output	I/O port in alternative function mode: • Input: Pn_m pin • Output: Alternative function internal output signal*1
				1	Direct I/O control alternative N-ch open-drain output	

Note 1. When PBDCn_m = 1, Pn_m pin level is read via PPRn_m.

The control registers in **Table 2.5** have the following effects:

- PMCn.PMCn_m
  This bit selects either port mode (PMCn_m = 0) or alternative function mode (PMCn_m = 1).
- PMn.PMn_m
  This bit selects input (PMn_m = 1) or output (PMn_m = 0) in port mode (PMCn_m = 0) and S/W I/O control alternative function mode (PMCn_m = 1, PIPCn_m = 0).
- PIBCn.PIBCn_m
  This bit disables (PIBCn_m = 0) or enables (PIBCn_m = 1) the input buffer in input port mode (PMCn_m = 0 and PMn_m = 1). When the input buffer is disabled, PPRn_m reads the Pn.Pn_m bit, otherwise the Pn_m pin level is returned.
- PIPCn.PIPCn_m
  This bit selects either the S/W or direct I/O control alternative mode.
- PBDCn.PBDCn_m
  Setting this bit to 1 forces to read the Pn_m pin level via PPRn_m. Thus this enables a bidirectional mode, where the level of pin Pn_m can also be read back when the port is operated in an output mode.

- PODCn.PODCn_m bits
  These bits select push-pull output (PODCn_m bit = 0) or N-ch open-drain output (PODCn_m bit = 1).

Write to the Pn Register

The data to be output via port Pn_m in port mode (PMCn.PMCn_m = 0) is held in the port register Pn.

The Pn data can be rewritten in the following two different ways:

- Direct write to the Pn register
  New data can be directly written to the Pn register.
- Indirect operation to the Pn bit (set/reset/not)
  The indirect Pn operation is possible using the following two registers:
  - Port set/reset register: PSRn
    When the bit PSRn.PSRn(m+16) = 1, the value of bit PSRn.PSRn_m determines the value of Pn.Pn_m.
    Thus Pn_m can be set/reset without a direct write to Pn.
  - Port NOT register: PNOTn
    Setting PNOTn.PNOTn_m = 1 inverts the bit Pn.Pn_m without a direct write to Pn_m.
    Additionally, reading the level of the output pin can read the inverted value of the value that was initially set.

The indirect Pn set/reset/not operation provides access to single bits (not limited to one bit) of the Pn register while leaving all other Pn bits untouched.

2.2.3 Port Group Configuration Register

This section starts with an overview of all configuration registers and then presents all registers in detail. The configuration registers are classified as follows:

- 2.2.3.2 Pin Function Configuration
- 2.2.3.3 Pin Data Input/Output
- 2.2.3.4 Configuration of Electrical Characteristics

2.2.3.1 Outline

The following registers are used for the configuration of the individual pins of the port groups:

Table 2.6 Registers for Port Group Configuration (1/2)

Module Name				
n = 0	n > 0	Register Name	Symbol	Address
Pin function setting				
PORT_AWO	PORT_ISO	Port control register	PCRn_m*2	<PORTUn_base> + 2000_H + n × 40_H + m x 4
PORTJ_AWO	–		JPCR0_m*2	<JPORTU0_base> + 2000_H + m x 4
PORT_AWO	PORT_ISO	Port mode control register	PMCn	<PORTUn_base> + 0014_H + n × 40_H
PORTJ_AWO	–		JPMCn	<JPORTU0_base> + 0014_H
PORT_AWO	PORT_ISO	Port mode control set/reset register	PMCSRn	<PORTUn_base> + 0024_H + n × 40_H
PORTJ_AWO	–		JPMCSRn	<JPORTU0_base> + 0024_H
PORT_AWO	PORT_ISO	Port IP control register	PIPCn	<PORTOn_base> + 4008_H + n × 40_H
PORT_AWO	PORT_ISO	Port mode register	PMn	<PORTUn_base> + 0010_H + n × 40_H
PORTJ_AWO	–		JPMn	<JPORTU0_base> + 0010_H
PORT_AWO	PORT_ISO	Port mode set/reset register	PMSRn	<PORTUn_base> + 0020_H + n × 40_H
PORTJ_AWO	–		JPMSRn	<JPORTU0_base> + 0020_H
PORT_AWO	PORT_ISO	Port input buffer control register	PIBCn	<PORTOn_base> + 4000_H + n × 40_H
PORTJ_AWO	–		JPIBCn	<JPORTO0_base> + 4000_H
PORT_AWO	PORT_ISO	Port function control register	PFCn	<PORTUn_base> + 0018_H + n × 40_H
PORTJ_AWO	–		JPFCn	<JPORTU0_base> + 0018_H
PORT_AWO	PORT_ISO	Port function control expansion register	PFCEn	<PORTUn_base> + $001C_H$ + n × 40_H
PORTJ_AWO	–		JPFCEn	<JPORTU0_base> + $001C_H$
–	PORT_ISO	Port function control additional expansion register	PFCAEn	<PORTUn_base> + 0028_H + n × 40_H
Pin data input/output				
PORT_AWO	PORT_ISO	Port bi-direction control register	PBDCn	<PORTOn_base> + 4004_H + n × 40_H
PORTJ_AWO	–		JPBDCn	<JPORTO0_base> + 4004_H
PORT_AWO	PORT_ISO	Port pin read register	PPRn	<PORTUn_base> + $000C_H$ + n × 40_H
PORTJ_AWO	–		JPPRn	<JPORTU0_base> + $000C_H$
PORT_AWO	PORT_ISO	Port register	Pn	<PORTUn_base> + 0000_H + n × 40_H
PORTJ_AWO	–		JPn	<JPORTU0_base> + 0000_H
PORT_AWO	PORT_ISO	Port NOT register	PNOTn	<PORTUn_base> + 0008_H + n × 40_H
PORTJ_AWO	–		JPNOTn	<JPORTU0_base> + 0008_H
PORT_AWO	PORT_ISO	Port set/reset register	PSRn	<PORTUn_base> + 0004_H + n × 40_H
PORTJ_AWO	–		JPSRn	<JPORTU0_base> + 0004_H

Table 2.6 Registers for Port Group Configuration (2/2)

Module Name				
n = 0	n > 0	Register Name	Symbol	Address
PORT_AWO	PORT_ISO	Port output level inversion register	PINVn	<PORTUn_base> + 0030_H + n × 40_H
PORTJ_AWO	–		JPINVn	<JPORTU0_base> + 0030_H
Specifying electrical characteristics				
PORT_AWO	PORT_ISO	Pull-up option register	PUn	<PORTOn_base> + $400C_H$ + n × 40_H
PORTJ_AWO	–		JPUn	<JPORTO0_base> + $400C_H$
PORT_AWO	PORT_ISO	Pull-down option register	PDn	<PORTOn_base> + 4010_H + n × 40_H
PORTJ_AWO	–		JPDn	<JPORTO0_base> + 4010_H
PORT_AWO	PORT_ISO	Port drive strength control register	PDSCn	<PORTOn_base> + 4018_H + n × 40_H
PORTJ_AWO	–		JPDSCn	<JPORTO0_base> + 4018_H
PORT_AWO	PORT_ISO	Port open drain control register	PODCn	<PORTOn_base> + 4014_H + n × 40_H
PORTJ_AWO	–		JPODCn	<JPORTO0_base> + 4014_H
PORT_AWO	PORT_ISO	Port input buffer selection register	PISn	<PORTOn_base> + $401C_H$ + n × 40_H
PORTJ_AWO	–		JPISn	<JPORTO0_base> + $401C_H$
PORT_AWO	PORT_ISO	Port input buffer selection advanced register	PISAn	<PORTOn_base> + 4024_H + n × 40_H
PORTJ_AWO	–		JPISAn	<JPORTO0_base> + 4024_H
Port register protection				
For details about the registers for the port register protection refer to Section 4, Write-Protected Registers.				

Note 1. n: Port group number
m: Bit number in a port group

Note 2. In the header files the bit symbol of PCR0_m and JPCR0_m are different from others.
e.g. PCRn_0.PFC
– PORTn n > 0 PORT_ISOPFC
– PORTn n = 0 PORT_AWOPCR_PFC0_0
– JPORTn n = 0 PORTJ_AWOPCR_JPFC0_0

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

Base address

The PORTn base addresses <PORTUn_base>, <PORTOn_base>, <JPORTU0_base> and <JPORTO0_base> are defined in Register Addresses, Section 2.2.1.2, Register Address.

Register initial value

The initial values after reset release depend on the port, and are not described in the following register descriptions, but are given in Section 2.3.1, Port Registers.

2.2.3.2 Pin Function Configuration

(1) PMCn/JPMC0 — Port Mode Control Register

This register specifies whether the individual pins of port group n are in port mode or in alternative mode.

Access: The PMCn register can be read/written in 16-bit units.
The JPMC0 register can be read/written in 8-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMCn_15	PMCn_14	PMCn_13	PMCn_12	PMCn_11	PMCn_10	PMCn_9	PMCn_8	PMCn_7	PMCn_6	PMCn_5	PMCn_4	PMCn_3	PMCn_2	PMCn_1	PMCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.7 PMCn register contents

Bit Position	Bit Name	Function
15 to 0	PMCn_[15:0]	Specifies the operation mode of the corresponding pin. 0: Port mode 1: Alternative mode

CAUTIONS

1. The I/O control is not performed by just setting alternative mode (PMCn.PMCn_m). When the alternative function performs direct I/O control, set also the PIPCn.PIPCn_m bit to 1.
2. When a port is used as an input pin in alternative mode, there is a pin that passes a noise filter. Such a pin might need FCLAnCTLm.DFNAnCTL and the DNFAnEN register to be set. For details, see Section 2.6, Noise Filter and Edge Level Detection Circuit, and Section 2.7, Description of Port Noise Filter & Edge/Level Detection.

NOTES

1. The control bits of the JTAG port mode control register (JPMC0) are JPMC0_[7:0].
2. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

(2) PMCSRn/JPMCSR0 — Port Mode Control Set/Reset Register

This register provides an alternative method to write data to a bit in the PMCn register.

The upper 16 bits of PMCSRn_[31:16] specify whether the data is written to the PMCn.PMCn_m bit specified in the lower 16 bits of PMCSRn_[15:0]

Access: This register can be read/written in 32-bit units.
Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of the PMCn register.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMCSRn_31	PMCSRn_30	PMCSRn_29	PMCSRn_28	PMCSRn_27	PMCSRn_26	PMCSRn_25	PMCSRn_24	PMCSRn_23	PMCSRn_22	PMCSRn_21	PMCSRn_20	PMCSRn_19	PMCSRn_18	PMCSRn_17	PMCSRn_16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMCSRn_15	PMCSRn_14	PMCSRn_13	PMCSRn_12	PMCSRn_11	PMCSRn_10	PMCSRn_9	PMCSRn_8	PMCSRn_7	PMCSRn_6	PMCSRn_5	PMCSRn_4	PMCSRn_3	PMCSRn_2	PMCSRn_1	PMCSRn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.8 PMCSRn register contents

Bit Position	Bit Name	Function
31 to 16	PMCSRn_ [31:16]	Specifies whether the value of the corresponding lower bit of PMCSRn_m is written to PMCn_m: 0: PMCn_m does not depend on PMCSRn_m. 1: The value of PMCn_m is the same as that of PMCSRn_m. Example: When PMCSRn.PMCSRn_31 = 1, the value of bit PMCSRn.PMCSRn_15 is written to bit PMCn.PMCn_15 and output.
15 to 0	PMCSRn_ [15:0]	Specifies the PMCn_m value when the corresponding upper bit PMCSRn_(m+16) is 1: 0: PMCn_m = 0 1: PMCn_m = 1

NOTES

1. The control bits of the JTAG port mode control set/reset register (JPMCSR0) are JPMCSR0_[31:0].
2. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

(3) PIPCn — Port IP Control Register

This register specifies whether the I/O direction of pin Pn_m is controlled by the port mode register PMn.PMn_m or by an alternative function.

When the Pn_m pin is operated in alternative mode (PMCn.PMCn_m = 1) and the alternative function directly controls the I/O direction of Pn_m, PIPCn.PIPCn_m must be set to 1 as well. This hands over I/O control to the alternative function and overrules the PMn.PMn_m setting.

Access: This register can be read/written in 16-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIPCn_15	PIPCn_14	PIPCn_13	PIPCn_12	PIPCn_11	PIPCn_10	PIPCn_9	PIPCn_8	PIPCn_7	PIPCn_6	PIPCn_5	PIPCn_4	PIPCn_3	PIPCn_2	PIPCn_1	PIPCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.9 PIPCn register contents

Bit Position	Bit Name	Function
15 to 0	PIPCn_[15:0]	Specifies the I/O control mode. 0: I/O mode is selected by PMn.PMn_m (software I/O control). 1: I/O mode is selected by the peripheral function (direct I/O control).

NOTE

For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

(4) PMn/JPM0 — Port Mode Register

The PMn register specifies whether the individual pins of port group n are in input mode or in output mode.

Access: The PMn register can be read/written in 16-bit units.
The JPM0 register can be read/written in 8-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMn_15	PMn_14	PMn_13	PMn_12	PMn_11	PMn_10	PMn_9	PMn_8	PMn_7	PMn_6	PMn_5	PMn_4	PMn_3	PMn_2	PMn_1	PMn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.10 PMn register contents

Bit Position	Bit Name	Function
15 to 0	PMn_[15:0]	Specifies input/output mode of the corresponding pin. 0: Output mode (output enabled) 1: Input mode (output disabled)

NOTES

1. When a port is used in input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1), the input buffer needs to be enabled (PIBCn.PIBCn_m = 1).
2. After reset, PIPCn.PIPCn_m = 0 is set (I/O mode is controlled with PMn.PMn_m). Thus, PMn_m specifies the I/O direction of port mode (PMCn.PMCn_m = 0) and alternative mode (PMCn.PMCn_m = 1).
3. The control bits of the JTAG port mode register (JPM0) are JPM0_[7:0].
4. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

(5) PMSRn/JPMSR0 — Port Mode Set/Reset Register n

This register provides an alternative method to write data to a bit in the PMn register.

The upper 16 bits of PMSRn_[31:16] specify whether the data is written to the PMn.PMn_m bit specified in the lower 16 bits of PMSRn_[15:0].

Access: This register can be read/written in 32-bit units.
Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of the PMn register.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PMSRn_31	PMSRn_30	PMSRn_29	PMSRn_28	PMSRn_27	PMSRn_26	PMSRn_25	PMSRn_24	PMSRn_23	PMSRn_22	PMSRn_21	PMSRn_20	PMSRn_19	PMSRn_18	PMSRn_17	PMSRn_16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PMSRn_15	PMSRn_14	PMSRn_13	PMSRn_12	PMSRn_11	PMSRn_10	PMSRn_9	PMSRn_8	PMSRn_7	PMSRn_6	PMSRn_5	PMSRn_4	PMSRn_3	PMSRn_2	PMSRn_1	PMSRn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.11 PMSRn register contents

Bit Position	Bit Name	Function
31 to 16	PMSRn_[31:16]	Specifies whether the value of the corresponding lower bit of PMSRn_m is written to PMn_m: 0: PMn_m does not depend on PMSRn_m. 1: The value of PMn_m is the same as that of PMSRn_m. Example: When PMSRn.PMSRn_31 = 1, the value of bit PMSRn.PMSRn_15 is written to bit PMn.PMn_15 and output.
15 to 0	PMSRn_[15:0]	Specifies the PMn_m value when the corresponding upper bit PMSRn_(m+16) is 1: 0: PMn_m = 0 1: PMn_m = 1

NOTES

1. The control bits of the JTAG port mode set/reset register (JPMSR0) are JPMSR0_[7:0].
2. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

(6) PIBCn/JPIBC0 — Port Input Buffer Control Register

This register enables and disables the input buffer of a port pin in input port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 1).

Additionally, when pins are in bidirectional mode (PBDCn.PBDCn_m = 1), it is capable of selecting shared output level loop-back function and pin output level-read function with the setting of PIBCn.PIBCn_m. For details, see (1) PBDCn/JPBDC0 — Port Bi-Direction Control Register in Section 2.2.3.3, Pin Data Input/Output.

Access: The PIBCn register can be read/written in 16-bit units.
The JPIBC0 register can be read/written in 8-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PIBCn_15	PIBCn_14	PIBCn_13	PIBCn_12	PIBCn_11	PIBCn_10	PIBCn_9	PIBCn_8	PIBCn_7	PIBCn_6	PIBCn_5	PIBCn_4	PIBCn_3	PIBCn_2	PIBCn_1	PIBCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.12 PIBCn register contents

Bit Position	Bit Name	Function
15 to 0	PIBCn_[15:0]	Enables/disables the input buffer: 0: Input buffer is disabled. 1: Input buffer is enabled.

NOTES

1. When the input buffer is disabled, it does not consume flow-through current even if the pin level is in Hi-Z state. Thus, the pin does not need to be fixed to a high or low level externally.
2. The control bits of the JTAG port input buffer control register (JPIBC0) are JPIBC0_[7:0].
3. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.
4. When the S/W I/O control alternative input mode (PMCn_m = 1 and PMn_m = 1), PIBCn_m bit should be set to "0".

CAUTION

The setting of this register is not effective in bidirectional mode (PBDCn.PBDCn_m = 1).

(7) PFCn/JPFC0 — Port Function Control Register

This register, together with the PFCEn and PFCAEn registers, specifies an alternative function of the pins.

Some alternative functions require direct I/O control of pin Pn_m. For such alternative functions PIPCn.PIPCn_m must be set to 1 as well.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: The PFCn register can be read/written in 16-bit units.
The JPFC0 register can be read/written in 8-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCn_15	PFCn_14	PFCn_13	PFCn_12	PFCn_11	PFCn_10	PFCn_9	PFCn_8	PFCn_7	PFCn_6	PFCn_5	PFCn_4	PFCn_3	PFCn_2	PFCn_1	PFCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.13 PFCn register contents

Bit Position	Bit Name	Function
15 to 0	PFCn_[15:0]	Specifies an alternative function of a pin. See Table 2.4, Outline of Alternative Mode Selection (PMCn.PMCn_m = 1).

NOTES

1. The control bits of the JTAG port function control register (JPFC0) are JPFC0_[7:0].
2. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

CAUTION

With this product, the I/O of some functions is multiplexed in two pins, but only either one can be used as a pin function. Setting the same pin function in two pins is prohibited.

For example, if P0_1 pin is used as INTP0, JP0_0 pin cannot be used as INTP0. In this case, JP0_0 pin must be configured as the pin function other than INTP0.

(8) PFCEn / JPFCE0 — Port Function Control Expansion Register

This register, together with the PFCn and PFCAEn registers, specifies an alternative function of the pins.

Some alternative functions directly controls input/output of pin Pn_m. For such alternative functions PIPCn.PIPCn_m must be set to 1 as well.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: The PFCEn register can be read/written in 16-bit units.
The JPFCE0 register can be read/written in 8-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCEn_15	PFCEn_14	PFCEn_13	PFCEn_12	PFCEn_11	PFCEn_10	PFCEn_9	PFCEn_8	PFCEn_7	PFCEn_6	PFCEn_5	PFCEn_4	PFCEn_3	PFCEn_2	PFCEn_1	PFCEn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.14 PFCEn register contents

Bit Position	Bit Name	Function
15 to 0	PFCEn_[15:0]	Specifies an alternative function of a pin. See Table 2.4, Outline of Alternative Mode Selection (PMCn.PMCn_m = 1) for details.

Table 2.15 Setting alternative functions

PFCAEn_m	PFCEn_m	PFCn_m	PMn_m	Function
0	0	0	1	1st alternative function / Input
			0	1st alternative function / Output
		1	1	2nd alternative function / Input
			0	2nd alternative function / Output
	1	0	1	3rd alternative function / Input
			0	3rd alternative function / Output
		1	1	4th alternative function / Input
			0	4th alternative function / Output
1	0	0	1	5th alternative function / Input
			0	5th alternative function / Output
		1	1	6th alternative function / Input
			0	6th alternative function / Output

NOTES

1. The control bits of the JTAG port function control expansion register (JPFCE0) are JPFCE0_[7:0].
2. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

CAUTION

With this product, the I/O of some functions is multiplexed in two pins, but only either one can be used as a pin function. Setting the same pin function in two pins is prohibited.

For example, if P0_1 pin is used as INTP0, JP0_0 pin cannot be used as INTP0. In this case, JP0_0 pin must be configured as the pin function other than INTP0.

(9) PFCAEn — Port Function Control Additional Expansion Register

This register, together with the PFCn and PFCEn registers, specifies an alternative function of the pins.

Some alternative functions directly controls input/output of pin Pn_m. For such alternative functions PIPCn.PIPCn_m must be set to 1 as well.

For other alternative functions, input/output must be specified by PMn.PMn_m.

Access: The PFCEn register can be read/written in 16-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PFCAEn_15	PFCAEn_14	PFCAEn_13	PFCAEn_12	PFCAEn_11	PFCAEn_10	PFCAEn_9	PFCAEn_8	PFCAEn_7	PFCAEn_6	PFCAEn_5	PFCAEn_4	PFCAEn_3	PFCAEn_2	PFCAEn_1	PFCAEn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.16 PFCAEn register contents

Bit Position	Bit Name	Function
15 to 0	PFCAEn_[15:0]	Specifies an alternative function of a pin. See Table 2.4, Outline of Alternative Mode Selection (PMCn.PMCn_m = 1) for details.

2.2.3.3 Pin Data Input/Output

(1) PBDCn/JPBDC0 — Port Bi-Direction Control Register

This register enables the input buffer and sets the port to bi-direction mode. In bi-direction mode, PPRn.PPRn_m can read the level of the Pn_m pin.

The Pn_m pin level is read via PPRn.PPRn_m in bidirectional mode.

- Alternative output level loopback function
  When the Pn_m pin is used as the alternative output function, the actual pin output level based on the alternative output function can be looped back to the alternative input side by setting PBDCn.PBDCn_m = 1 and PIBCn.PIBCn_m= 0. For example, the pin output level based on the first alternative function can be looped back to the same alternative input side. Also the pin output level can be read via PPRn.PPRn_m.
- Pin output level read function
  When the Pn_m pin is used as the general output port function or the alternative output function, the actual pin output level can be read via PPRn.PPRn_m by setting PBDCn.PBDCn_m = 1 and PIBCn.PIBCn_m = 1. Under this setting, the pin output level will never be looped back to the alternative input side even in alternative output mode.

Access: The PBDCn register can be read/written in 16-bit units.
The JPBDC0 register can be read/written in 8-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PBDCn_15	PBDCn_14	PBDCn_13	PBDCn_12	PBDCn_11	PBDCn_10	PBDCn_9	PBDCn_8	PBDCn_7	PBDCn_6	PBDCn_5	PBDCn_4	PBDCn_3	PBDCn_2	PBDCn_1	PBDCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.17 PBDCn register contents

Bit Position	Bit Name	Function
15 to 0	PBDCn_[15:0]	Enables/disables bi-direction mode of the corresponding pin. 0: Bi-direction mode disabled 1: Bi-direction mode enabled

CAUTION

When the Pn_m port is used as an alternative output function (PMCn.PMCn_m = 1, PMn.PMn_m = 0), if bidirectional mode is enabled (PBDCn.PBDCn_m = 1), the level of the Pn_m pin can be read with PPRn.PPRn_m.

However, the output of the alternative output function is input to the alternative input function of the same pin (the alternative input function that is set with PFCn.PFCn_m, PFCEn.PFCEn_m and PFCAEn_m). If this alternative input function is used in another pin, note that the output of the alternative output function and alternative input in other pins are input internally (OR input).

NOTES

1. The control bits of the JTAG port bi-direction control register (JPBDC0) are JPBDC0_[7:0].
2. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

(2) PPRn/JPPR0 — Port Pin Read Register

This register reflects an actual Pn_m pin level, a Pn.Pn_m bit value, or an output level of the alternative function. The value to be read depends on various control settings as described in **Table 2.5, PPRn_m Read Values**.

Access: The PPRn register can be read/written in 16-bit units.
The JPPR0 register can be read/written in 8-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PPRn_15	PPRn_14	PPRn_13	PPRn_12	PPRn_11	PPRn_10	PPRn_9	PPRn_8	PPRn_7	PPRn_6	PPRn_5	PPRn_4	PPRn_3	PPRn_2	PPRn_1	PPRn_0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 2.18 PPRn register contents

Bit Position	Bit Name	Function
15 to 0	PPRn_[15:0]	Pin Pn_m, Pn.Pn_m value or alternative function output.

NOTES

1. For details about the read value of the PPRn register, see Section 2.2.2.3, Pin Data Input/Output.
2. The control bits of the JTAG port pin read register (JPPR0) are JPPR0_[7:0].
3. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

(3) Pn/JP0 — Port Register

This register sets and holds the Pn.Pn_m data to be output via the related Pn_m port in output port mode (PMCn.PMCn_m = 0 and PMn.PMn_m = 0).

Access: The Pn register can be read/written in 16-bit units.
The JP0 register can be read/written in 8-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Pn_15	Pn_14	Pn_13	Pn_12	Pn_11	Pn_10	Pn_9	Pn_8	Pn_7	Pn_6	Pn_5	Pn_4	Pn_3	Pn_2	Pn_1	Pn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.19 Pn register contents

Bit Position	Bit Name	Function
15 to 0	Pn_[15:0]	Sets the output level of pin m (m = 0 to 15): 0: Low level output 1: High level output

NOTE

- The bits of this register can be manipulated by various means; refer to the subsection, Write to the Pn Register in Section 2.2.2.3, Pin Data Input/Output.
- The control bits of the JTAG port register (JP0) are JP0_[7:0].
- For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

(4) PNOTn/JPNOT0 — Port NOT Register

This register enables inverting the Pn_m bit of the port register without directly writing to Pn.

Access: The PNOTn register can be read/written in 16-bit units.
The JPNOT0 register can be read/written in 8-bit units.
The read value is always 0000_H.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PNOTn_15	PNOTn_14	PNOTn_13	PNOTn_12	PNOTn_11	PNOTn_10	PNOTn_9	PNOTn_8	PNOTn_7	PNOTn_6	PNOTn_5	PNOTn_4	PNOTn_3	PNOTn_2	PNOTn_1	PNOTn_0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 2.20 PNOTn register contents

Bit Position	Bit Name	Function
15 to 0	PNOTn_[15:0]	Specifies if Pn.Pn_m is inverted or not: 0: Pn.Pn_m is not inverted ($\overline{Pn\_m} \rightarrow Pn\_m$) 1: Pn.Pn_m is inverted ($Pn\_m \rightarrow Pn\_m$)

NOTES

1. The control bits of the JTAG port not register (JPNOT0) are JPNOT0_[7:0].
2. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

(5) PSRn/JPSR0 — Port Set/Reset Register

This register provides an alternative method to write data to a bit in the Pn register.

The upper 16 bits of PSRn specify whether the data is written to the Pn.Pn_m bit specified in the lower 16 bits of PSRn.

Access: This register can be read/written in 32-bit units.
Bits 31 to 16 are always read as 0000_H. Reading bits 15 to 0 returns the value of the Pn register.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PSRn_31	PSRn_30	PSRn_29	PSRn_28	PSRn_27	PSRn_26	PSRn_25	PSRn_24	PSRn_23	PSRn_22	PSRn_21	PSRn_20	PSRn_19	PSRn_18	PSRn_17	PSRn_16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PSRn_15	PSRn_14	PSRn_13	PSRn_12	PSRn_11	PSRn_10	PSRn_9	PSRn_8	PSRn_7	PSRn_6	PSRn_5	PSRn_4	PSRn_3	PSRn_2	PSRn_1	PSRn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.21 PSRn register contents

Bit Position	Bit Name	Function
31 to 16	PSRn_[31:16]	Specifies whether the value of the corresponding lower bit of PSRn_m is written to Pn_m: 0: Pn_m does not depend on PSRn_m. 1: The value of Pn_m is the same as that of PSRn_m. Example: When PSRn.PSRn_31 = 1, the value of bit PSRn.PSRn_15 is written to bit Pn.Pn15.
15 to 0	PSRn_[15:0]	Specifies the Pn_m value when the corresponding upper bit PSRn_(m+16) is 1: 0: Pn_m = 0 1: Pn_m = 1

NOTES

1. The control bits of the JTAG port set/reset register (JPSR0) are JPSR0_[31:0].
2. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

(6) PINVn/JPINV0 — Port Output Level Inversion Register

This register enables inverting the output level from a pin. It is effective when the pin is in output mode regardless of port output mode or alternative output mode.

Writing to this register is protected by a special sequence of instructions by using the protection command register PPCMDn.
For details, refer to **Section 4, Write-Protected Registers**.

Access: The PINVn register can be read/written in 16-bit units.
The JPINV0 register can be read/written in 8-bit units.
Writing to this register is protected by a specific instruction sequence. For details, see Section 2.2.3.5, Port Register Protection.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PINVn_15	PINVn_14	PINVn_13	PINVn_12	PINVn_11	PINVn_10	PINVn_9	PINVn_8	PINVn_7	PINVn_6	PINVn_5	PINVn_4	PINVn_3	PINVn_2	PINVn_1	PINVn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.22 PINVn register contents

Bit Position	Bit Name	Function
15 to 0	PINVn_[15:0]	Specifies whether the output level from a pin is inverted or not. 0: Pin output level is not inverted. 1: Pin output level is inverted.

NOTES

1. The control bits of the JTAG port output level inversion register (JPINV0) are JPINV0_[31:0].
2. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

2.2.3.4 Configuration of Electrical Characteristics

(1) PUn/JPU0 — Pull-Up Option Register

This register specifies whether an on-chip pull-up resistor is connected to an input pin.

Access: The PUn register can be read/written in 16-bit units.
The JPU0 register can be read/written in 8-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PUn_15	PUn_14	PUn_13	PUn_12	PUn_11	PUn_10	PUn_9	PUn_8	PUn_7	PUn_6	PUn_5	PUn_4	PUn_3	PUn_2	PUn_1	PUn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.23 PUn register contents

Bit Position	Bit Name	Function
15 to 0	PUn_[15:0]	Specifies whether an on-chip pull-up resistor is connected to the corresponding pin: 0: No on-chip pull-up resistor is connected. 1: On-chip pull-up resistor is connected.

NOTES

1. Do not set PUn.PUn_m = 1 and PDn.PDn_m = 1 to a single pin.
2. The on-chip pull-up resistor has no effect when the pin is operated in output mode.
3. The control bits of the JTAG pull-up option register (JPU0) are JPU0_[7:0].
4. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

(2) PDn/JPD0 — Pull-down option register

This register specifies whether an on-chip pull-down resistor is connected to an input pin.

Access: The PDn register can be read/written in 16-bit units.
The JPD0 register can be read/written in 8-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDn_15	PDn_14	PDn_13	PDn_12	PDn_11	PDn_10	PDn_9	PDn_8	PDn_7	PDn_6	PDn_5	PDn_4	PDn_3	PDn_2	PDn_1	PDn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.24 PDn register contents

Bit Position	Bit Name	Function
15 to 0	PDn_[15:0]	Specifies whether a on-chip pull-down resistor is connected to the corresponding pin: 0: No on-chip pull-down resistor is connected. 1: On-chip pull-down resistor is connected.

NOTES

1. Do not set PUn.PUn_m = 1 and PDn.PDn_m = 1 to a single pin.
2. The on-chip pull-down resistor has no effect when the pin is operated in output mode.
3. The control bits of the JTAG pull-down option register (JPD0) are JPD0_[7:0].
4. For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

(3) PODCn/JPODC0 — Port Open-drain Control Register

This register selects push-pull or open-drain as the output buffer function.

Writing to this register is protected by a special sequence of instructions by using the protection command register PPCMDn.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units. Writing to this register is protected by a specific instruction sequence. For details, see Section 2.2.3.5, Port Register Protection.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PODCn_15	PODCn_14	PODCn_13	PODCn_12	PODCn_11	PODCn_10	PODCn_9	PODCn_8	PODCn_7	PODCn_6	PODCn_5	PODCn_4	PODCn_3	PODCn_2	PODCn_1	PODCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.25 PODCn register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When written, write the initial value.
15 to 0	PODCn_[15:0]	Specify the output circuit characteristics of pin m (m = 0 to 15). 0: push-pull 1: N-ch open-drain

(4) PDSCn/JPDSC0 — Port drive strength control register

This register specifies the output drive strength of a port pin. This function is related to fast or slow mode (high or low drive strength) of an output buffer.

Writing to this register is protected by a special sequence of instructions by using the protection command register PPCMDn.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Updating of this register needs a correct write sequence using the PPCMDn register.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

CAUTION

The upper 16 bits are also intended for the inverted value of a port register write sequence that is to be protected. See Section 2.2.3.5, Port Register Protection **for details.**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDSCn_15	PDSCn_14	PDSCn_13	PDSCn_12	PDSCn_11	PDSCn_10	PDSCn_9	PDSCn_8	PDSCn_7	PDSCn_6	PDSCn_5	PDSCn_4	PDSCn_3	PDSCn_2	PDSCn_1	PDSCn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.26 PDSCn register contents

Bit Position	Bit Name	Function
15 to 0	PDSCn_[15:0]	Specifies the output buffer characteristics of pin m (m = 0 to 15). 0: low drivability (pins supplied by ISMVCC) 1: high drivability (pins supplied by ISMVCC) 0: slow (other pins of relevant port groups) 1: fast (other pins of relevant port groups)

NOTE

For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

PDSC21 have to be set 0000 $03FF_H$ in D1M2(H).

NOTE

For D1M2 and D1M2H the initial value of the PDSC21 register have to be changed to PDSC21 = 0x03FF to select the output buffer type for the port P21.

- P21_[9:0] - HS Port Buffer for SFMA usage
- P21_[12:10] - MLB Port Buffer for MLBB usage

Because of this the use of the initial value for PDSC21 in D1M2 and D1M2(H) is restricted.

(5) PISn — Port Input Buffer Selection Register

This register specifies the input buffer characteristics together with the port input buffer selection advanced register PISAn.

Access: This register can be read/written in 16-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PISn_15	PISn_14	PISn_13	PISn_12	PISn_11	PISn_10	PISn_9	PISn_8	PISn_7	PISn_6	PISn_5	PISn_4	PISn_3	PISn_2	PISn_1	PISn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.27 PISn register contents

Bit Position	Bit Name	Function
15 to 0	PISn_[15:0]	Specifies the input buffer characteristics together with the PISAn register. See Section 2.2.5.4, Input buffer characteristics for details.

(6) PISAn — Port Input Buffer Selection Advanced Register

This register specifies the input buffer characteristics together with the port input buffer selection register PISn.

Access: This register can be read/written in 16-bit units.

Address: See Table 2.6, Registers for Port Group Configuration.

Initial value: See Section 2.3.1, Port Registers.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PISAn_15	PISAn_14	PISAn_13	PISAn_12	PISAn_11	PISAn_10	PISAn_9	PISAn_8	PISAn_7	PISAn_6	PISAn_5	PISAn_4	PISAn_3	PISAn_2	PISAn_1	PISAn_0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.28 PISAn register contents

Bit Position	Bit Name	Function
15 to 0	PISAn_[15:0]	Specifies the input buffer characteristics together with the PISn register. See Section 2.2.5.4, Input buffer characteristics for details.

2.2.3.5 Port Register Protection

(1) Protection registers

For details about the registers for the port register protection refer to Section 4, Write-Protected Registers.

(2) Port Register to be Protected

- Port Output Level Inversion Register (PINVn, JPINV0)
- Port Open-drain Control Register (PODCn, JPODC0)
- Port Drive Strength Control Register (PDSCn, JPDSC0)

(3) Write Sequence of the Port Register to be Protected

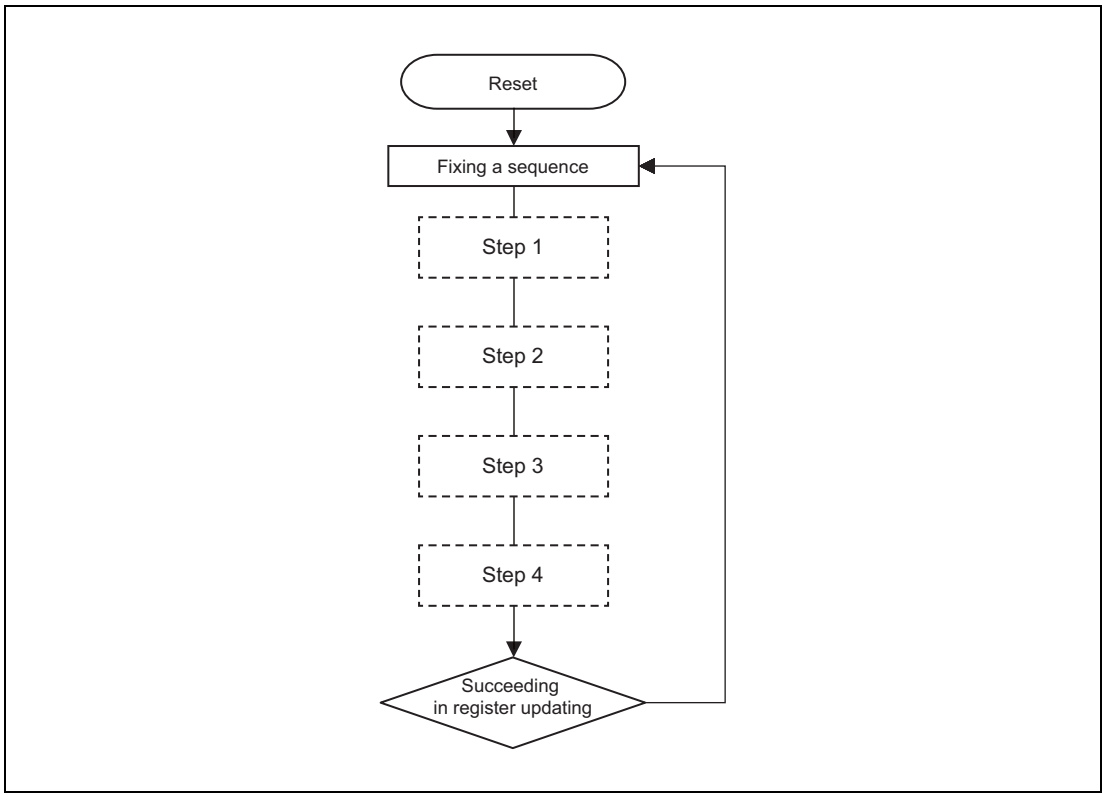


Figure 2.8 Write Sequence of the Port Register to be Protected

Step 1: Write $A5_H$ to the PPCMD register to initialize a write sequence.

Step 2: Write data to the register to be protected in accordance with each register's size (not updated).

Step 3: Write inverted data to the same register to be protected in accordance with each register's size (not updated).

Step 4: Write data to the same register to be protected in accordance with each register's size again (successfully updated). Verify successful write of the desired value to the protected register by reading the protected register written to.

2.2.3.6 Pin-unit Register

(1) Port Control Register (PCRn_m)

Each register of a port group can be accessed via this register and a PCR register can set all functions of a single pin. For example, setting bit 6 of the PCRn_m register to 1 sets bit m of the PMCn register to 1 also. Although writing to the PDSCn, PODCn and PINVn registers are protected, an access to them via the PCRn_m register does not require a protection release sequence.

CAUTIONS

1. **Multiple bits in the PCRn_m register can be batch-set within the ranges described in Section 2.2.4, Example of Port Configuration Flow, Section (2), Individual Setting.**
2. **If the bits are batch-set out of the range for setting of PCRn_m, the pin may output an unexpected signal level.**

Access: This register can be read/written in 32-bit units.

Initial value: See Section 2.3.1, Port Registers.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	PINV	—	PODC	—	—	—	PDSC	—	PISA	—	PIS	PU	PD	PBDC	PIBC
R/W	R	R/W	R	R/W	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	P	—	—	—	PPR	—	PMC	PIPC	PM	—	PFCAE	PFCE	PFC
R/W	R	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 2.29 PCRn_m register contents (1/2)

Bit Position	Bit Name	Function
31	—	Reserved
30	PINV	Same function as bit m of the PINVn register
29	—	Reserved
28	PODC	Same function as bit m of the PODCn register
27 to 25	—	Reserved
24	PDSC	Same function as bit m of the PDSCn register
23	—	Reserved
22	PISA	Same function as bit m of the PISAn register
21	—	Reserved
20	PIS	Same function as bit m of the PISn register
19	PU	Same function as bit m of the PUn register
18	PD	Same function as bit m of the PDn register
17	PBDC	Same function as bit m of the PBDCn register
16	PIBC	Same function as bit m of the PIBCn register
15 to 13	—	Reserved
12	P	Same function as bit m of the Pn register
11 to 9	—	Reserved

Table 2.29 PCRn_m register contents (2/2)

Bit Position	Bit Name	Function
8	PPR	The value of bit m of the PPRn register can be read.
7	—	Reserved
6	PMC	Same function as bit m of the PMCn register
5	PIPC	Same function as bit m of the PIPCn register
4	PM	Same function as bit m of the PMn register
3	—	Reserved
2	PFCAE	Same function as bit m of the PFCAEn register
1	PFCE	Same function as bit m of the PFCEn register
0	PFC	Same function as bit m of the PFCn register

NOTE

For details about the relevant port groups and bits, see Section 2.3.1, Port Registers.

2.2.4 Example of Port Configuration Flow

The followings are examples of the port configuration flow.

CAUTION

Even if a port is used in S/W I/O control alternative mode, it may be temporarily switched to alternative input mode in the following port configuration flows. This may occur during the period from setting of PMCn_m = 1 to that of PMn_m = 0.

(1) Batch Setting

The following shows an example to collectively set a port group.

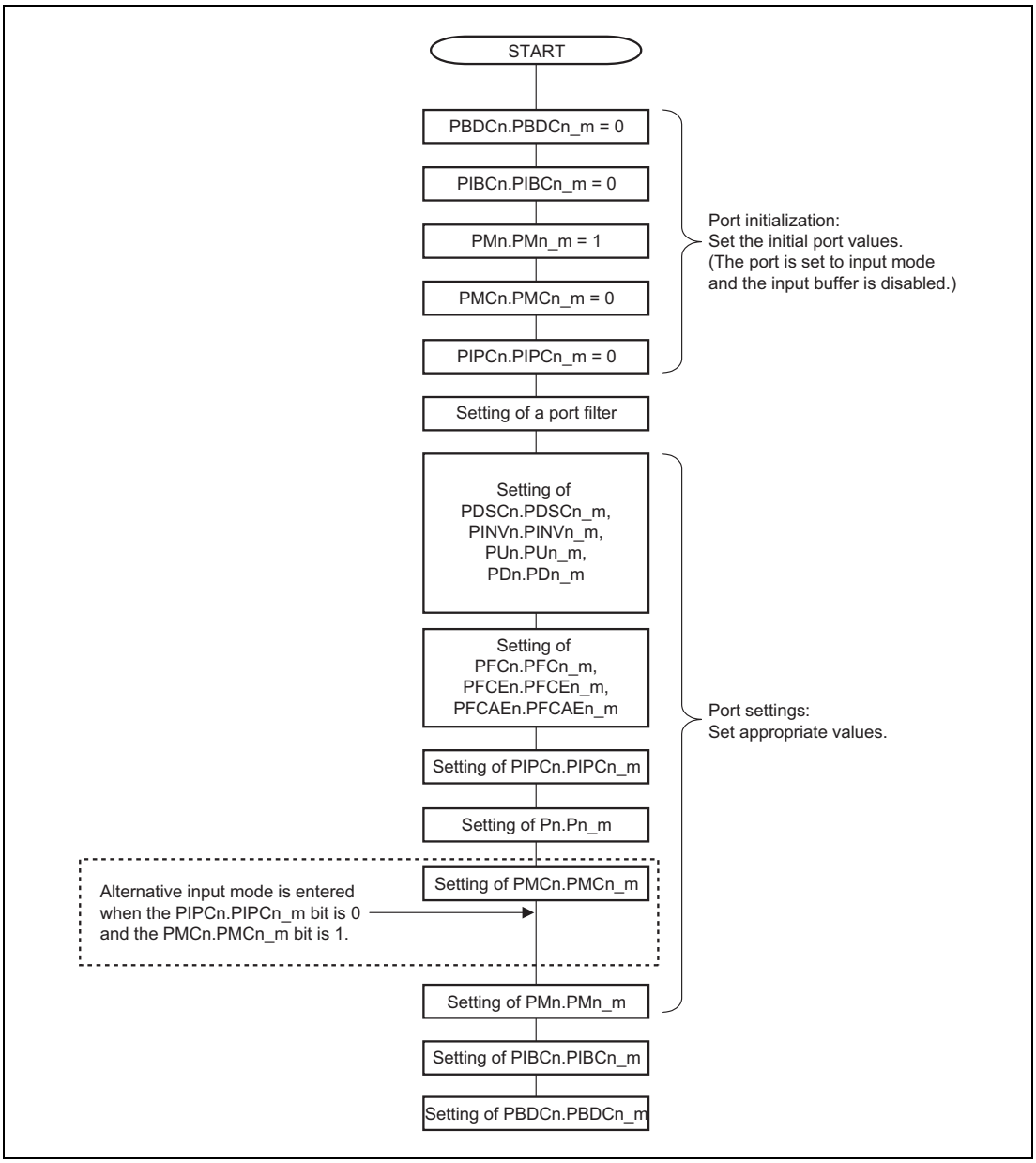


Figure 2.9 Example of Port Configuration Flow (batch setting)

(2) Individual Setting

The following shows an example to individually set a port group.

And it is capable of setting simultaneous bits at once within the Settable range described below by using PCRn_m register.

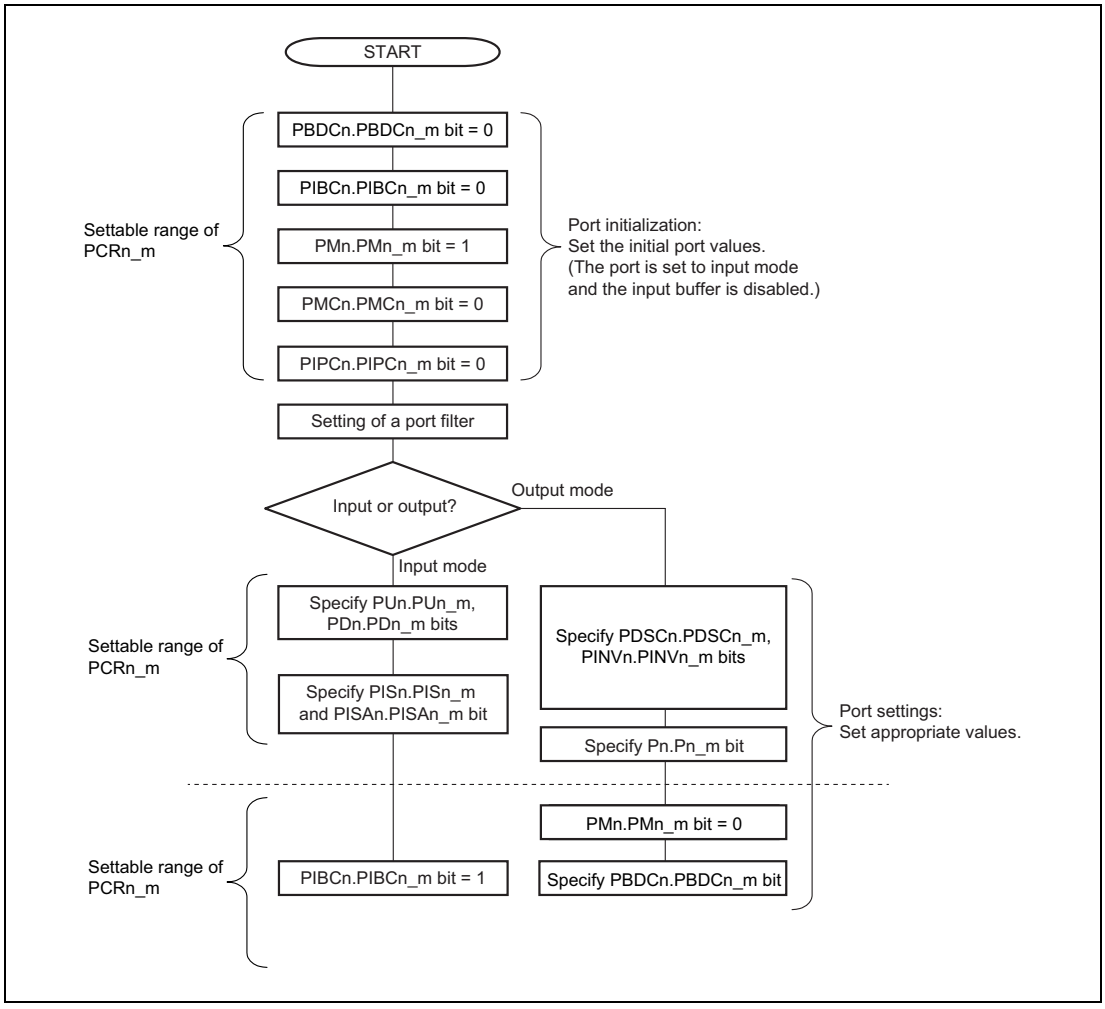


Figure 2.10 Example of Port Configuration Flow (in port mode)

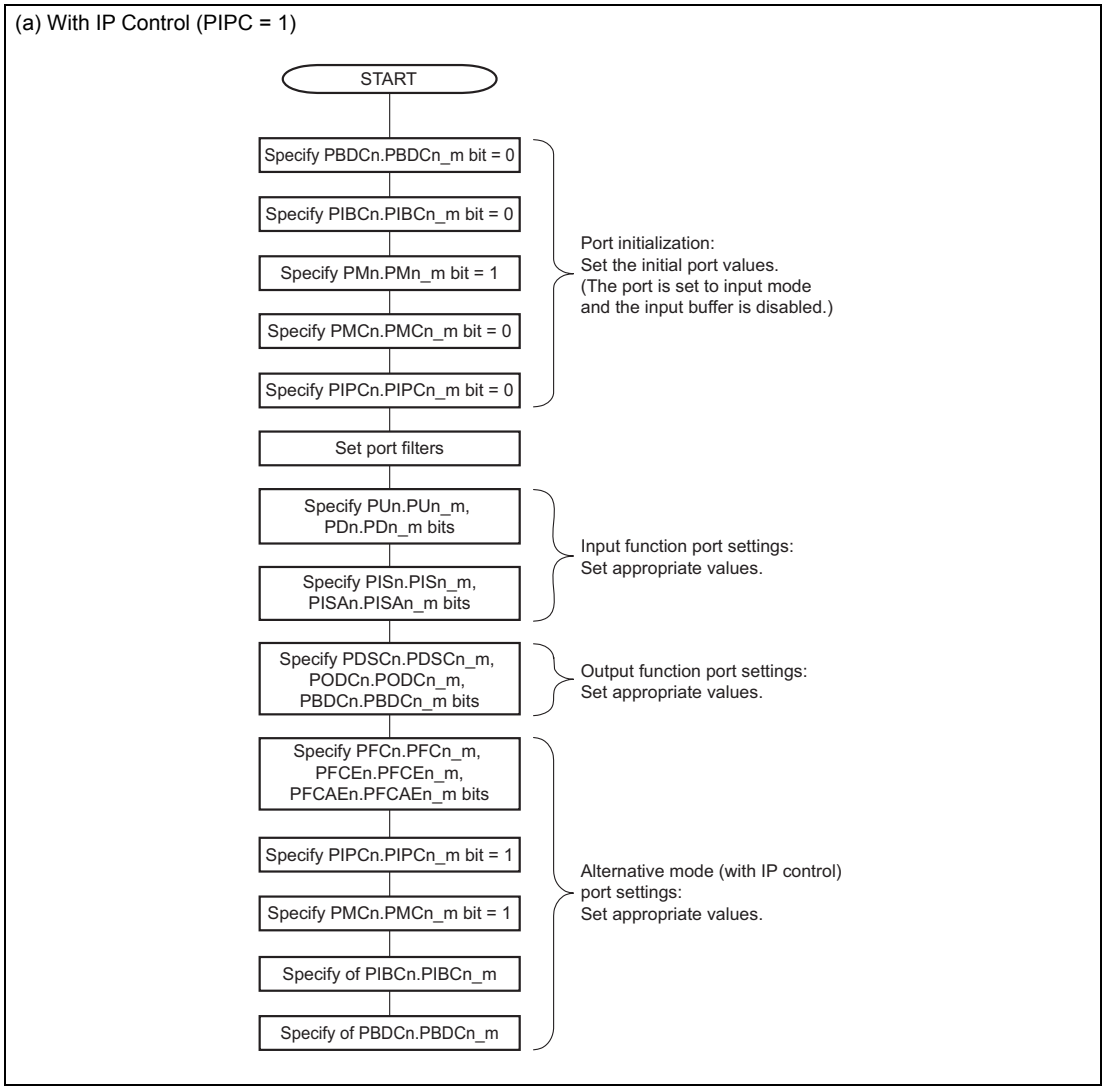


Figure 2.11 Example of Port Configuration Flow (in alternative mode) (1/2)

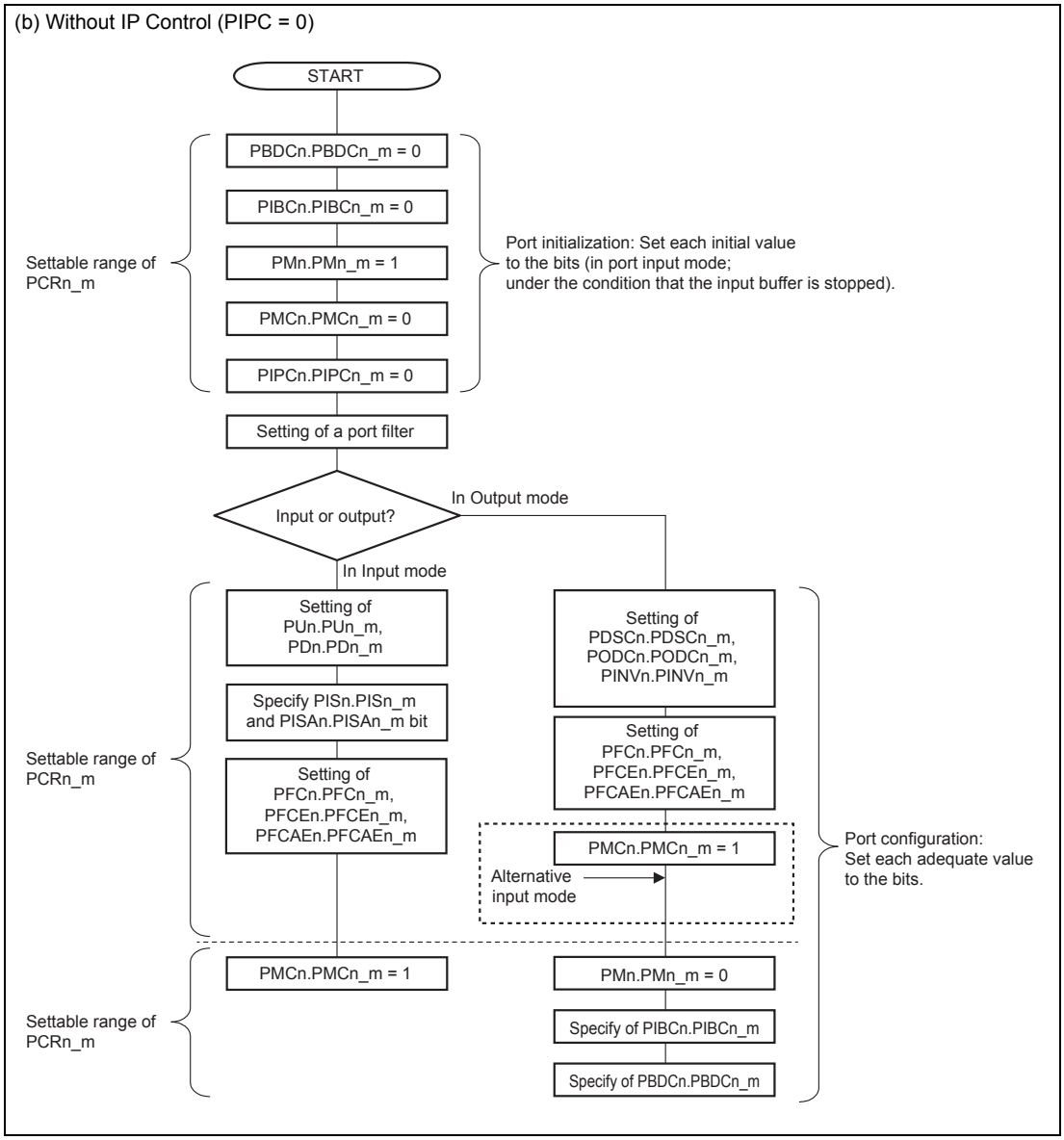


Figure 2.12 Example of Port Configuration Flow (in alternative mode) (2/2)

2.2.5 Functional Selection

2.2.5.1 Register Configuration in Use of the Alternative Function

When the pin alternative function is used, set PMCn_m = 1 and select the alternative numbers of PFCn_m, PFCEn_m and PFCAEn_m. In several peripheral functions, a single alternative I/O function is allocated to multiple pins. However, such an alternative function should not be enabled in multiple pins at the same time. For example, an external interrupt input, INTP7 is allocated to P0_6 and P44_11, but this alternative function can only be selected in either of them.

2.2.5.2 Alternative Function to be used in Direct I/O Control Alternative Mode

When the alternative functions described in **Table 2.30** are used, switch to direct I/O control alternative mode. When setting PIPCn_m = 1, the PMn_m value which has been set is ignored because the peripheral function enables or disables inputs and outputs of the buffer.

Table 2.30 List of the Pins which Require PIPC Register Setting (1/3)

Category	Pin Name
CSIG0	CSIG0SO
	CSIG0SC
CSIG1	CSIG1SO
	CSIG1SC
CSIG2	CSIG2SO
	CSIG2SC
CSIG3	CSIG3SO
	CSIG3SC
CSIH0	CSIH0SO
	CSIH0SC
CSIH1	CSIH1SO
	CSIH1SC
MLBB0	MLBB0SIG
	MLBB0DAT
ETNB0	ETNB0MDIO

Table 2.30 List of the Pins which Require PIPC Register Setting (2/3)

Category	Pin Name
ISM0	ISM11
	ISM12
	ISM13
	ISM14
	ISM21
	ISM22
	ISM23
	ISM24
	ISM31
	ISM32
	ISM33
	ISM34
	ISM41
	ISM42
	ISM43
	ISM44
	ISM51
	ISM52
	ISM53
	ISM54
	ISM61
	ISM62
	ISM63
	ISM64
SSIF0	SSIF0TXD
	SSIF0SCK
	SSIF0WS
SSIF1	SSIF1TXD
	SSIF1SCK
	SSIF1WS
SSIFn	SSIFACK
SFMA0	SFMA0O[3:0]0
	SFMA0O[3:0]1

Table 2.30 List of the Pins which Require PIPC Register Setting (3/3)

Category	Pin Name
LCBI0	LCBI0D0
	LCBI0D1
	LCBI0D2
	LCBI0D3
	LCBI0D4
	LCBI0D5
	LCBI0D6
	LCBI0D7
	LCBI0D8
	LCBI0D9
	LCBI0D10
	LCBI0D11
	LCBI0D12
	LCBI0D13
	LCBI0D14
	LCBI0D15
	LCBI0D16
	LCBI0D17
	$\overline{\text{LCBI0VSYNC}}$
	$\overline{\text{LCBI0HSYNC}}$
	$\overline{\text{LCBI0WRRW}}$
	$\overline{\text{LCBI0CS}}$
	LCBI0A0DE
	$\overline{\text{LCBI0RDE}}$
HYPB0	$\overline{\text{MDQ[7:0]}}$
	MDQS
NFMA0	NAND_D[7:0]

2.2.5.3 Register Setting in Use of an Analog Input Pin

Since ADCEnIm(SAR-A/D) for an analog input are always connected to the A/D converter, setting of a port register to select a pin function is not required.

2.2.5.4 Input buffer characteristics

The charateristic of most port input buffers is selectable via the PISn and PISAn registers.

The following tables show the selectable options.

Note that some port input buffers have a fixed charateristic, i.e. their default charateristic can not be changed.

(1) D1L1 input buffers characteristics

Table 2.31 D1L1 ports input buffer characteristics (1/3)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
JP0	JP0_0	TTL		Schmitt4	Schmitt1	Schmitt1
	JP0_1	—		—		Schmitt4
	JP0_2	TTL		Schmitt4	Schmitt1	Schmitt1
	JP0_3	TTL		Schmitt4		Schmitt4
	JP0_4	TTL		Schmitt4		Schmitt4
	JP0_5	—		—		Schmitt4
P0	P0_0	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_1	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_2	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_3	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_4	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_5	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_6	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_7	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_8	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_9	TTL		Schmitt4	Schmitt1	Schmitt1
P1	P1_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
P3	P3_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_9	CMOS1		Schmitt4	Schmitt1	Schmitt1

Table 2.31 D1L1 ports input buffer characteristics (2/3)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P10	P10_0	—		—		CMOS1
	P10_1	—		—		CMOS1
	P10_2	—		—		CMOS1
	P10_3	—		—		CMOS1
	P10_4	—		—		CMOS1
	P10_5	—		—		CMOS1
	P10_6	—		—		CMOS1
	P10_7	—		—		CMOS1
	P10_8	—		—		CMOS1
	P10_9	—		—		CMOS1
	P10_10	—		—		CMOS1
	P10_11	—		—		CMOS1
P11	P11_0	—		—		CMOS1
	P11_1	—		—		CMOS1
	P11_2	—		—		CMOS1
	P11_3	—		—		CMOS1
P16	P16_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
P17	P17_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_11	CMOS1		Schmitt4	Schmitt1	Schmitt1

Table 2.31 D1L1 ports input buffer characteristics (3/3)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P21	P21_0	—		CMOS1	CMOS1	CMOS1
	P21_1	—		CMOS1	CMOS1	CMOS1
	P21_2	—		CMOS1	CMOS1	CMOS1
	P21_3	—		CMOS1	CMOS1	CMOS1
	P21_4	—		CMOS1	CMOS1	CMOS1
	P21_5	—		CMOS1	CMOS1	CMOS1
P42	P42_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
P43	P43_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
P44	P44_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
P45	P45_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_12	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_13	CMOS1		Schmitt4	Schmitt1	Schmitt1

(2) D1L2(H) and D1M1(H) input buffers characteristics

Table 2.32 D1L2(H) and D1M1(H) ports input buffer characteristics (1/4)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
JP0	JP0_0	TTL		Schmitt4	Schmitt1	Schmitt1
	JP0_1	—		—		Schmitt4
	JP0_2	TTL		Schmitt4	Schmitt1	Schmitt1
	JP0_3	TTL		Schmitt4		Schmitt4
	JP0_4	TTL		Schmitt4		Schmitt4
	JP0_5	—		—		Schmitt4
P0	P0_0	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_1	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_2	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_3	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_4	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_5	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_6	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_7	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_8	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_9	TTL		Schmitt4	Schmitt1	Schmitt1
P1	P1_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
P3	P3_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_9	CMOS1		Schmitt4	Schmitt1	Schmitt1

Table 2.32 D1L2(H) and D1M1(H) ports input buffer characteristics (2/4)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P10	P10_0	—		—		CMOS1
	P10_1	—		—		CMOS1
	P10_2	—		—		CMOS1
	P10_3	—		—		CMOS1
	P10_4	—		—		CMOS1
	P10_5	—		—		CMOS1
	P10_6	—		—		CMOS1
	P10_7	—		—		CMOS1
	P10_8	—		—		CMOS1
	P10_9	—		—		CMOS1
	P10_10	—		—		CMOS1
	P10_11	—		—		CMOS1
P11	P11_0	—		—		CMOS1
	P11_1	—		—		CMOS1
	P11_2	—		—		CMOS1
	P11_3	—		—		CMOS1
P16	P16_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
P17	P17_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_11	CMOS1		Schmitt4	Schmitt1	Schmitt1

Table 2.32 D1L2(H) and D1M1(H) ports input buffer characteristics (3/4)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P21	P21_0	—		CMOS1		CMOS1
	P21_1	—		CMOS1		CMOS1
	P21_2	—		CMOS1		CMOS1
	P21_3	—		CMOS1		CMOS1
	P21_4	—		CMOS1		CMOS1
	P21_5	—		CMOS1		CMOS1
	P21_6	—		CMOS1		CMOS1
	P21_7	—		CMOS1		CMOS1
	P21_8	—		CMOS1		CMOS1
	P21_9	—		CMOS1		CMOS1
P42	P42_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_12	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_13	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_14	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_15	CMOS1		Schmitt4	Schmitt1	Schmitt1
P43	P43_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
P44	P44_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_11	CMOS1		Schmitt4	Schmitt1	Schmitt1

Table 2.32 D1L2(H) and D1M1(H) ports input buffer characteristics (4/4)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P45	P45_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_12	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_13	CMOS1		Schmitt4	Schmitt1	Schmitt1

(3) D1M2(H) input buffers characteristics

Table 2.33 D1M2(H) ports input buffer characteristics (1/6)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
JP0	JP0_0	TTL		Schmitt4	Schmitt1	Schmitt1
	JP0_1	—		—		Schmitt4
	JP0_2	TTL		Schmitt4	Schmitt1	Schmitt1
	JP0_3	TTL		Schmitt4		Schmitt4
	JP0_4	TTL		Schmitt4		Schmitt4
	JP0_5	—		—		Schmitt4
P0	P0_0	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_1	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_2	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_3	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_4	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_5	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_6	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_7	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_8	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_9	TTL		Schmitt4	Schmitt1	Schmitt1

Table 2.33 D1M2(H) ports input buffer characteristics (2/6)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P1	P1_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
P2	P2_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P2_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P2_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P2_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P2_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P2_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P2_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P2_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P2_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P2_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P2_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P2_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
P3	P3_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_12	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_13	CMOS1		Schmitt4	Schmitt1	Schmitt1

Table 2.33 D1M2(H) ports input buffer characteristics (3/6)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P10	P10_0	—		—		CMOS1
	P10_1	—		—		CMOS1
	P10_2	—		—		CMOS1
	P10_3	—		—		CMOS1
	P10_4	—		—		CMOS1
	P10_5	—		—		CMOS1
	P10_6	—		—		CMOS1
	P10_7	—		—		CMOS1
	P10_8	—		—		CMOS1
	P10_9	—		—		CMOS1
	P10_10	—		—		CMOS1
	P10_11	—		—		CMOS1
P11	P11_0	—		—		CMOS1
	P11_1	—		—		CMOS1
	P11_2	—		—		CMOS1
	P11_3	—		—		CMOS1
	P11_4	—		—		CMOS1
	P11_5	—		—		CMOS1
	P11_6	—		—		CMOS1
	P11_7	—		—		CMOS1
P16	P16_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_11	CMOS1		Schmitt4	Schmitt1	Schmitt1

Table 2.33 D1M2(H) ports input buffer characteristics (4/6)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P17	P17_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
P21	P21_0	TTL(MLB)*2		CMOS1		CMOS1
	P21_1	TTL(MLB)*2		CMOS1		CMOS1
	P21_2	TTL(MLB)*2		CMOS1		CMOS1
	P21_3	TTL(MLB)*2		CMOS1		CMOS1
	P21_4	TTL(MLB)*2		CMOS1		CMOS1
	P21_5	TTL(MLB)*2		CMOS1		CMOS1
	P21_6	TTL(MLB)*2		CMOS1		CMOS1
	P21_7	TTL(MLB)*2		CMOS1		CMOS1
	P21_8	TTL(MLB)*2		CMOS1		CMOS1
	P21_9	TTL(MLB)*2		CMOS1		CMOS1
	P21_10	TTL(MLB)*2		CMOS1		TTL(MLB)
	P21_11	TTL(MLB)*2		CMOS1		TTL(MLB)
	P21_12	TTL(MLB)*2		CMOS1		TTL(MLB)
P40	P40_0	—		—		CMOS1*1
	P40_1	—		—		CMOS1*1
	P40_2	—		—		CMOS1*1
	P40_3	—		—		CMOS1*1
	P40_4	—		—		CMOS1*1
	P40_5	—		—		CMOS1*1

Table 2.33 D1M2(H) ports input buffer characteristics (5/6)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P42	P42_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_12	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_13	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_14	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_15	CMOS1		Schmitt4	Schmitt1	Schmitt1
P43	P43_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_12	CMOS1		Schmitt4	Schmitt1	Schmitt1
P44	P44_0	—		—		CMOS1
	P44_1	—		—		CMOS1
	P44_2	—		—		CMOS1
	P44_3	—		—		CMOS1
	P44_4	—		—		CMOS1
	P44_5	—		—		CMOS1
	P44_6	—		—		CMOS1
	P44_7	—		—		CMOS1
	P44_8	—		—		CMOS1
	P44_9	—		—		CMOS1
	P44_10	—		—		CMOS1
	P44_11	—		—		CMOS1

Table 2.33 D1M2(H) ports input buffer characteristics (6/6)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P45	P45_0	—		—		CMOS1
	P45_1	—		—		CMOS1
	P45_2	—		—		CMOS1
	P45_3	—		—		CMOS1
	P45_4	—		—		CMOS1
	P45_5	—		—		CMOS1
	P45_6	—		—		CMOS1
	P45_7	—		—		CMOS1
	P45_8	—		—		CMOS1
	P45_9	—		—		CMOS1
	P45_10	—		—		CMOS1
	P45_11	—		—		CMOS1
	P45_12	—		—		CMOS1
	P45_13	—		—		CMOS1
P46	P46_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_12	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_13	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_14	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P46_15	CMOS1		Schmitt4	Schmitt1	Schmitt1
P47	P47_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P47_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P47_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P47_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P47_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P47_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P47_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P47_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P47_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P47_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P47_10	CMOS1		Schmitt4	Schmitt1	Schmitt1

Note 1. CMOS1 only for GPIO usage. MIPI Video Input Interface using MIPI CSI-2 input characteristics HS-RX and LP-RX (automatically selected).

Note 2. TTL(MLB) is only available for D1M2H devices.

(4) D1M1-V2, D1M1A input buffers characteristics

Table 2.34 D1M1-V2, D1M1A ports input buffer characteristics (1/4)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
JP0	JP0_0	TTL		Schmitt4	Schmitt1	Schmitt1
	JP0_1	—		—		Schmitt4
	JP0_2	TTL		Schmitt4	Schmitt1	Schmitt1
	JP0_3	TTL		Schmitt4		Schmitt4
	JP0_4	TTL		Schmitt4		Schmitt4
	JP0_5	—		—		Schmitt4
P0	P0_0	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_1	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_2	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_3	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_4	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_5	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_6	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_7	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_8	TTL		Schmitt4	Schmitt1	Schmitt1
	P0_9	TTL		Schmitt4	Schmitt1	Schmitt1
P1	P1_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P1_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
P3	P3_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P3_9	CMOS1		Schmitt4	Schmitt1	Schmitt1

Table 2.34 D1M1-V2, D1M1A ports input buffer characteristics (2/4)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P10	P10_0	—		—		CMOS1
	P10_1	—		—		CMOS1
	P10_2	—		—		CMOS1
	P10_3	—		—		CMOS1
	P10_4	—		—		CMOS1
	P10_5	—		—		CMOS1
	P10_6	—		—		CMOS1
	P10_7	—		—		CMOS1
	P10_8	—		—		CMOS1
	P10_9	—		—		CMOS1
	P10_10	—		—		CMOS1
	P10_11	—		—		CMOS1
P11	P11_0	—		—		CMOS1
	P11_1	—		—		CMOS1
	P11_2	—		—		CMOS1
	P11_3	—		—		CMOS1
P16	P16_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P16_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
P17	P17_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P17_11	CMOS1		Schmitt4	Schmitt1	Schmitt1

Table 2.34 D1M1-V2, D1M1A ports input buffer characteristics (3/4)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P21	P21_0	—		CMOS1		CMOS1
	P21_1	—		CMOS1		CMOS1
	P21_2	—		CMOS1		CMOS1
	P21_3	—		CMOS1		CMOS1
	P21_4	—		CMOS1		CMOS1
	P21_5	—		CMOS1		CMOS1
	P21_6	—		CMOS1		CMOS1
	P21_7	—		CMOS1		CMOS1
	P21_8	—		CMOS1		CMOS1
	P21_9	—		CMOS1		CMOS1
P22	P22_10	—		CMOS1		CMOS1
P42	P42_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_12	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_13	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_14	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P42_15	CMOS1		Schmitt4	Schmitt1	Schmitt1
P43	P43_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P43_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
P44	P44_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P44_11	CMOS1		Schmitt4	Schmitt1	Schmitt1

Table 2.34 D1M1-V2, D1M1A ports input buffer characteristics (4/4)

Port		PISAn_m = 1		PISAn_m = 0		Default
		PISn_m = 1	PISn_m = 0	PISn_m = 1	PISn_m = 0	
P45	P45_0	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_1	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_2	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_3	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_4	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_5	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_6	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_7	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_8	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_9	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_10	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_11	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_12	CMOS1		Schmitt4	Schmitt1	Schmitt1
	P45_13	CMOS1		Schmitt4	Schmitt1	Schmitt1

For the electrical specification of the different input buffer types refer to the Data Sheet.

2.2.5.5 Output Buffer Control (PDSC)

For several ports the drive strength of an output buffer can be selected as

- PDSCn.PDSCn_m = 0: slow
- PDSCn.PDSCn_m = 1: fast

The applicable output ports can be identified by the exixtence of a PDSCn.PDSCn_m control bit in the port control register lists in Section 2.3.1, Port Registers.

For the electrical specification of the different output buffer types refer to the Data Sheet.

2.2.5.6 Output buffer drive strength control register (DSCTRL) (D1M1-V2, D1M1A only)

The output buffers of some ports feature a special output drive strength control register. This register can control the drive capability of the output buffer.

Access: This register can be read/written in 32-bit units.

Address: FFC0 6054_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	P22_10_DS[1:0]		P21_9_1_DS[1:0]		P21_0_DS[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SDRDSA[1:0]		SDRDSD3[1:0]		SDRDSD2[1:0]		SDRDSD1[1:0]		SDRDSD0[1:0]		SDRDSC[1:0]		—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 2.35 DSCTRL register contents (1/2)

Bit Position	Bit Name	Function
31 to 22	—	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	P22_10_ DS[1:0]	Output drive strength of P22_10 00_B: target impedance 25 Ω 01_B: target impedance 33 Ω 10_B: target impedance 50 Ω 11_B: target impedance 100 Ω
19, 18	P21_9_1_ DS[1:0]	Output drive strength of P21_[9:1] 00_B: target impedance 25 Ω 01_B: target impedance 33 Ω 10_B: target impedance 50 Ω 11_B: target impedance 100 Ω
17, 16	P21_0_ DS[1:0]	Output drive strength of P21_0 00_B: target impedance 25 Ω 01_B: target impedance 33 Ω 10_B: target impedance 50 Ω 11_B: target impedance 100 Ω
15, 14	—	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	SDRDSA[1:0]	Output drive strength of SDRAA[12:0], SDRABA[1:0], SDRACSZ, SDRARASZ, SDRACASZ, SDRAWEZ (D1M1A only) 00_B: target impedance 25 Ω 01_B: target impedance 33 Ω 10_B: target impedance 50 Ω 11_B: target impedance 100 Ω
11, 10	SDRDSD3[1:0]	Output drive strength of SDRADQ[31:24], SDRADM3 (D1M1A only) 00_B: target impedance 25 Ω 01_B: target impedance 33 Ω 10_B: target impedance 50 Ω 11_B: target impedance 100 Ω
9, 8	SDRDSD2[1:0]	Output drive strength of SDRADQ[23:16], SDRADM2 (D1M1A only) 00_B: target impedance 25 Ω 01_B: target impedance 33 Ω 10_B: target impedance 50 Ω 11_B: target impedance 100 Ω

Table 2.35 DSCTRL register contents (2/2)

Bit Position	Bit Name	Function
7, 6	SDRDSD1[1:0]	Output drive strength of SDRADQ[15:8], SDRADM1 (D1M1A only) 00_B: target impedance 25 Ω 01_B: target impedance 33 Ω 10_B: target impedance 50 Ω 11_B: target impedance 100 Ω
5, 4	SDRDSD0[1:0]	Output drive strength of SDRADQ[7:0], SDRADM0 (D1M1A only) 00_B: target impedance 25 Ω 01_B: target impedance 33 Ω 10_B: target impedance 50 Ω 11_B: target impedance 100 Ω
3, 2	SDRDSC[1:0]	Output drive strength of SDRACK, SDRACKE (D1M1A only) 00_B: target impedance 25 Ω 01_B: target impedance 33 Ω 10_B: target impedance 50 Ω 11_B: target impedance 100 Ω
1, 0	—	Reserved These bits are always read as 0. The write value should always be 0.

NOTE

In the header files the module name of the above register is defined as:

SELB.

2.3 Organization of Port Groups

2.3.1 Port Registers

The tables in this subsection show detailed bitmaps of control registers in each port group. In the bitmap field, "√" means an effective bit and "—" means a reserved one. Reserved areas are always read as initial values. The write value also should be an initial value.

2.3.1.1 List of D1L1, D1L2 Port Registers

(1) Port 0 D1L1, D1L2 registers

Table 2.36 Port group 0 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P0	PMC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMCSR0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC0	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	—	
	PM0	R/W	$FFFF_H$	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMSR0	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFCE0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PBDC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPR0	R	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	P0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PSR0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT0	W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PINV0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PU0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PD0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PDSC0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PISA0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPCMD0	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS0	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(2) Port 1 D1L1, D1L2 registers

Table 2.37 Port group 1 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P1	PMC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PMCSR1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	Upper 16 bits
	PIPC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	
	PM1	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PMSR1	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	Upper 16 bits
	PIBC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PFC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PFCE1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PBDC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PPR1	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	P1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PSR1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	Upper 16 bits
	PNOT1	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PINV1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PU1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PD1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PDSC1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PISA1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	
	PPCMD1	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS1	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(3) Port 3 D1L1, D1L2 registers

Table 2.38 Port group 3 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P3	PMC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PMCSR3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	Upper 16 bits
	PIPC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	—	—	—	
	PM3	R/W	$FFFF_H$	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PMSR3	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	Upper 16 bits
	PIBC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PFC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PFCE3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PBDC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PPR3	R	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	P3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PSR3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	Upper 16 bits
	PNOT3	W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PINV3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PU3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PD3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PDSC3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PISA3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PPCMD3	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS3	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(4) Port 10 D1L1, D1L2 registers

Table 2.39 Port group 10 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P10	PMC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR10	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PM10	R/W	$FFFF_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR10	R/W	$0000\ FFFF_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR10	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR10	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT10	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PODC10	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD10	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS10	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(5) Port 11 D1L1, D1L2 registers

Table 2.40 Port group 11 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P11	PMC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PMCSR11	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Upper 16 bits
	PM11	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PMSR11	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Upper 16 bits
	PIBC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PFC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PBDC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PPR11	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	P11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PSR11	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Upper 16 bits
	PNOT11	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PINV11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PODC11	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD11	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS11	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(6) Port 16 D1L1, D1L2 registers

Table 2.41 Port group 16 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P16	PMC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR16	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PM16	R/W	$F000_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR16	R/W	$0000\ F000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR16	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR16	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT16	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC16	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC16	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD16	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS16	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(7) Port 17 D1L1, D1L2 registers

Table 2.42 Port group 17 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P17	PMC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PM17	R/W	$F000_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR17	R/W	$0000\ F000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PBDC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR17	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT17	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD17	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS17	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(8) Port 21 D1L1, D1L2 registers

Table 2.43 Port group 21 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P21	PMC21	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PMCSR21	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PIPC21	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	—	—	
	PM21	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PMSR21	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PIBC21	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PFC21	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PBDC21	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PPR21	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	P21	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PSR21	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PNOT21	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PINV21	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PU21	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PD21	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PODC21	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD21	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS21	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(9) Port 42 D1L1, D1L2 registers

Table 2.44 Port group 42 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P42	PMC42	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PMCSR42	R/W	$0000\ 0000_H$	32	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	Lower 16 bits
					—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	Upper 16 bits
	PIPC42	R/W	0000_H	16	—	—	—	—	—	√	—	√	—	—	—	—	—	—	—	—	
	PM42	R/W	$FFFF_H$	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PMSR42	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	Lower 16 bits
					—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	Upper 16 bits
	PIBC42	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PFC42	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PFCE42	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PBDC42	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PPR42	R	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	P42	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PSR42	R/W	$0000\ 0000_H$	32	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	Lower 16 bits
					—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	Upper 16 bits
	PNOT42	W	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PINV42	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PU42	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PD42	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PDSC42	R/W	$0000\ 0000_H$	32	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC42	R/W	$0000\ 0000_H$	32	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS42	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PISA42	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	—	—	—	—	—	—	—	
	PPCMD42	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS42	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(10) Port 43 D1L1, D1L2 registers

Table 2.45 Port group 43 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P43	PMC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PMCSR43	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Upper 16 bits
	PIPC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PM43	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PMSR43	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Upper 16 bits
	PIBC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PFC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PFCE43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PBDC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PPR43	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	P43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PSR43	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Upper 16 bits
	PNOT43	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PINV43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PU43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PD43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PDSC43	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC43	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PISA43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PPCMD43	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS43	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(11) Port 44 D1L1, D1L2 registers

Table 2.46 Port group 44 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P44	PMC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PMCSR44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	Upper 16 bits
	PIPC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PM44	R/W	$FFFF_H$	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PMSR44	R/W	$0000\ FFFF_H$	32	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	Upper 16 bits
	PIBC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PFC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PFCE44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PBDC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PPR44	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	P44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PSR44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	Upper 16 bits
	PNOT44	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PINV44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PU44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PD44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PDSC44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PISA44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PPCMD44	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS44	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(12) Port 45 D1L1, D1L2 registers

Table 2.47 Port group 45 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P45	PMC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR45	R/W	0000 0000_H	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PM45	R/W	$FFFF_H$	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR45	R/W	0000 $FFFF_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR45	R	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	P45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR45	R/W	0000 0000_H	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT45	W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PU45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PD45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC45	R/W	0000 0000_H	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC45	R/W	0000 0000_H	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD45	W	0000 0000_H	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS45	R	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(13) JTAG port JP0 D1L1, D1L2 registers

Table 2.48 JTAG port group JP0 D1L1, D1L2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
JP0	JPMC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPMCSR0	R/W	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	JPM0	R/W	FF_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPMSR0	R/W	0000 $00FF_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	JPIBC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPFC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√	
	JPFCE0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√	
	JPBDC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPPR0	R	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JP0	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPSR0	R/W	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	JPNOT0	W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPINV0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPU0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPD0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPDSC0	R/W	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	√	—	—	—	√	—	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	JPODC0	R/W	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	JPIS0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	
	JPISA0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	—	√	√	√	—	√	
	JPPCMD0	W	0000 0000_H	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	JPPROTS0	R	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

2.3.1.2 List of D1L2H, D1M1(H) Port Registers

(1) Port 0 D1L2H, D1M1(H) registers

Table 2.49 Port group 0 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P0	PMC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMCSR0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC0	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	—	
	PM0	R/W	$FFFF_H$	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMSR0	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFCE0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PBDC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPR0	R	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	P0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PSR0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT0	W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PINV0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PU0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PD0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PDSC0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PISA0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPCMD0	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS0	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(2) Port 1 D1L2H, D1M1(H) registers

Table 2.50 Port group 1 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P1	PMC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PMCSR1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PIPC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	—	√	√	—	√	
	PM1	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PMSR1	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PIBC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PFC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PFCE1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PBDC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PPR1	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	P1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PSR1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PNOT1	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PINV1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PU1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PD1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PDSC1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PISA1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PPCMD1	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS1	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(3) Port 3 D1L2H, D1M1(H) registers

Table 2.51 Port group 3 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P3	PMC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMCSR3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	—	—	√ *1	
	PM3	R/W	$FFFF_H$	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMSR3	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFCE3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PBDC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPR3	R	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	P3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PSR3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT3	W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PINV3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PU3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PD3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PDSC3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PISA3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPCMD3	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS3	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

Note 1. Not available in D1L2H.

(4) Port 10 D1L2H, D1M1(H) registers

Table 2.52 Port group 10 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P10	PMC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR10	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PM10	R/W	$FFFF_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR10	R/W	$0000\ FFFF_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR10	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR10	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT10	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PODC10	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD10	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS10	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(5) Port 11 D1L2H, D1M1(H) registers

Table 2.53 Port group 11 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P11	PMC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PMCSR11	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Upper 16 bits
	PM11	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PMSR11	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Upper 16 bits
	PIBC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PFC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PBDC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PPR11	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	P11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PSR11	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Upper 16 bits
	PNOT11	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PINV11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PODC11	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD11	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS11	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(6) Port 16 D1L2H, D1M1(H) registers

Table 2.54 Port group 16 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P16	PMC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR16	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PM16	R/W	$F000_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR16	R/W	$0000\ F000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR16	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR16	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT16	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC16	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC16	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD16	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS16	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(7) Port 17 D1L2H, D1M1(H) registers

Table 2.55 Port group 17 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P17	PMC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PM17	R/W	$F000_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR17	R/W	$0000\ F000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PBDC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR17	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT17	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD17	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS17	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(8) Port 21 D1L2H, D1M1(H) registers

Table 2.56 Port group 21 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P21	PMC21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMCSR21	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PM21	R/W	$FFFF_H$	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMSR21	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFC21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFCE21	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	PBDC21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPR21	R	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	P21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PSR21	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT21	W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PINV21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PU21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PD21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PODC21	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD21	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS21	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(9) Port 42 D1L2H, D1M1(H) registers

Table 2.57 Port group 42 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P42	PMC42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR42	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC42	R/W	0000_H	16	—	—	—	—	—	√	—	√	—	—	√	—	√	—	—	—	
	PM42	R/W	$FFFF_H$	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR42	R/W	$0000\ FFFF_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR42	R	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	P42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR42	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT42	W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PU42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PD42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC42	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC42	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD42	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS42	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(10) Port 43 D1L2H, D1M1(H) registers

Table 2.58 Port group 43 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P43	PMC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PMCSR43	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Upper 16 bits
	PIPC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PM43	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PMSR43	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Upper 16 bits
	PIBC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PFC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PFCE43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PBDC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PPR43	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	P43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PSR43	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Upper 16 bits
	PNOT43	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PINV43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PU43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PD43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PDSC43	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC43	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PISA43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PPCMD43	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS43	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(11) Port 44 D1L2H, D1M1(H) registers

Table 2.59 Port group 44 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P44	PMC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PM44	R/W	$FFFF_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR44	R/W	$0000\ FFFF_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√ *1	√	√	√	√	√	
	PBDC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR44	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT44	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PU44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PD44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD44	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS44	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

Note 1. Not available in D1L2H.

(12) Port 45 D1L2H, D1M1(H) registers

Table 2.60 Port group 45 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P45	PMC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR45	R/W	0000 0000_H	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PM45	R/W	$FFFF_H$	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR45	R/W	0000 $FFFF_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR45	R	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	P45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR45	R/W	0000 0000_H	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT45	W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PU45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PD45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC45	R/W	0000 0000_H	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC45	R/W	0000 0000_H	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD45	W	0000 0000_H	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS45	R	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(13) JTAG port JP0 D1L2H, D1M1(H) registers

Table 2.61 JTAG port group JP0 D1L2H, D1M1(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap																Remarks
					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
JP0	JPMC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPMCSR0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	JPM0	R/W	FF_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPMSR0	R/W	$0000\ 00FF_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	JPIBC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPFC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√	
	JPFCE0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√	
	JPBDC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPPR0	R	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JP0	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPSR0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	JPNOT0	W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPINV0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPU0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPD0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPDSC0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	—	—	—	√	—	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	JPODC0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	JPIS0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	
	JPISA0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	—	√	√	√	—	√	
	JPPCMD0	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	JPPROTS0	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

2.3.1.3 List of D1M2(H) Port Registers

(1) Port 0 D1M2(H) registers

Table 2.62 Port group 0 D1M2(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap																Remarks
					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
P0	PMC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMCSR0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC0	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	—	
	PM0	R/W	$FFFF_H$	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMSR0	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFCE0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PBDC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPR0	R	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	P0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PSR0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT0	W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PINV0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PU0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PD0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PDSC0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PISA0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPCMD0	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS0	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(2) Port 1 D1M2(H) registers

Table 2.63 Port group 1 D1M2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P1	PMC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PMCSR1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PIPC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	—	√	√	—	√	
	PM1	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PMSR1	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PIBC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PFC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PFCE1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PBDC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PPR1	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	P1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PSR1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PNOT1	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PINV1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PU1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PD1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PDSC1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PISA1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PPCMD1	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS1	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

Table 2.64 Port group 1 D1M2H registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P1	PMC1	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR1	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC1	R/W	0000_H	16	—	—	—	—	√	—	√	√	—	√	√	—	√	√	—	√	
	PM1	R/W	$FFFF_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR1	R/W	$0000\ FFFF_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC1	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC1	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE1	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC1	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR1	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P1	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR1	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT1	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV1	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PU1	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PD1	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC1	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC1	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS1	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA1	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD1	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS1	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(3) Port 2 D1M2(H) registers

Table 2.65 Port group 2 D1M2(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P2	PMC2	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR2	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC2	R/W	0000_H	16	—	—	—	—	—	—	—	—	√	—	√	—	√	—	√	—	
	PM2	R/W	$FFFF_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR2	R/W	$0000\ FFFF_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC2	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC2	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE2	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC2	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR2	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P2	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR2	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT2	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV2	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PU2	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PD2	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC2	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC2	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS2	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA2	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD2	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS2	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(4) Port 3 D1M2(H) registers

Table 2.66 Port group 3 D1M2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P3	PMC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMCSR3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	—	—	√	
	PM3	R/W	$FFFF_H$	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMSR3	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFCE3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PBDC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPR3	R	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	P3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PSR3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT3	W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PINV3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PU3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PD3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PDSC3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PISA3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPCMD3	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS3	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

Table 2.67 Port group 3 D1M2H registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P3	PMC3	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR3	R/W	$0000\ 0000_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	—	—	√	
	PM3	R/W	$FFFF_H$	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR3	R/W	$0000\ FFFF_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC3	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC3	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE3	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC3	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR3	R	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	P3	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR3	R/W	$0000\ 0000_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT3	W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV3	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PU3	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PD3	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC3	R/W	$0000\ 0000_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC3	R/W	$0000\ 0000_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS3	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA3	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD3	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS3	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(5) Port 10 D1M2(H) registers

Table 2.68 Port group 10 D1M2(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P10	PMC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR10	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PM10	R/W	$FFFF_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR10	R/W	$0000\ FFFF_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR10	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR10	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT10	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PODC10	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD10	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS10	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(6) Port 11 D1M2(H) registers

Table 2.69 Port group 11 D1M2(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P11	PMC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	
	PMCSR11	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Upper 16 bits
	PM11	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	
	PMSR11	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	
	PFC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	
	PBDC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	
	PPR11	R	0000_H	16	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	
	P11	R/W	0000_H	16	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	
	PSR11	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT11	W	0000_H	16	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	
	PINV11	R/W	0000_H	16	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	
	PODC11	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD11	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS11	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(7) Port 16 D1M2(H) registers

Table 2.70 Port group 16 D1M2(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P16	PMC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR16	R/W	0000 0000_H	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PM16	R/W	$F000_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR16	R/W	0000 $F000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR16	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR16	R/W	0000 0000_H	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT16	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC16	R/W	0000 0000_H	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC16	R/W	0000 0000_H	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD16	W	0000 0000_H	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS16	R	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(8) Port 17 D1M2(H) registers

Table 2.71 Port group 17 D1M2 registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P17	PMC17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PMCSR17	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Upper 16 bits
	PIPC17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PM17	R/W	$F000_H$	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PMSR17	R/W	$0000\ F000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Upper 16 bits
	PIBC17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PFC17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PFCE17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PBDC17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PPR17	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	P17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PSR17	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Upper 16 bits
	PNOT17	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PINV17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PDSC17	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC17	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PISA17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PPCMD17	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS17	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

Table 2.72 Port group 17 D1M2H registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P17	PMC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PM17	R/W	$F000_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR17	R/W	$0000\ F000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PBDC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR17	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT17	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD17	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS17	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(9) Port 21 D1M2(H) registers

Table 2.73 Port group 21 D1M2(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P21	PMC21	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR21	R/W	0000 0000_H	32	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC21	R/W	0000_H	16	—	—	—	√	√	—	√	√	√	√	√	√	√	√	—	—	
	PM21	R/W	$FFFF_H$	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR21	R/W	0000 $FFFF_H$	32	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC21	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC21	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE21	R/W	0000_H	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	PBDC21	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR21	R	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	P21	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR21	R/W	0000 0000_H	32	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT21	W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV21	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC21	R/W	0000 0000_H	32	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PU21	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PD21	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PODC21	R/W	0000 0000_H	32	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PISA21	R/W	$1C00_H$	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD21	W	0000 0000_H	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS21	R	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(10) Port 40 D1M2H registers

Table 2.74 Port group 40 D1M2H registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P40	PMC40	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PMCSR40	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PM40	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PMSR40	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PIBC40	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PBDC40	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PPR40	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	P40	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PSR40	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PNOT40	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PINV40	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PU40	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PD40	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PDSC40	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC40	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD40	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS40	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(11) Port 42 D1M2H registers

Table 2.75 Port group 42 D1M2H registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P42	PMC42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR42	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC42	R/W	0000_H	16	—	—	—	—	—	√	—	√	—	—	√	—	√	—	—	—	
	PM42	R/W	$FFFF_H$	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR42	R/W	$0000\ FFFF_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR42	R	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	P42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR42	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT42	W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PU42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PD42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC42	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC42	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD42	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS42	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(12) Port 43 D1M2(H) registers

Table 2.76 Port group 43 D1M2(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P43	PMC43	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR43	R/W	$0000\ 0000_H$	32	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PM43	R/W	$FFFF_H$	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR43	R/W	$0000\ FFFF_H$	32	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC43	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC43	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE43	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	—	—	—	—	—	√	√	
	PBDC43	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR43	R	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	P43	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR43	R/W	$0000\ 0000_H$	32	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT43	W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV43	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PU43	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PD43	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC43	R/W	$0000\ 0000_H$	32	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC43	R/W	$0000\ 0000_H$	32	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS43	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA43	R/W	0000_H	16	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD43	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS43	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(13) Port 44 D1M2(H) registers

Table 2.77 Port group 44 D1M2(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P44	PMC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	—	√	√	√	√	—	
	PM44	R/W	$FFFF_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR44	R/W	$0000\ FFFF_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR44	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT44	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PU44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PD44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD44	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS44	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(14) Port 45 D1M2(H) registers

Table 2.78 Port group 45 D1M2(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P45	PMC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR45	R/W	$0000\ 0000_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PM45	R/W	$FFFF_H$	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR45	R/W	$0000\ FFFF_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR45	R	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	P45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR45	R/W	$0000\ 0000_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT45	W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PU45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PD45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC45	R/W	$0000\ 0000_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC45	R/W	$0000\ 0000_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD45	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS45	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(15) Port 46 D1M2(H) registers

Table 2.79 Port group 46 D1M2(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P46	PMC46	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR46	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PM46	R/W	$FFFF_H$	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR46	R/W	$0000\ FFFF_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC46	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC46	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	—	√	√	√	√	√	
	PFCE46	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	
	PBDC46	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR46	R	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	P46	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR46	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT46	W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV46	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PU46	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PD46	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC46	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC46	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS46	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA46	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD46	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS46	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(16) Port 47 D1M2(H) registers

Table 2.80 Port group 47 D1M2(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P47	PMC47	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR47	R/W	$0000\ 0000_H$	32	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC47	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	√	—	—	—	—	—	—	
	PM47	R/W	$FFFF_H$	16	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	
	PMSR47	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC47	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	
	PFC47	R/W	0000_H	16	—	—	—	—	—	—	√	—	√	√	√	√	√	√	√	√	
	PBDC47	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	
	PPR47	R	0000_H	16	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	
	P47	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	
	PSR47	R/W	$0000\ 0000_H$	32	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT47	W	0000_H	16	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	
	PINV47	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	
	PU47	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	
	PD47	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	
	PDSC47	R/W	$0000\ 0000_H$	32	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC47	R/W	$0000\ 0000_H$	32	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS47	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	
	PISA47	R/W	0000_H	16	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD47	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS47	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(17) JTAG port JP0 D1M2(H) registers

Table 2.81 JTAG port group JP0 D1M2(H) registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
JP0	JPMC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPMCSR0	R/W	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	JPM0	R/W	FF_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPMSR0	R/W	0000 $00FF_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	JPIBC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPFC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√	
	JPFCE0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√	
	JPBDC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPPR0	R	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JP0	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPSR0	R/W	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	JPNOT0	W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPINV0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPU0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPD0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPDSC0	R/W	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	√	—	—	—	√	—	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	JPODC0	R/W	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	JPIS0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	
	JPISA0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	—	√	√	√	—	√	
	JPPCMD0	W	0000 0000_H	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	JPPROTS0	R	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

2.3.1.4 List of D1M1-V2, D1M1A Port Registers

(1) Port 0 D1M1-V2, D1M1A registers

Table 2.82 Port group 0 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap																Remarks
					15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
P0	PMC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMCSR0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC0	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	—	
	PM0	R/W	$FFFF_H$	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMSR0	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFCE0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PBDC0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPR0	R	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	P0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PSR0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT0	W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PINV0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PU0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PD0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PDSC0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC0	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PISA0	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPCMD0	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS0	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(2) Port 1 D1M1-V2, D1M1A registers

Table 2.83 Port group 1 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P1	PMC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PMCSR1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PIPC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	—	√	√	—	√	
	PM1	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PMSR1	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PIBC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PFC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PFCE1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PBDC1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PPR1	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	P1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PSR1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	PNOT1	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PINV1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PU1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PD1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PDSC1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC1	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PISA1	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	PPCMD1	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS1	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(3) Port 3 D1M1-V2, D1M1A registers

Table 2.84 Port group 3 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P3	PMC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMCSR3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	—	—	√	
	PM3	R/W	$FFFF_H$	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMSR3	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFCE3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PBDC3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPR3	R	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	P3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PSR3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT3	W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PINV3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PU3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PD3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PDSC3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC3	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PISA3	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPCMD3	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS3	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(4) Port 10 D1M1-V2, D1M1A registers

Table 2.85 Port group 10 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P10	PMC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR10	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PM10	R/W	$FFFF_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR10	R/W	$0000\ FFFF_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR10	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR10	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT10	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV10	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PODC10	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD10	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS10	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(5) Port 11 D1M1-V2, D1M1A registers

Table 2.86 Port group 11 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P11	PMC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PMCSR11	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Upper 16 bits
	PM11	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PMSR11	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Upper 16 bits
	PIBC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PFC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PBDC11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PPR11	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	P11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PSR11	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Upper 16 bits
	PNOT11	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PINV11	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PODC11	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD11	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS11	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(6) Port 16 D1M1-V2, D1M1A registers

Table 2.87 Port group 16 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P16	PMC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR16	R/W	0000 0000_H	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PM16	R/W	$F000_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR16	R/W	0000 $F000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR16	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR16	R/W	0000 0000_H	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT16	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC16	R/W	0000 0000_H	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC16	R/W	0000 0000_H	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA16	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD16	W	0000 0000_H	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS16	R	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(7) Port 17 D1M1-V2, D1M1A registers

Table 2.88 Port group 17 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P17	PMC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PM17	R/W	$F000_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR17	R/W	$0000\ F000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE17	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	√	√	√	√	
	PBDC17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR17	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT17	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC17	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA17	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD17	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS17	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(8) Port 21 D1M1-V2, D1M1A registers

Table 2.89 Port group 21 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P21	PMC21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMCSR21	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	—	—	
	PM21	R/W	$FFFF_H$	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PMSR21	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFC21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PFCE21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PBDC21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PPR21	R	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	P21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PSR21	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT21	W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PINV21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PU21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PD21	R/W	0000_H	16	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	
	PODC21	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD21	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS21	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(9) Port 22 D1M1-V2, D1M1A registers

Table 2.90 Port group 22 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P22	PMC22	R/W	0000_H	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	PMCSR22	R/W	0000 0000_H	32	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	Lower 16 bits
					—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIPC22	R/W	0000_H	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	PM22	R/W	$FFFF_H$	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	PMSR22	R/W	0000 $FFFF_H$	32	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	Lower 16 bits
					—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIBC22	R/W	0000_H	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	PFC22	R/W	0000_H	16	—	—	—	—	—	√ *1	—	—	—	—	—	—	—	—	—	—	
	PFCE22	R/W	0000_H	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	PBDC22	R/W	0000_H	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	PPR22	R	0000_H	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	P22	R/W	0000_H	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	PSR22	R/W	0000 0000_H	32	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	Lower 16 bits
					—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PNOT22	W	0000_H	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	PINV22	R/W	0000_H	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	PU22	R/W	0000_H	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	PD22	R/W	0000_H	16	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	
	PODC22	R/W	0000 0000_H	32	—	—	—	—	—	√	—	—	—	—	—	—	—	—	—	—	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPCMD22	W	0000 0000_H	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS22	R	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

Note 1. Not available in D1M1-V2.

(10) Port 42 D1M1-V2, D1M1A registers

Table 2.91 Port group 42 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P42	PMC42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR42	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC42	R/W	0000_H	16	—	—	—	—	—	√	—	√	—	—	√	—	√	—	—	—	
	PM42	R/W	$FFFF_H$	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR42	R/W	$0000\ FFFF_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCAE42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR42	R	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	P42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR42	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT42	W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PU42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PD42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC42	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC42	R/W	$0000\ 0000_H$	32	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA42	R/W	0000_H	16	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD42	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS42	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

(11) Port 43 D1M1-V2, D1M1A registers

Table 2.92 Port group 43 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P43	PMC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PMCSR43	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Upper 16 bits
	PIPC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√ *1	√ *1	
	PM43	R/W	$FFFF_H$	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PMSR43	R/W	$0000\ FFFF_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Upper 16 bits
	PIBC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PFC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√ *1	√	
	PFCE43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√ *1	√ *1	
	PBDC43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PPR43	R	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	P43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PSR43	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Upper 16 bits
	PNOT43	W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PINV43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PU43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PD43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PDSC43	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC43	R/W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PISA43	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	√	
	PPCMD43	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS43	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

Note 1. Not available in D1M1A.

(12) Port 44 D1M1-V2, D1M1A registers

Table 2.93 Port group 44 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P44	PMC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC44	R/W	0000_H	16	—	—	—	—	√ *1	√ *1	√	√ *1	√	√ *1	—	√	√ *1	√	√ *1	—	
	PM44	R/W	$FFFF_H$	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR44	R/W	$0000\ FFFF_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PBDC44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR44	R	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	P44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT44	W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PU44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PD44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC44	R/W	$0000\ 0000_H$	32	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA44	R/W	0000_H	16	—	—	—	—	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD44	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS44	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

Note 1. Not available in D1M1A.

(13) Port 45 D1M1-V2, D1M1A registers

Table 2.94 Port group 45 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
P45	PMC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMCSR45	R/W	$0000\ 0000_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIPC45	R/W	0000_H	16	—	—	√	√	√ *1	√ *1	√ *1	√ *1	√ *1	√ *1	√	√ *1	√	√ *1	√ *1	√ *1	
	PM45	R/W	$FFFF_H$	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PMSR45	R/W	$0000\ FFFF_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PIBC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCE45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PFCAE45	R/W	0000_H	16	—	—	√ *2	—	—	—	—	—	—	—	—	—	—	—	—	—	
	PBDC45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPR45	R	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	P45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PSR45	R/W	$0000\ 0000_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Upper 16 bits
	PNOT45	W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PINV45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PU45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PD45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PDSC45	R/W	$0000\ 0000_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PODC45	R/W	$0000\ 0000_H$	32	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PIS45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PISA45	R/W	0000_H	16	—	—	√	√	√	√	√	√	√	√	√	√	√	√	√	√	
	PPCMD45	W	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	PPROTS45	R	$0000\ 0000_H$	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

Note 1. Not available in D1M1A.

Note 2. Not available in D1M1-V2.

(14) JTAG port JP0 D1M1-V2, D1M1A registers

Table 2.95 JTAG port group JP0 D1M1-V2, D1M1A registers

Port group name	Register name	R/W	Initial value	Access size	Bitmap 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Remarks
JP0	JPMC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPMCSR0	R/W	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	JPM0	R/W	FF_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPMSR0	R/W	0000 $00FF_H$	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	JPIBC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPFC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√	
	JPFCE0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	—	√	√	√	√	
	JPBDC0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPPR0	R	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JP0	R/W	0000_H	16	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPSR0	R/W	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Upper 16 bits
	JPNOT0	W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPINV0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPU0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPD0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	
	JPDSC0	R/W	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	√	—	—	—	√	—	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	JPODC0	R/W	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	JPIS0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	—	—	—	√	—	√	
	JPISA0	R/W	00_H	8	—	—	—	—	—	—	—	—	—	—	—	√	√	√	—	√	
	JPPCMD0	W	0000 0000_H	32	—	—	—	—	—	—	—	—	√	√	√	√	√	√	√	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits
	JPPROTS0	R	0000 0000_H	32	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	√	Lower 16 bits
					—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	Upper 16 bits

2.3.2 List of Alternative Function Pins

The following sectiosn define the alternative functions of each port.

The first table in each section specifies all ports of the RH850/D1L/D1M devices with all possible alternative functions.

The following tables specify which signals and which ports are available for each device.

2.3.2.1 Port 0 (P0)

Table 2.96 Port 0 (P0)

Port mode (PMC0_m = 0)	Control Mode (PMC0_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P0_0	NMI	RLIN31TX	$\overline{\text{CSIH0SSI}}$	CSIH0CSS0	TAUJ0I0	TAUJ0O0	INTP5	OSTM0 TTOUT	MODE0
P0_1	RLIN31RX / INTP0			CSIH0SO	TAUJ0I1	TAUJ0O1		$\overline{\text{ERROROUT}}$	FLMD1
P0_2	INTP1	RLIN32TX	CSIH0SI		TAUJ0I3	TAUJ0O3		AWOT0 TOUT	MODE1
P0_3	RLIN32RX / INTP2		CSIH0SC		TAUJ0I2	TAUJ0O2		CAN0DREN	
P0_4	INTP5	OSTM1 TTOUT		TAUB1O3*1	TAUB1I3	CSCXFOUT		CAN0TX	
P0_5	TAUJ0I1	TAUJ0O1		TAUB1O5*1	TAUB1I5	RTCA0OUT	CAN0RX / INTP6		
P0_6	INTP7	RLIN30TX		TAUB1O7*1	TAUB1I7			CAN1TX	
P0_7	RLIN30RX / INTP8			TAUB1O9*1	TAUJ0I0	TAUJ0O0	CAN1RX / INTP8	CAN2DREN	
P0_8	INTP9	RLIN30TX			TAUJ0I1	TAUJ0O1		CAN2TX	
P0_9	RLIN30RX / INTP10				TAUJ0I0	TAUJ0O0	CAN2RX / INTP10	CAN1DREN	

Note 1. These ports are equipped with an XOR Compare Unit. Refer to Section 2.4.2, Port output check by XOR Compare Unit for details.

Table 2.97 P0 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
All	All	√	√	√	√	√	√	√	√	√

Note: —: signal not available, √: signal available

Table 2.98 P0 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P0_m	√	√	√	√	√	√	√	√	√

Note: —: port not available, √: port available

2.3.2.2 Port 1 (P1)

Table 2.99 Port 1 (P1)

Port mode (PMC1_m = 0)	Control Mode (PMC1_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P1_0	TAUB1I2	TAUB1O2		CSIH1SO	CSIG2RYI	CSIG2RYO			
P1_1	TAUB1I4	TAUB1O4	CSIH1SI		$\overline{\text{CSIG2SSI}}$		$\overline{\text{CSIH0SSI}}$	CSIH0CSS0	
P1_2	TAUB1I8	TAUB1O8	CSIH1SC				CSIH0RYI	CSIH0RYO	
P1_3		CSIG1SO		CSIH0SO			ADCE0 TRIGI3		
P1_4	CSIG1SI		CSIH0SI		$\overline{\text{CSIG0SSI}}$		$\overline{\text{CSIH1SSI}}$	CSIH1CSS0	
P1_5	CSIG1SC		CSIH0SC		CSIG0RYI	CSIG0RYO	CSIH1RYI	CSIH1RYO	
P1_6		CSIG2SO					TAUB0I1	TAUB0O1	
P1_7	CSIG2SI				$\overline{\text{CSIG1SSI}}$		TAUB0I2	TAUB0O2	
P1_8	CSIG2SC				CSIG1RYI	CSIG1RYO	TAUB0I3	TAUB0O3	
P1_9		CSIG0SO					TAUB0I5	TAUB0O5	
P1_10	CSIG0SI						TAUB0I6	TAUB0O6	
P1_11	CSIG0SC						TAUB0I7	TAUB0O7	

Table 2.100 P1 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
All	All	√*1	√*1	√*1	√*1	√*1	√*1	√	√*1	√*1

Note 1. Signals are partly available. Refer to the following table about the port availability.
Note 2. —: signal not available, √: signal available

Table 2.101 P1 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P1_0 to P1_2	√	√	√	√	√	√	√	√	√
P1_3 to P1_5	—	—	√	√	√	√	√	√	√
P1_6 to P1_11	—	—	—	—	—	—	√	—	—

Note: —: port not available, √: port available

2.3.2.3 Port 2 (P2)

Table 2.102 Port 2 (P2)

Port mode (PMC2_m = 0)	Control Mode (PMC2_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P2_0	$\overline{\text{CSIG3SSI}}$			PWGA0O		RIIC1SDA	TAUB2I1	TAUB2O1	
P2_1		CSIG3SO		PWGA1O		RIIC1SCL	TAUB2I3	TAUB2O3	
P2_2	CSIG3SI			PWGA2O	TAUJ0I2	TAUJ0O2	TAUB2I5	TAUB2O5	
P2_3		CSIG3SC		PWGA3O	TAUJ0I3	TAUJ0O3	TAUB2I7	TAUB2O7	
P2_4	$\overline{\text{CSIG1SSI}}$			PWGA4O	TAUB0I9	TAUB0O9	TAUB2I2	TAUB2O2	
P2_5		CSIG1SO		PWGA5O	TAUB0I10	TAUB0O10	TAUB2I6	TAUB2O6	
P2_6	CSIG1SI			PWGA6O	TAUB0I11	TAUB0O11	TAUB2I9	TAUB2O9	
P2_7		CSIG1SC		PWGA7O	TAUB0I13	TAUB0O13	TAUB2I10	TAUB2O10	
P2_8	$\overline{\text{CSIG0SSI}}$			PWGA8O	TAUB0I14	TAUB0O14	TAUB2I11	TAUB2O11	
P2_9				PWGA9O	$\overline{\text{CSIH0SSI}}$	CSIH0CSS0	TAUB2I13	TAUB2O13	
P2_10	$\overline{\text{CSIG1SSI}}$			PWGA10O	$\overline{\text{CSIH1SSI}}$	CSIH1CSS0	TAUB2I14	TAUB2O14	
P2_11	$\overline{\text{CSIG2SSI}}$			PWGA11O			TAUB2I15	TAUB2O15	

Table 2.103 P2 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
All	All	—	—	—	—	—	√	√	—	—

Note: —: signal not available, √: signal available

Table 2.104 P2 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P2_m	—	—	—	—	—	√	√	—	—

Note: —: port not available, √: port available

2.3.2.4 Port 3 (P3)

Table 2.105 Port 3 (P3)

Port mode (PMC3_m = 0)	Control Mode (PMC3_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P3_0		RIIC0SDA		ETNB0MDIO			TAUB1I10	TAUB1O10	
P3_1		RIIC0SCL		ETNB0MDC			TAUB1I11	TAUB1O11	
P3_2		PCMP0 AP0*1	SSIF0RXD	TAUB1O9		SG0FAO*1	TAUB0I4	TAUB0O4	
P3_3		PCMP0 AN0*1	TAUB1I11	SSIF0TXD	TAUB1I12	SG1FAO		SG0FAOL	
P3_4		PCMP0 AP1*1		SSIF0SCK	TAUB1I14	SG2FAO	TAUB0I8	TAUB0O8	
P3_5		PCMP0 AN1*1		SSIF0WS	TAUB2I12	SG3FAO		SG0AO	
P3_6		PCMP0 BP0*1		SSIFACK	TAUB2I14	SG4FAO	TAUB0I12	TAUB0O12	
P3_7		PCMP0 BN0*1		SSIF1WS		SG0FAOL*1		SG0AO*1	
P3_8		PCMP0 BP1*1		SSIF1SCK	TAUB1I13	SG1FAOL		SG1AO	
P3_9		PCMP0 BN1*1	SSIF1RXD	SSIF1TXD	TAUB1I15	SG2FAOL		SG2AO	
P3_10	TAUB1I6	PWGA12O	SSIF1RXD	TAUB1O6	TAUB2I13	SG3FAOL		SG3AO	
P3_11		PWGA13O	TAUB0I15	TAUB0O15	TAUB2I15	SG4FAOL		SG4AO	
P3_12		PWGA14O	TAUB1I2	TAUB1O2			ADCE0 TRIGI1	TAUB2O14	
P3_13		PWGA15O	TAUB1I10	TAUB1O10			ADCE0 TRIGI2	TAUB2O15	

Note 1. These ports are equipped with an XOR Compare Unit. Refer to Section 2.4.2, Port output check by XOR Compare Unit for details.

Table 2.106 P3 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
Ethernet AVB MAC	All ETNB0	—	—	—	√	√	√	√	√	√
All others	All others	√*1	√*1	√*1	√*1	√*1	√*1	√	√*1	√*1

Note 1. Signals are partly available. Refer to the following table about the port availability.
Note 2. —: signal not available, √: signal available

Table 2.107 P3 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P3_0 to P3_1	—	—	√	√	√	√	√	√	√
P3_2 to P3_9	√	√	√	√	√	√	√	√	√
P3_10 to P3_13	—	—	—	—	—	—	√	—	—

Note: —: port not available, √: port available

2.3.2.5 Port 10 (P10)

Table 2.108 Port 10 (P10)

Port mode (PMC10_m = 0)	Control Mode (PMC10_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P10_0			TAUB0I1	TAUB0O1	TAUB2I1	TAUB2O1			ADCE0I0
P10_1			TAUB0I2	TAUB0O2	TAUB2I2	TAUB2O2			ADCE0I1
P10_2			TAUB0I3	TAUB0O3	TAUB2I3	TAUB2O3			ADCE0I2
P10_3			TAUB0I5	TAUB0O5	TAUB2I5	TAUB2O5			ADCE0I3
P10_4			TAUB0I6	TAUB0O6	TAUB2I6	TAUB2O6			ADCE0I4
P10_5	ADCE0 TRIGI1		TAUB0I7	TAUB0O7	TAUB2I7	TAUB2O7			ADCE0I5
P10_6	ADCE0 TRIGI2		TAUB0I9	TAUB0O9	TAUB2I9	TAUB2O9			ADCE0I6
P10_7	ADCE0 TRIGI3		TAUB0I10	TAUB0O10	TAUB2I10	TAUB2O10			ADCE0I7
P10_8			TAUB0I11	TAUB0O11	TAUB2I11	TAUB2O11			ADCE0I8
P10_9			TAUB0I13	TAUB0O13	TAUB2I13	TAUB2O13			ADCE0I9
P10_10			TAUB0I14	TAUB0O14	TAUB2I14	TAUB2O14			ADCE0I10
P10_11			TAUB0I15	TAUB0O15	TAUB2I15	TAUB2O15			ADCE0I11

Table 2.109 P10 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
All	All	√	√	√	√	√	√	√	√	√

Note: —: signal not available, √: signal available

Table 2.110 P10 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P10_m	√	√	√	√	√	√	√	√	√

Note: —: port not available, √: port available

2.3.2.6 Port 11 (P11)

Table 2.111 Port 11 (P11)

Port mode (PMC11_m = 0)	Control Mode (PMC11_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P11_0			TAUB1I1	TAUB1O1					ADCE0I12
P11_1			TAUB1I2	TAUB1O2					ADCE0I13
P11_2			TAUB1I3	TAUB1O3					ADCE0I14
P11_3			TAUB1I5	TAUB1O5					ADCE0I15
P11_4			TAUB1I6	TAUB1O6					ADCE0I16
P11_5			TAUB1I7	TAUB1O7					ADCE0I17
P11_6			TAUB1I9	TAUB1O9					ADCE0I18
P11_7			TAUB1I10	TAUB1O10					ADCE0I19

Table 2.112 P11 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
A/D Converter	ADCE0I[19:16]	—	—	—	—	—	√	√	—	—
All others	All others	√*1	√*1	√*1	√*1	√*1	√	√	√*1	√*1

Note 1. Signals are partly available. Refer to the following table about the port availability.
Note 2. —: signal not available, √: signal available

Table 2.113 P11 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P11_0 to P11_3	√	√	√	√	√	√	√	√	√
P11_4 to P11_7	—	—	—	—	—	√	√	—	—

Note: —: port not available, √: port available

2.3.2.7 Port 16 (P16)

Table 2.114 Port 16 (P16)

Port mode (PMC16_m = 0)	Control Mode (PMC16_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P16_0	TAUB0I1	ISM11		PWGA16O		TAUB0O1*1	TAUB0I0	TAUB0O0	ZPD11
P16_1	TAUB0I3	ISM12		PWGA17O		TAUB0O3*1	TAUB1I0	TAUB1O0	ZPD12
P16_2	TAUB0I5	ISM13		PWGA18O		TAUB0O5*1	TAUB2I0	TAUB2O0	ZPD13
P16_3	TAUB0I7	ISM14		PWGA19O		TAUB0O7*1			ZPD14
P16_4	TAUB0I9	ISM21		PWGA20O		TAUB0O9*1			ZPD21
P16_5	TAUB0I11	ISM22		PWGA21O		TAUB0O11*1			ZPD22
P16_6	TAUB0I13	ISM23		PWGA22O		TAUB0O13*1			ZPD23
P16_7	TAUB0I15	ISM24		PWGA23O		TAUB1O1*1	TAUB1I1	TAUB0O15	ZPD24
P16_8	TAUB2I0	ISM51		TAUB1O11		PCMP0 AP0*1			ZPD51
P16_9	TAUB2I1	ISM52		TAUB1O13		PCMP0 AN0*1			ZPD52
P16_10	TAUB2I2	ISM53		TAUB1O14		PCMP0 AP1*1			ZPD53
P16_11	TAUB2I3	ISM54		TAUB1O15		PCMP0 AN1*1			ZPD54

Note 1. These ports are equipped with an XOR Compare Unit. Refer to Section 2.4.2, Port output check by XOR Compare Unit for details.

Table 2.115 P16 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
All	All	√	√	√	√	√	√	√	√	√

Note: —: signal not available, √: signal available

Table 2.116 P16 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P16_m	√	√	√	√	√	√	√	√	√

Note: —: port not available, √: port available

2.3.2.8 Port 17 (P17)

Table 2.117 Port 17 (P17)

Port mode (PMC17_m = 0)	Control Mode (PMC17_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P17_0	TAUB2I4	ISM61		TAUB2O1		PCMP0 BP0*1			ZPD61
P17_1	TAUB2I5	ISM62	ADCE0 TRIGI1	TAUB2O2		PCMP0 BN0*1			ZPD62
P17_2	TAUB2I6	ISM63	ADCE0 TRIGI2	TAUB2O3		PCMP0 BP1*1			ZPD63
P17_3	TAUB2I7	ISM64	ADCE0 TRIGI3	TAUB2O5		PCMP0 BN1*1			ZPD64
P17_4	TAUB2I8	ISM31		TAUB2O6					ZPD31
P17_5	TAUB2I9	ISM32		TAUB2O7					ZPD32
P17_6	TAUB2I10	ISM33		TAUB2O9					ZPD33
P17_7	TAUB2I11	ISM34		TAUB2O10					ZPD34
P17_8	TAUB2I12	ISM41		TAUB2O11					ZPD41
P17_9	TAUB2I13	ISM42		TAUB2O13					ZPD42
P17_10	TAUB2I14	ISM43		TAUB2O14					ZPD43
P17_11	TAUB2I15	ISM44		TAUB2O15					ZPD44

Note 1. These ports are equipped with an XOR Compare Unit. Refer to Section 2.4.2, Port output check by XOR Compare Unit for details.

Table 2.118 P17 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
All	All	√	√	√	√	√	√*1	√	√	√

Note 1. Signals are partly available. Refer to the following table about the port availability.
Note 2. —: signal not available, √: signal available

Table 2.119 P17 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P17_0 to P17_3	√	√	√	√	√	√	√	√	√
P17_4 to P17_11	√	√	√	√	√	—	√	√	√

Note: —: port not available, √: port available

2.3.2.9 Port 21 (P21)

Table 2.120 Port 21 (P21)

Port mode (PMC21_m = 0)	Control Mode (PMC21_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P21_0		SFMA0CLK	TAUB0I13	TAUB0O13		MCK			
P21_1		SFMA0SSL	TAUB0I11	TAUB0O11		MCS			
P21_2	SFMA0I00	SFMA0O00	TAUB0I10	TAUB0O10	MDQI0	MDQO0			
P21_3	SFMA0I10	SFMA0O10	TAUB0I9	TAUB0O9	MDQI1	MDQO1			
P21_4	SFMA0I20	SFMA0O20	TAUB0I7	TAUB0O7	MDQI2	MDQO2			
P21_5	SFMA0I30	SFMA0O30	TAUB0I6	TAUB0O6	MDQI3	MDQO3			
P21_6	SFMA0I01	SFMA0O01	TAUB0I5	TAUB0O5	MDQI4	MDQO4	SFMA2I00	SFMA2O00	
P21_7	SFMA0I11	SFMA0O11	TAUB0I3	TAUB0O3	MDQI5	MDQO5	SFMA2I10	SFMA2O10	
P21_8	SFMA0I21	SFMA0O21	TAUB0I2	TAUB0O2	MDQI6	MDQO6	SFMA2I20	SFMA2O20	
P21_9	SFMA0I31	SFMA0O31	TAUB0I1	TAUB0O1	MDQI7	MDQO7	SFMA2I30	SFMA2O30	
P21_10	MLBB0CLK		TAUB0I4	TAUB0O4	TAUB0I0	TAUB0O0			
P21_11		MLBB0DAT	TAUB0I14	TAUB0O14					
P21_12		MLBB0SIG	TAUB0I15	TAUB0O15					

Table 2.121 P21 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
Serial Flash Memory I/F	SFMA0O[3:0]0	√	√	√	√	√	√	√	√	√
	SFMA0I[3:0]0	√	√	√	√	√	√	√	√	√
	SFMA0O[3:0]1	—	—	√	√	√	√	√	√	√
	SFMA0I[3:0]1	—	—	√	√	√	√	√	√	√
	SFMA2O[3:0]0	—	—	—	—	—	—	—	√	—
	SFMA2I[3:0]0	—	—	—	—	—	—	—	√	—
Media Local Bus	All MLBB0	—	—	—	—	—	—	√	—	—
HyperBus/OctaBus Memory I/F	All HYPB0/OCTA0	—	—	—	—	—	—	—	√	√
All others	All others	√*1	√*1	√*1	√*1	√*1	√	√	√*1	√*1

Note 1. Signals are partly available. Refer to the following table about the port availability.
Note 2. —: signal not available, √: signal available

Table 2.122 P21 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P21_0 to P21_5	√	√	√	√	√	√	√	√	√
P21_6 to P21_9	—	—	√	√	√	√	√	√	√
P21_10 to P21_12	—	—	—	—	—	√	√	—	—

Note: —: port not available, √: port available

PDSC21 has to be set 0000 $03FF_H$ in D1M2(H).

2.3.2.10 Port 22 (P22)

Table 2.123 Port 22 (P22)

Port mode (PMC22_m = 0)	Control Mode (PMC22_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P22_10						MDQS		SFMA2CLK	

Table 2.124 P22 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
Serial Flash Memory I/F	SFMA2CLK	—	—	—	—	—	—	—	√	—
HyperBus/OctaBus Memory I/F	All HYPB0/OCTA0	—	—	—	—	—	—	—	√	√

Table 2.125 P22 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P22_10	—	—	—	—	—	—	—	√	√

Note: —: port not available, √: port available

2.3.2.11 Port 40 (P40)

Table 2.126 Port 40 (P40)

Port mode (PMC40_m = 0)	Control Mode (PMC40_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P40_0	MIPI0_VI_DATA0P								
P40_1	MIPI0_VI_DATA0N								
P40_2	MIPI0_VI_DATA1P								
P40_3	MIPI0_VI_DATA1N								
P40_4	MIPI0_VI_CLKP								
P40_5	MIPI0_VI_CLKN								

Table 2.127 P40 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
All	All	—	—	—	—	—	—	√	—	—

Note: —: signal not available, √: signal available

Table 2.128 P40 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P40_m	—	—	—	—	—	—	√	—	—

Note: —: port not available, √: port available

2.3.2.12 Port 42 (P42)

Table 2.129 Port 42 (P42) - alternative functions 1 to 4

Port mode (PMC42_m = 0)	Control Mode (PMC42_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P42_0		ETNB0TXD3	VDCE0_VI_DATA7_ITU	PWGA0O	VDCE0_VI_DATA17	TAUB2O1	RIIC1SDA		
P42_1		ETNB0TXD2	VDCE0_VI_DATA6_ITU	PWGA1O	VDCE0_VI_DATA16	TAUB2O3	RIIC1SCL		
P42_2		ETNB0TXD1	VDCE0_VI_DATA5_ITU	PWGA2O	VDCE0_VI_DATA15	TAUB2O5	$\overline{\text{CSIG0SSI}}$		
P42_3	$\overline{\text{CSIH0SSI}}$	ETNB0TXD0	VDCE0_VI_DATA4_ITU	PWGA3O	VDCE0_VI_DATA14	TAUB2O7		CSIG0SO	
P42_4	$\overline{\text{CSIH1SSI}}$	ETNB0TXEN	VDCE0_VI_DATA3_ITU	PWGA4O	VDCE0_VI_DATA13	TAUB2O2	CSIG0SI		
P42_5	$\overline{\text{CSIG2SSI}}$	ETNB0TXER	VDCE0_VI_DATA2_ITU	PWGA5O	VDCE0_VI_DATA12	TAUB2O6	CSIG0SC		
P42_6	ETNB0COL		VDCE0_VI_DATA1_ITU	PWGA6O	VDCE0_VI_DATA11	TAUB2O9	CSIG3RYI	CSIG3RYO	
P42_7	ETNB0CRSDV	SG3FAOL	VDCE0_VI_DATA0_ITU	PWGA12O	VDCE0_VI_DATA10	TAUB1O6	$\overline{\text{CSIG3SSI}}$	SG3AO	
P42_8	ETNB0TXCLK	SG3FAO	VDCE0_VI_CLK	PWGA13O	VDCE0_VI_DATA9	TAUB0O15		CSIG3SO	
P42_9	ETNB0RXCLK	SG4FAOL	VDCE0_VI_HSYNC	PWGA14O	VDCE0_VI_DATA8	TAUB1O2	CSIG3SI	SG4AO	
P42_10	ETNB0RXD3	SG4FAO	VDCE0_VI_VSYNC	PWGA15O	VDCE0_VI_DATA7	TAUB1O10	CSIG3SC		
P42_11	ETNB0RXD2	VDCE0_VO_TCON3	TAUB2I9	PWGA7O	VDCE0_VI_DATA6	TAUB2O10	TAUB0I14	TAUB0O14	
P42_12	ETNB0RXD1	VDCE0_VO_TCON1	TAUB2I10	PWGA8O	VDCE0_VI_DATA5	TAUB2O11	TAUJ0I2	TAUJ0O2	
P42_13	ETNB0RXD0	VDCE0_VO_TCON4	TAUB2I11	PWGA9O	VDCE0_VI_DATA4	TAUB2O13	TAUJ0I3	TAUJ0O3	
P42_14	ETNB0RXDV	VDCE0_VO_TCON5	ADCE0TRIGI1	PWGA10O	VDCE0_VI_DATA3	TAUB2O14	TAUB1I2		
P42_15	ETNB0RXER	VDCE0_VO_TCON6	ADCE0TRIGI2	PWGA11O	VDCE0_VI_DATA2	TAUB2O15	TAUB1I10		

Table 2.130 Port 42 (P42) - alternative functions 5 and 6 (D1M1A, D1M1-V2 only)

	Control Mode (PMC42_m = 1)			
	5th Alternative		6th Alternative	
Port mode (PMC42_m = 0)	Input	Output	Input	Output
P42_0		NAND_CE		SFMA1CLK
P42_1		NAND_CLE		SFMA1SSL
P42_2		NAND_ALE	SFMA1I00	SFMA1O00
P42_3		NAND_REB	SFMA1I10	SFMA1O10
P42_4		NAND_WEB	SFMA1I20	SFMA1O20
P42_5	NAND_RB0		SFMA1I30	SFMA1O30
P42_6	NAND_DI[0]	NAND_DO[0]	SFMA1I01	SFMA1O01
P42_7	NAND_DI[1]	NAND_DO[1]	SFMA1I11	SFMA1O11
P42_8	NAND_DI[2]	NAND_DO[2]	SFMA1I21	SFMA1O21
P42_9	NAND_DI[3]	NAND_DO[3]	SFMA1I31	SFMA1O31
P42_10	NAND_DI[4]	NAND_DO[4]	CSIG3SC	
P42_11	NAND_DI[5]	NAND_DO[5]	CSIG3SI	
P42_12	NAND_DI[6]	NAND_DO[6]		CSIG3SO
P42_13	NAND_DI[7]	NAND_DO[7]	$\overline{\text{CSIG3SSI}}$	
P42_14	INTP7			
P42_15	INTP4			

Table 2.131 P42 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
Ethernet AVB MAC	all ETNB0	—	—	—	√	√	—	√	√	√
Video Display Controller	all VDCE0_VO	—	—	√	√	√	—	—	√	√
	VDCE0_VI 2nd alternative inputs	—	—	—	√	√	—	√	√	√
	VDCE0_VI 3rd alternative inputs	—	—	—	√	√	—	—	√	√
INTP	INTP4, INTP7	—	—	—	—	—	—	—	√	√
NFMA0	all NFMA	—	—	—	—	—	—	—	√	—
SFMA1	all SFMA1	—	—	—	—	—	—	—	√	√
All others	All others	√*1	√*1	√	√	√	—	√	√	√

Note 1. Signals are partly available. Refer to the following table about the port availability.

Note 2. —: signal not available, √: signal available

Table 2.132 P42 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P42_0 to P42_6	—	—	√	√	√	—	√	√	√
P42_7 to P42_10	√	√	√	√	√	—	√	√	√
P42_11 to P42_15	—	—	√	√	√	—	√	√	√

Note: —: port not available, √: port available

2.3.2.13 Port 43 (P43)

Table 2.133 Port 43 (P43)

Port mode (PMC43_m = 0)	Control Mode (PMC43_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P43_0		VDCE0_VO_TCON2		VDCE0_VO_TCON3		$\overline{\text{LCBI0}}$ HSYNC		$\overline{\text{LCBI0CS}}$	
P43_1		VDCE0_VO_TCON0				$\overline{\text{LCBI0}}$ VSYNC		$\overline{\text{LCBI0}}$ WRRW	
P43_2		VDCE0_VO_TCON3	VDCE1_VI_DATA7_ITU						
P43_3		VDCE0_VO_TCON1	VDCE1_VI_DATA6_ITU						
P43_4		VDCE0_VO_TCON4	VDCE1_VI_DATA5_ITU						
P43_5		VDCE0_VO_TCON5	VDCE1_VI_DATA4_ITU						
P43_6		VDCE0_VO_TCON6	VDCE1_VI_DATA3_ITU	VDCE1_VO_TCON3					
P43_7	VDCE1_VI_DATA23		VDCE1_VI_DATA2_ITU		VDCE0_VI_DATA2_ITU		TAUB0I9	TAUB0O9	
P43_8	VDCE1_VI_DATA22		VDCE1_VI_DATA1_ITU		VDCE0_VI_DATA1_ITU		TAUB0I10	TAUB0O10	
P43_9	VDCE1_VI_DATA21		VDCE1_VI_DATA0_ITU		VDCE0_VI_DATA0_ITU		TAUB0I11	TAUB0O11	
P43_10	VDCE1_VI_DATA20		VDCE1_VI_CLK		VDCE0_VI_CLK		TAUB0I13	TAUB0O13	
P43_11	VDCE1_VI_DATA19		VDCE1_VI_HSYNC		VDCE0_VI_HSYNC		TAUB0I14	TAUB0O14	
P43_12	VDCE1_VI_DATA18		VDCE1_VI_VSYNC		VDCE0_VI_VSYNC		TAUB0I15	TAUB0O15	

Table 2.134 P43 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
LCD Bus I/F	All LCBI	√	√	√	√	—	—	—	—	√
Video Display Controller	All VDCE0_VO	—	√*1	√*1	√*1	√*1	√	√	√*1	√*1
	All VDCE0_VI	—	—	—	—	—	√	√	—	—
	All VDCE1_VI	—	—	—	—	—	√	√	—	—
All others	All others	—	—	—	—	—	√	√	—	—

Note 1. Signals are partly available. Refer to the following table about the port availability.

Note 2. —: signal not available, √: signal available

Table 2.135 P43 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P43_0 to P43_1	√	√	√	√	√	√	√	√	√
P43_2 to P43_12	—	—	—	—	—	√	√	—	—

Note: —: port not available, √: port available

2.3.2.14 Port 44 (P44)

Table 2.136 Port 44 (P44)

Port mode (PMC44_m = 0)	Control Mode (PMC44_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P44_0		VDCE0_VO_DATA23					CSIG2RYI	CSIG2RYO	
P44_1	TAUB1I10	VDCE0_VO_DATA22			VDCE0_VI_DATA1	$\overline{\text{LCBI0RDE}}$	$\overline{\text{CSIG2SSI}}$	TAUB1O10	
P44_2		VDCE0_VO_DATA21	RIIC0SDA		VDCE0_VI_DATA0	LCBI0A0DE		CSIG2SO	
P44_3	TAUB1I11	VDCE0_VO_DATA20	RIIC0SCL		VDCE0_VI_CLK	$\overline{\text{LCBI0CS}}$	CSIG2SI	TAUB1O11	
P44_4		VDCE0_VO_DATA19		PWGA14O	VDCE0_VI_HSYNC	$\overline{\text{LCBI0}}$ $\overline{\text{WRRW}}$		CSIG2SC	
P44_5		VDCE0_VO_DATA18		PWGA15O	VDCE0_VI_VSYNC				
P44_6		VDCE0_VO_DATA17	INTP3	PWGA0O		LCBI0D0	$\overline{\text{CSIG3SSI}}$	RLIN33TX	
P44_7		VDCE0_VO_DATA16	RLIN33RX / INTP4	PWGA1O		LCBI0D1		CSIG3SO	
P44_8		VDCE0_VO_DATA15	NMI	PWGA2O		LCBI0D2	CSIG3SI	TAUB2O4	
P44_9		VDCE0_VO_DATA14	INTP1	PWGA3O		LCBI0D3		CSIG3SC	
P44_10		VDCE0_VO_DATA13	INTP5	PWGA4O		LCBI0D4		RIIC1SDA	
P44_11		VDCE0_VO_DATA12	INTP7	PWGA5O		LCBI0D5		RIIC1SCL	

Table 2.137 P44 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
LCD Bus I/F	All LCBI0	√	√	√	√	—	—	—	—	√
Video Display Controller	VDCE0_VO_Data[23:18]_24bpp	—	—	—	√	√	√	√	√	√
	All VDCE0_VI	—	—	—	√	√	—	—	√	√
All others	All others	√*1	√*1	√	√	√	√	√	√	√

Note 1. Signals are partly available. Refer to the following table about the port availability.

Note 2. —: signal not available, √: signal available

Table 2.138 P44 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P44_0	—	—	√	√	√	√	√	√	√
P44_1 to P44_4	√	√	√	√	√	√	√	√	√
P44_5	—	—	√	√	√	√	√	√	√
P44_6 to P44_11	√	√	√	√	√	√	√	√	√

Note: —: port not available, √: port available

2.3.2.15 Port 45 (P45)

Table 2.139 Port 45 (P45) - alternative functions 1 to 4

Port mode (PMC45_m = 0)	Control Mode (PMC45_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P45_0		VDCE0_VO_CLKP		PWGA13O	ADCE0TRIGI1	$\overline{\text{LCBI0RDE}}$	TAUB2I6	TAUB2O6	
P45_1		VDCE0_VO_CLKN		PWGA12O	ADCE0TRIGI2	LCBI0A0DE	TAUB2I8	VDCE0_VO_TCON3*2	
P45_2		VDCE0_VO_DATA11		PWGA6O		LCBI0D6	$\overline{\text{CSIG0SSI}}$	TAUB2O12	
P45_3		VDCE0_VO_DATA10		PWGA7O		LCBI0D7		CSIG0SO	
P45_4		VDCE0_VO_DATA9		PWGA8O		LCBI0D8	CSIG0SI	TAUB2O10	
P45_5		VDCE0_VO_DATA8		PWGA9O		LCBI0D9		CSIG0SC	
P45_6		VDCE0_VO_DATA7		PWGA10O		LCBI0D10	TAUJ0I2	TAUJ0O2	
P45_7		VDCE0_VO_DATA6		PWGA11O		LCBI0D11	TAUJ0I3	TAUJ0O3	
P45_8	TAUB0I2	VDCE0_VO_DATA5		TAUB1O3*1		LCBI0D12	TAUB1I3	TAUB0O2	
P45_9	TAUB0I6	VDCE0_VO_DATA4		TAUB1O5*1		LCBI0D13	TAUB1I5	TAUB0O6	
P45_10	TAUB0I10	VDCE0_VO_DATA3		TAUB1O7*1		LCBI0D14	$\overline{\text{CSIG1SSI}}$	TAUB0O10	
P45_11	TAUB0I14	VDCE0_VO_DATA2		TAUB1O9*1		LCBI0D15	CSIG1SI	TAUB0O14	
P45_12		VDCE0_VO_DATA1				LCBI0D16		CSIG1SO	
P45_13		VDCE0_VO_DATA0				LCBI0D17		CSIG1SC	

Note 1. These ports are equipped with an XOR Compare Unit. Refer to Section 2.4.2, Port output check by XOR Compare Unit for details.

Note 2. The usage of pin P45_1 for VDCE0_VO_TCON3 or SFMA2SSL can be changed with the register bit XCRAMCFG0.CSSEL2 as described in Section 14.4, Bus Switch for external memory interfaces (D1M1-V2, D1M1A only).

Table 2.140 Port 45 (P45) - alternative functions 5 (D1M1A only)

	Control Mode (PMC45_m = 1)	
	5th Alternative	
Port mode (PMC45_m = 0)	Input	Output
P45_0		
P45_1		
P45_2		
P45_3		
P45_4		
P45_5		
P45_6		
P45_7		
P45_8		
P45_9		
P45_10		
P45_11		
P45_12		
P45_13		SFMA2SSL*1
P45_14		
P45_15		

Note 1. The usage of pin P45_13 for SFMA2 or OCTA0 can be changed with the register bit XCRAMCFG0.CSSEL as described in Section 14.4, Bus Switch for external memory interfaces (D1M1-V2, D1M1A only).

Table 2.141 P45 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
LCD Bus I/F	All LCBI0	√	√	√	√	—	—	—	—	√
Video Display Controller	All VDCE0	—	√	√	√	√	√	√	√	√
Serial Flash Memory I/F / OCTA Memory I/F	SFMA2SSL/ OCTA0MCS1	—	—	—	—	—	—	—	√	—
All others	All others	√	√	√	√	√	√	√	√	√

Note: —: signal not available, √: signal available

Table 2.142 P45 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P45_m	√	√	√	√	√	√	√	√	√

Note: —: port not available, √: port available

2.3.2.16 Port 46 (P46)

Table 2.143 Port 46 (P46)

Port mode (PMC46_m = 0)	Control Mode (PMC46_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P46_0	VDCE1_VI_DATA23	VDCE1_VO_DATA23	VDCE1_VI_DATA7_ITU	VDCE1_VO_TCON3	VDCE0_VI_DATA7_ITU				
P46_1	VDCE1_VI_DATA22	VDCE1_VO_DATA22	VDCE1_VI_DATA6_ITU	VDCE1_VO_TCON1	VDCE0_VI_DATA6_ITU				
P46_2	VDCE1_VI_DATA21	VDCE1_VO_DATA21	VDCE1_VI_DATA5_ITU	VDCE1_VO_TCON4	VDCE0_VI_DATA5_ITU				
P46_3	VDCE1_VI_DATA20	VDCE1_VO_DATA20	VDCE1_VI_DATA4_ITU	VDCE1_VO_TCON5	VDCE0_VI_DATA4_ITU				
P46_4	VDCE1_VI_DATA19	VDCE1_VO_DATA19	VDCE1_VI_DATA3_ITU	VDCE1_VO_TCON6	VDCE0_VI_DATA3_ITU				
P46_5	VDCE1_VI_DATA18	VDCE1_VO_DATA18							
P46_6	VDCE1_VI_DATA17	VDCE1_VO_DATA17		ETNB0TXD3					
P46_7	VDCE1_VI_DATA16	VDCE1_VO_DATA16		ETNB0TXD2					
P46_8	VDCE1_VI_DATA15	VDCE1_VO_DATA15		ETNB0TXD1					
P46_9	VDCE1_VI_DATA14	VDCE1_VO_DATA14		ETNB0TXD0					
P46_10	VDCE1_VI_DATA13	VDCE1_VO_DATA13		ETNB0TXEN					
P46_11	VDCE1_VI_DATA12	VDCE1_VO_DATA12		ETNB0TXER					
P46_12	VDCE1_VI_DATA11	VDCE1_VO_DATA11	ETNB0COL						
P46_13	VDCE1_VI_DATA10	VDCE1_VO_DATA10	ETNB0CRSDV						
P46_14	VDCE1_VI_DATA9	VDCE1_VO_DATA9	ETNB0TXCLK						
P46_15	VDCE1_VI_DATA8	VDCE1_VO_DATA8	ETNB0RXCLK						

Table 2.144 P46 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1MA	D1M1-V2
Ethernet AVB MAC	All ETNB0	—	—	—	—	—	√	√	—	—
Video Display Controller	All VDCE0_VI	—	—	—	—	—	—	√	—	—
	All VDCE1_VO	—	—	—	—	—	√	√	—	—
	All VDCE1_VI	—	—	—	—	—	√	√	—	—
All others	All others	—	—	—	—	—	√	√	—	—

Note: —: signal not available, √: signal available

Table 2.145 P46 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P46_m	—	—	—	—	—	√	√	—	—

Note: —: port not available, √: port available

2.3.2.17 Port 47 (P47)

Table 2.146 Port 47 (P47)

Port mode (PMC47_m = 0)	Control Mode (PMC47_m = 1)								Dedicated function
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
P47_0	VDCE1_VI_DATA7	VDCE1_VO_DATA7	ETNB0RXD3						
P47_1	VDCE1_VI_DATA6	VDCE1_VO_DATA6	ETNB0RXD2						
P47_2	VDCE1_VI_DATA5	VDCE1_VO_DATA5	ETNB0RXD1						
P47_3	VDCE1_VI_DATA4	VDCE1_VO_DATA4	ETNB0RXD0						
P47_4	VDCE1_VI_DATA3	VDCE1_VO_DATA3	ETNB0RXDV						
P47_5	VDCE1_VI_DATA2	VDCE1_VO_DATA2	ETNB0RXER						
P47_6	VDCE1_VI_DATA1	VDCE1_VO_DATA1	ETNB0MDIO						
P47_7	VDCE1_VI_DATA0	VDCE1_VO_DATA0		ETNB0MDC					
P47_8	VDCE1_VI_CLK	VDCE1_VO_CLK							
P47_9	VDCE1_VI_HSYNC	VDCE1_VO_TCON2		VDCE1_VO_TCON3					
P47_10	VDCE1_VI_VSYNC	VDCE1_VO_TCON0							

Table 2.147 P47 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
Ethernet AVB MAC	All ETNB0	—	—	—	—	—	√	√	—	—
Video Display Controller	All VDCE1_VO	—	—	—	—	—	√	√	—	—
	All VDCE1_VI	—	—	—	—	—	√	√	—	—
All others	All others	—	—	—	—	—	√	√	—	—

Note: —: signal not available, √: signal available

Table 2.148 P47 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
P47_m	—	—	—	—	—	√	√	—	—

Note: —: port not available, √: port available

2.3.2.18 JTAG Port 0 (JP0)

Table 2.149 JTAG Port 0 (JP0)

Port mode (JPMC0_m = 0)	Control Mode (JPMC0_m = 1)								Dedicated function*1
	1st Alternative		2nd Alternative		3rd Alternative		4th Alternative		
	Input	Output	Input	Output	Input	Output	Input	Output	
JP0_0	INTP0		TAUB2I8	TAUB2O8	FLSCI3RXD	FLSCI3TXD			DCUTDI/LPDIO/LPDI
JP0_1	INTP9		TAUB1I6	TAUB1O6		FLSCI3TXD			DCUTDO/–/LPDO
JP0_2	INTP7		TAUB1I12	TAUB1O12	FLSCI3SCKI			$\overline{\text{ERROROUT}}$	DCUTCK/–/LPDCLK
JP0_3	INTP3	RLIN33TX	TAUJ0I2	TAUJ0O2		CSCXFOUT			DCUTMS/–/–
JP0_4									$\overline{\text{DCUTRST}}$/–/–
JP0_5	RLIN33RX / INTP4		TAUJ0I3	TAUJ0O3		RTCA0OUT			$\overline{\text{DCUTRDY}}$/–/LPDCLKOUT

Note 1. The debugger signals are given in the following order: JTAG/1-pin low-pin debug/4-pin low-pin debug.

Table 2.150 JP0 signals availability

Module	Signals	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
All	All	√	√	√	√	√	√	√	√	√

Note: —: signal not available, √: signal available

Table 2.151 JP0 ports availability

Port	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M2	D1M2H	D1M1A	D1M1-V2
JP0_m	√	√	√	√	√	√	√	√	√

Note: —: port not available, √: port available

2.4 Output ports check functions

Additionally to the capability of analyzing several PWM output signals, generated by the PWM Generators (refer to Section 32, PWM Generators and Diagnostic (PWM-Diag) for details), two other facilities allow to monitor output signals of further modules.

2.4.1 Port output check by timers

The PWM output signals of

- all Stepper Motor Controller/Driver (ISM) outputs
- the Sound Generators outputs
  - SG1: SG1FAO and SG1FAOL
  - SG2: SG2FAO and SG2FAOL
  - SG3: SG3FAO at port P3_5 and SG3FAOL at port P3_10
  - SG4: SG4FAO at port P3_6 and SG4FAOL at port P3_11

can be input to the capture inputs of some Timer Array Unit B (TAUB) channels.

This is possible due to the assignment of the ISM and SGn output signals and TAUB input signals to the same ports and concurrent activation of the output and input buffer of that port.

By this means the timing properties, e.g. cycle duration and duty factor, can be measured and pin short circuits can be detected.

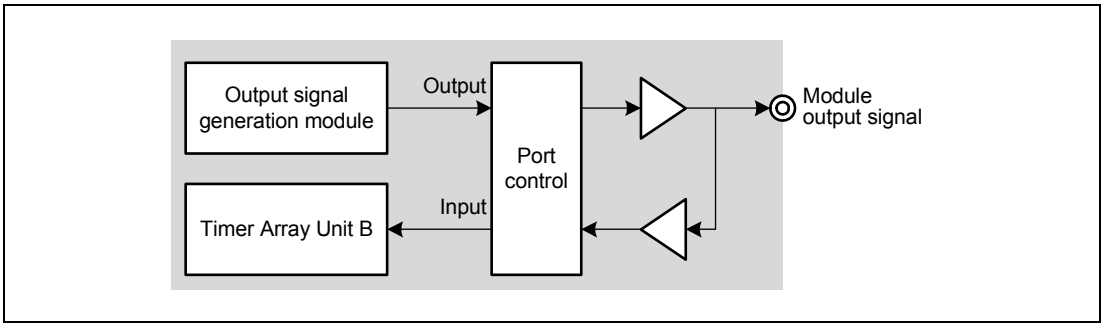


Figure 2.13 Port output check by timers

2.4.2 Port output check by XOR Compare Unit

Another way to detect a mismatch between the intended signal output and the real level at the output pin involves an XOR compare between the module output and the pin level.

The XOR Compare Unit can detect short-circuits of the pins.

Like with the output check by a timer, as described above, the output signal and the signal to compare are assigned to the same port with output and input buffers active.

A de-glitch filter suppresses possible spikes in the XOR output signal, which may occur due to signal propagation times and external capacitive load of the output buffer.
Refer to Section 2.6, Noise Filter and Edge Level Detection Circuit for details about the filter.

Since this port check can be permanently active - it does not require any CPU involvement - a mismatch between the output and input signal generates an error signals towards the Error Control Module and thus can assert interrupts or a reset.

XOR Compare Units are available for following modules, that generate output signals:

- Timer Array Unit B (TAUB) outputs:
  - TAUB0
    TAUB0O1, TAUB0O3, TAUB0O5, TAUB0O7, TAUB0O9, TAUB0O11, TAUB0O13
  - TAUB1
    TAUB1O1, TAUB1O3, TAUB1O5, TAUB1O7, TAUB1O9
- all PCM-PWM Convert (PCMP) outputs:
  PCMP0AP0, PCMP0AN0, PCMP0BP0, PCMP0BN0, PCMP0AP1, PCMP0AN1, PCMP0BP1, PCMP0BN1
- the Sound Generator SG0 outputs:
  SG0FAO, SG0FAOLat port P3_7, SG0AO at port P3_7

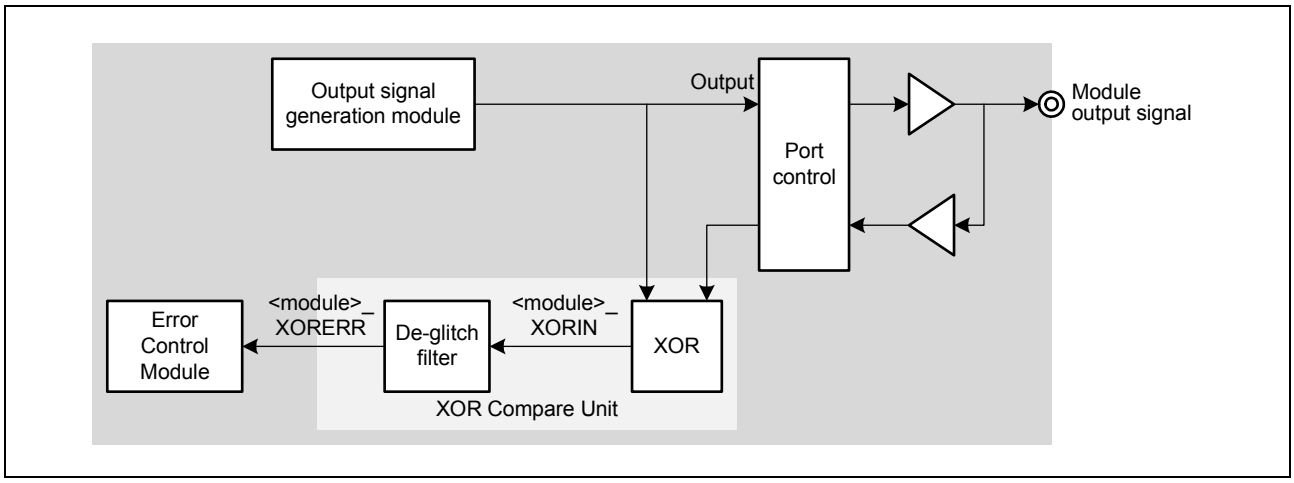


Figure 2.14 Port output check by XOR Compare Units

The XOR compare unit can be separately enabled or disabled for each signal by use of the following registers.

2.4.2.1 PRMR0CFG0 — Port XOR compare unit control register 0

This register controls the XOR compare units for Timer Array Unit B output signals.

Access: This register can be read/written in 32-bit units.

Address: FFC0 6000_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	PRMR0XDBE19	PRMR0XDBE17	PRMR0XDBE15	PRMR0XDBE13	PRMR0XDBE11	0	PRMR0XDBE013	PRMR0XDBE011	PRMR0XDBE09	PRMR0XDBE07	PRMR0XDBE05	PRMR0XDBE03	PRMR0XDBE01
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.152 PRMR0CFG0 register contents

Bit Position	Bit Name	Function
12	PRMR0XDBE19	Enable XOR compare unit for TAUB1O9 0: XOR compare unit disabled 1: XOR compare unit enabled
11	PRMR0XDBE17	Enable XOR compare unit for TAUB1O7 0: XOR compare unit disabled 1: XOR compare unit enabled
10	PRMR0XDBE15	Enable XOR compare unit for TAUB1O5 0: XOR compare unit disabled 1: XOR compare unit enabled
9	PRMR0XDBE13	Enable XOR compare unit for TAUB1O3 0: XOR compare unit disabled 1: XOR compare unit enabled
8	PRMR0XDBE11	Enable XOR compare unit for TAUB1O1 0: XOR compare unit disabled 1: XOR compare unit enabled
6	PRMR0XDBE013	Enable XOR compare unit for TAUB0O13 0: XOR compare unit disabled 1: XOR compare unit enabled
5	PRMR0XDBE011	Enable XOR compare unit for TAUB0O11 0: XOR compare unit disabled 1: XOR compare unit enabled
4	PRMR0XDBE09	Enable XOR compare unit for TAUB0O9 0: XOR compare unit disabled 1: XOR compare unit enabled
3	PRMR0XDBE07	Enable XOR compare unit for TAUB0O7 0: XOR compare unit disabled 1: XOR compare unit enabled
2	PRMR0XDBE05	Enable XOR compare unit for TAUB0O5 0: XOR compare unit disabled 1: XOR compare unit enabled
1	PRMR0XDBE03	Enable XOR compare unit for TAUB0O3 0: XOR compare unit disabled 1: XOR compare unit enabled
0	PRMR0XDBE01	Enable XOR compare unit for TAUB0O1 0: XOR compare unit disabled 1: XOR compare unit enabled

2.4.2.2 PRMR0CFG1 — Port XOR compare unit control register 1

This register controls the XOR compare units for PCM-PWM Converter output signals.

Access: This register can be read/written in 32-bit units.

Address: FFC0 6004_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	PRMR0 XDPE7	PRMR0 XDPE6	PRMR0 XDPE5	PRMR0 XDPE4	PRMR0 XDPE3	PRMR0 XDPE2	PRMR0 XDPE1	PRMR0 XDPE0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.153 PRMR0CFG1 register contents

Bit Position	Bit Name	Function
7	PRMR0XDPE 7	Enable XOR compare unit for PCMP0BN1 0: XOR compare unit disabled 1: XOR compare unit enabled
6	PRMR0XDPE 6	Enable XOR compare unit for PCMP0BN0 0: XOR compare unit disabled 1: XOR compare unit enabled
5	PRMR0XDPE 5	Enable XOR compare unit for PCMP0AP0 0: XOR compare unit disabled 1: XOR compare unit enabled
4	PRMR0XDPE 4	Enable XOR compare unit for PCMP0AN0 0: XOR compare unit disabled 1: XOR compare unit enabled
3	PRMR0XDPE 3	Enable XOR compare unit for PCMP0AP1 0: XOR compare unit disabled 1: XOR compare unit enabled
2	PRMR0XDPE 2	Enable XOR compare unit for PCMP0BP1 0: XOR compare unit disabled 1: XOR compare unit enabled
1	PRMR0XDPE 1	Enable XOR compare unit for PCMP0BP0 0: XOR compare unit disabled 1: XOR compare unit enabled
0	PRMR0XDPE 0	Enable XOR compare unit for PCMP0AN1 0: XOR compare unit disabled 1: XOR compare unit enabled

2.4.2.3 PRMR0CFG2 — Port XOR compare unit control register 2

This register controls the XOR compare units for the Sound Generator output signals.

Access: This register can be read/written in 32-bit units.

Address: FFC0 6008_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	PRMR0 XDPE2	PRMR0 XDPE1	PRMR0 XDPE0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 2.154 PRMR0CFG2 register contents

Bit Position	Bit Name	Function
2	PRMR0XDPE 2	Enable XOR compare unit for SG0AO at port P3_7 0: XOR compare unit disabled 1: XOR compare unit enabled
1	PRMR0XDPE 1	Enable XOR compare unit for SG0FAOL at port P3_7 0: XOR compare unit disabled 1: XOR compare unit enabled
0	PRMR0XDPE 0	Enable XOR compare unit for SG0FAO 0: XOR compare unit disabled 1: XOR compare unit enabled

CAUTION

The XOR compare unit for SG0FAO (at port P3_2), SG0FAOL (at port P3_7) and SG0AO (at port P3_7) can be enabled for each of this functions here separately.

However the three functions share the same de-glitch filter (DNFA10NFEN8, refer to Table 2.156, List of XOR Compare Unit Filters). By that means the signal that is provided to the de-glitch filter is the logic OR-function of all activated XOR compare functions of SG0.

2.5 DNF

Digital Noise Filter (DNF) eliminates digital noise from external input signals. This product includes two sorts of DNF: peripheral function DNF and edge detection DNF.

2.5.1 Example of Noise Elimination

Figure 2.15 shows an example of noise elimination in peripheral function DNF and edge detection DNF. In this example, the sampling clock, the sampling count, and the current output level are set to 1/2 of the DNF input clock, two (twice), and low, respectively. "○" in the figure means that high level is detected.

In input examples 1, 2, and 3, the output level changes from low to high because the same level is detected twice in a row through sampling. In input examples 4, 5, and 6, on the other hand, the same level is not detected twice consecutively. Therefore, these inputs are regarded as noise and the input signal state is eliminated.

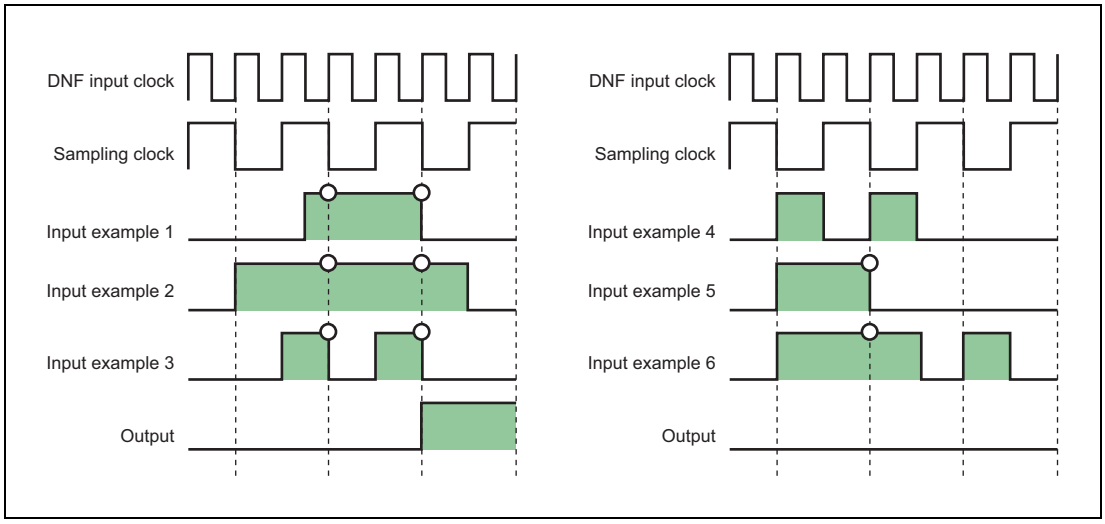


Figure 2.15 Timing Chart of Digital Noise Elimination

2.6 Noise Filter and Edge Level Detection Circuit

Some signals that are input to a pin pass the filter to eliminate noises and glitches. This product supports both analog filter and digital filter.

Additionally, the product also supports the function of detecting an edge or level after a pass through a filter.

The first section describes port input signals to which filters are allocated and the types of the filters, the noise filter and edge level detection control register and its control bits, and an overview of the register address and others.

For details about the digital/analog filter function and the noise filter and edge level detection control register, see Section 2.6.3.1, Analog Filter Type A Input Pin, Section 2.6.3.2, Analog Filter Type B Input Pin and Section 2.6.3.3, Digital Filter Type C (No Edge Detection) Input Pin.

2.6.1 Allocation of Port Filters

The following is a list of input pins that have analog filter or digital filter.

Table 2.155 List of Input Pins with a Noise Filter (1/3)

Input Pin	Filter Type	Digital Noise Elimination Control Register			Digital Noise Elimination Enable Register			Filter Control Register	
		Control Register	Address	Input Clock	Control Register	Control Bit	Address	Control Register	Address
$\overline{\text{RESET}}$	A	—	—	—	—	—	—	—	—
$\overline{\text{DCUTRST}}$	A	—	—	—	—	—	—	—	—
FLMD0	A	—	—	—	—	—	—	—	—
FLMD1	A	—	—	—	—	—	—	—	—
MODE0	A	—	—	—	—	—	—	—	—
MODE1	A	—	—	—	—	—	—	—	—
PWRGD	A	—	—	—	—	—	—	—	—
NMI	B	—	—	—	—	—	—	FCLA0CTL0	FFC3 4000_H
INTP0	B	—	—	—	—	—	—	FCLA0CTL1	FFC3 4004_H
INTP1	B	—	—	—	—	—	—	FCLA0CTL2	FFC3 4008_H
INTP2	B	—	—	—	—	—	—	FCLA0CTL3	FFC3 $400C_H$
INTP3	B	—	—	—	—	—	—	FCLA0CTL4	FFC3 4010_H
INTP4	B	—	—	—	—	—	—	FCLA0CTL5	FFC3 4014_H
INPT5	B	—	—	—	—	—	—	FCLA0CTL6	FFC3 4018_H
INTP6	B	—	—	—	—	—	—	FCLA0CTL7	FFC3 $401C_H$
INTP7	B	—	—	—	—	—	—	FCLA1CTL0	FFC3 4100_H
INTP8	B	—	—	—	—	—	—	FCLA1CTL1	FFC3 4104_H
INTP9	B	—	—	—	—	—	—	FCLA1CTL2	FFC3 4108_H
INTP10	B	—	—	—	—	—	—	FCLA1CTL3	FFC3 $410C_H$
TAUB0I0	C	DNFA0CTL	FFC3 0000_H	C_ISO_TAUB01	DNFA0EN	DNFA0NFEN0	FFC3 0004_H	—	—
TAUB0I1	C					DNFA0NFEN1		—	—
TAUB0I2	C					DNFA0NFEN2		—	—
TAUB0I3	C					DNFA0NFEN3		—	—
TAUB0I4	C					DNFA0NFEN4		—	—
TAUB0I5	C					DNFA0NFEN5		—	—
TAUB0I6	C					DNFA0NFEN6		—	—
TAUB0I7	C					DNFA0NFEN7		—	—
TAUB0I8	C					DNFA0NFEN8		—	—
TAUB0I9	C					DNFA0NFEN9		—	—
TAUB0I10	C					DNFA0NFEN10		—	—
TAUB0I11	C					DNFA0NFEN11		—	—
TAUB0I12	C					DNFA0NFEN12		—	—
TAUB0I13	C					DNFA0NFEN13		—	—
TAUB0I14	C					DNFA0NFEN14		—	—
TAUB0I15	C					DNFA0NFEN15		—	—

Table 2.155 List of Input Pins with a Noise Filter (2/3)

Input Pin	Filter Type	Digital Noise Elimination Control Register			Digital Noise Elimination Enable Register			Filter Control Register	
		Control Register	Address	Input Clock	Control Register	Control Bit	Address	Control Register	Address
TAUB1I0	C	DNFA1CTL	FFC3 0100$_H$	C_ISO_TAUB01	DNFA1EN	DNFA1NFEN0	FFC3 0104$_H$	—	—
TAUB1I1	C					DNFA1NFEN1		—	—
TAUB1I2	C					DNFA1NFEN2		—	—
TAUB1I3	C					DNFA1NFEN3		—	—
TAUB1I4	C					DNFA1NFEN4		—	—
TAUB1I5	C					DNFA1NFEN5		—	—
TAUB1I6	C					DNFA1NFEN6		—	—
TAUB1I7	C					DNFA1NFEN7		—	—
TAUB1I8	C					DNFA1NFEN8		—	—
TAUB1I9	C					DNFA1NFEN9		—	—
TAUB1I10	C					DNFA1NFEN10		—	—
TAUB1I11	C					DNFA1NFEN11		—	—
TAUB1I12	C					DNFA1NFEN12		—	—
TAUB1I13	C					DNFA1NFEN13		—	—
TAUB1I14	C					DNFA1NFEN14		—	—
TAUB1I15	C					DNFA1NFEN15		—	—
TAUB2I0	C	DNFA2CTL	FFC3 0200$_H$	C_ISO_TAUB2	DNFA2EN	DNFA2NFEN0	FFC3 0204$_H$	—	—
TAUB2I1	C					DNFA2NFEN1		—	—
TAUB2I2	C					DNFA2NFEN2		—	—
TAUB2I3	C					DNFA2NFEN3		—	—
TAUB2I4	C					DNFA2NFEN4		—	—
TAUB2I5	C					DNFA2NFEN5		—	—
TAUB2I6	C					DNFA2NFEN6		—	—
TAUB2I7	C					DNFA2NFEN7		—	—
TAUB2I8	C					DNFA2NFEN8		—	—
TAUB2I9	C					DNFA2NFEN9		—	—
TAUB2I10	C					DNFA2NFEN10		—	—
TAUB2I11	C					DNFA2NFEN11		—	—
TAUB2I12	C					DNFA2NFEN12		—	—
TAUB2I13	C					DNFA2NFEN13		—	—
TAUB2I14	C					DNFA2NFEN14		—	—
TAUB2I15	C					DNFA2NFEN15		—	—
TAUJ0I0	C	DNFA3CTL	FFC3 0300$_H$	C_ISO_TAUJ	DNFA3EN	DNFA3NFEN0	FFC3 0304$_H$	—	—
TAUJ0I1	C					DNFA3NFEN1		—	—
TAUJ0I2	C					DNFA3NFEN2		—	—
TAUJ0I3	C					DNFA3NFEN3		—	—

Table 2.155 List of Input Pins with a Noise Filter (3/3)

Input Pin	Filter Type	Digital Noise Elimination Control Register			Digital Noise Elimination Enable Register			Filter Control Register	
		Control Register	Address	Input Clock	Control Register	Control Bit	Address	Control Register	Address
CSIG0SC	C	DNFA5CTL	FFC3 0500$_H$	CLKJIT	DNFA5EN	DNFA5NFEN0	FFC3 0504$_H$	—	—
CSIG0SI	C					DNFA5NFEN1		—	—
CSIG0SSI	C					DNFA5NFEN2		—	—
CSIG0RYI	C					DNFA5NFEN3		—	—
CSIG1SC	C					DNFA5NFEN4		—	—
CSIG1SI	C					DNFA5NFEN5		—	—
CSIG1SSI	C					DNFA5NFEN6		—	—
CSIG1RYI	C					DNFA5NFEN7		—	—
CSIG2SC	C					DNFA5NFEN8		—	—
CSIG2SI	C					DNFA5NFEN9		—	—
CSIG2SSI	C					DNFA5NFEN10		—	—
CSIG2RYI	C					DNFA5NFEN11		—	—
CSIG3SC	C					DNFA5NFEN12		—	—
CSIG3SI	C					DNFA5NFEN13		—	—
CSIG3SSI	C					DNFA5NFEN14		—	—
CSIG3RYI	C					DNFA5NFEN15		—	—
CSIH0SC	C	DNFA6CTL	FFC3 0600$_H$	CLKJIT	DNFA6EN	DNFA6NFEN0	FFC3 0604$_H$	—	—
CSIH0SI (P0_2)	C					DNFA6NFEN1		—	—
CSIH0SSI	C					DNFA6NFEN2		—	—
CSIH0RYI	C					DNFA6NFEN3		—	—
CSIH1SC	C					DNFA6NFEN4		—	—
CSIH1SI	C					DNFA6NFEN5		—	—
CSIH1SSI	C					DNFA6NFEN6		—	—
CSIH1RYI	C					DNFA6NFEN7		—	—
CSIH0SI (P1_4)	C					DNFA6NFEN8		—	—
ADCE0TRG1	C	DNFA7CTL	FFC3 0700$_H$	C_ISO_ADCE	DNFA7EN	DNFA7NFEN0	FFC3 0704$_H$	—	—
ADCE0TRG2	C					DNFA7NFEN1		—	—
ADCE0TRG3	C					DNFA7NFEN2		—	—
SSIF0SCK	C	DNFA8CTL	FFC3 0800$_H$	C_ISO_PCLK	DNFA8EN	DNFA8NFEN0	FFC3 0804$_H$	—	—
SSIF0WS	C					DNFA8NFEN1		—	—
SSIF0RXD	C					DNFA8NFEN2		—	—
SSIFACK	C					DNFA8NFEN3		—	—
SSIF1SCK	C					DNFA8NFEN4		—	—
SSIF1WS	C					DNFA8NFEN5		—	—
SSIF1RXD	C					DNFA8NFEN6		—	—

2.6.2 XOR Compare Unit Filters

The XOR Compare Unit output signals are passed through a digital noise filter.

Refer to Section 2.4.2, Port output check by XOR Compare Unit for details.

Table 2.156 List of XOR Compare Unit Filters

XOR Comapre Unit output signal	Filter Type	Digital Noise Elimination Control Register			Digital Noise Elimination Enable Register			Filter Control Register	
		Control Register	Address	Sampling Clock	Control Register	Control Bit	Address	Control Register	Address
TAUB0O1_XORIN	C	DNFA9CTL	FFC3 0900$_H$	C_ISO_TAUB01	DNFA9EN	DNFA9NFEN0	FFC3 0904$_H$	—	—
TAUB0O3_XORIN	C					DNFA9NFEN1		—	—
TAUB0O5_XORIN	C					DNFA9NFEN2		—	—
TAUB0O7_XORIN	C					DNFA9NFEN3		—	—
TAUB0O9_XORIN	C					DNFA9NFEN4		—	—
TAUB0O11_XORIN	C					DNFA9NFEN5		—	—
TAUB0O13_XORIN	C					DNFA9NFEN6		—	—
—	C					DNFA9NFEN7		—	—
TAUB1O1_XORIN	C					DNFA9NFEN8		—	—
TAUB1O3_XORIN	C					DNFA9NFEN9		—	—
TAUB1O5_XORIN	C					DNFA9NFEN10		—	—
TAUB1O7_XORIN	C					DNFA9NFEN11		—	—
TAUB1O9_XORIN	C					DNFA9NFEN12		—	—
PCMP0AP0_XORIN	C	DNFA10CTL	FFC3 0A00$_H$	CLKFIX	DNFA10EN	DNFA10NFEN0	FFC3 0A04$_H$	—	—
PCMP0AN0_XORIN	C					DNFA10NFEN1		—	—
PCMP0BP0_XORIN	C					DNFA10NFEN2		—	—
PCMP0BN0_XORIN	C					DNFA10NFEN3		—	—
PCMP0AP1_XORIN	C					DNFA10NFEN4		—	—
PCMP0AN1_XORIN	C					DNFA10NFEN5		—	—
PCMP0BP1_XORIN	C					DNFA10NFEN6		—	—
PCMP0BN1_XORIN	C					DNFA10NFEN7		—	—
SG0FAO, SG0FAOL (P3_7), SG0AO (P3_7)*1	C					DNFA10NFEN8		—	—

Note 1. ch8 of DNFA10 is for the logic OR-function of all activated XOR compare functions of SG0.

2.6.3 Filter Types

2.6.3.1 Analog Filter Type A Input Pin

The input pin of analog filter type A has only an analog filter.

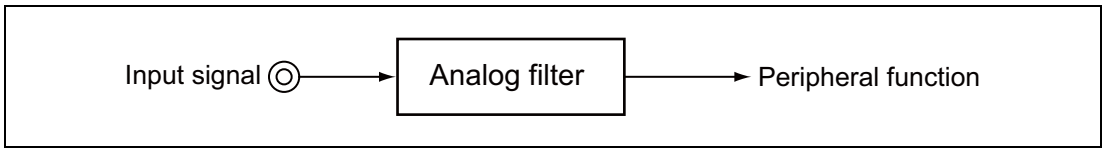


Figure 2.16 Block Diagram of Analog Filter Type A

2.6.3.2 Analog Filter Type B Input Pin

The input pin of analog filter type B has an analog filter, that is controlled with the following register.

- Filter control register FCLAnCTL
  Each FCLAnCTL control register controls filter processing.

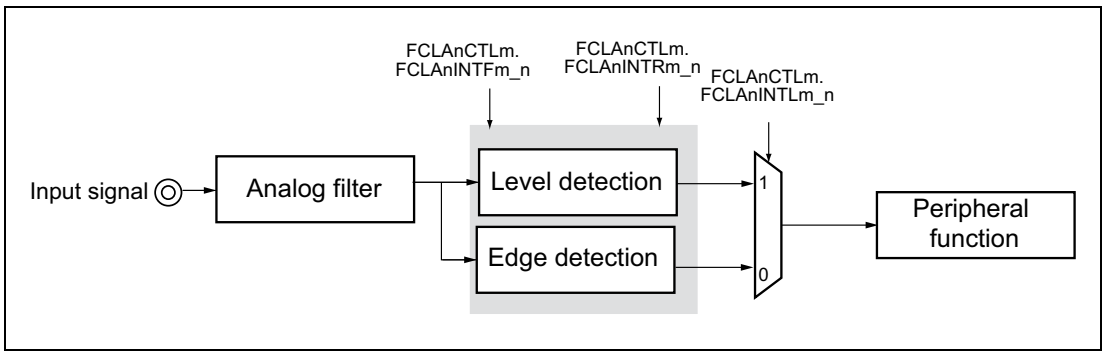


Figure 2.17 Block Diagram of Analog Filter Type B

2.6.3.3 Digital Filter Type C (No Edge Detection) Input Pin

The input pin of digital filter type C has a digital filter that is controlled with the following registers.

- Digital noise elimination control register DNFAnCTL
  Each DNFAnCTL control register controls digital filter processing.
- Digital noise elimination enable register DNFAnEN
  Each DNFAnEN enables digital filter processing.

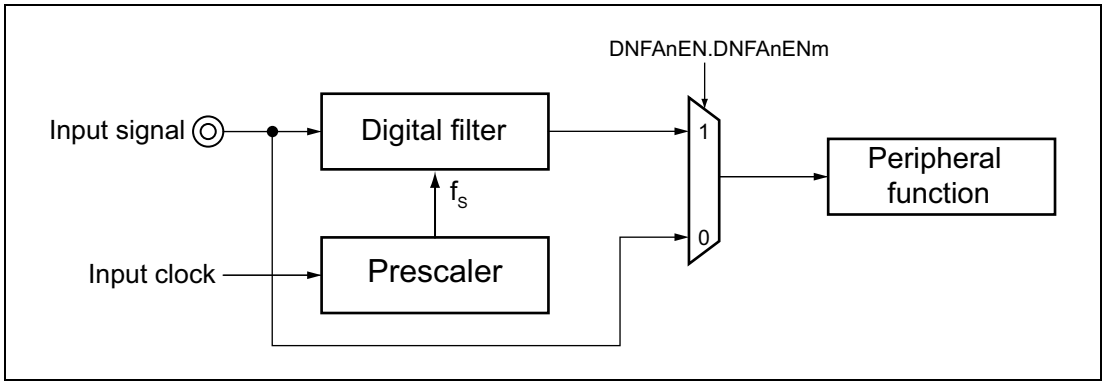


Figure 2.18 Block Diagram of Digital Filter Type C

CAUTIONS

1. **When the output signal from the digital filter is input to an alternative function, allow at least the following interval to elapse after the digital filter is enabled (DNFAnEN.DNFAnNFENm (m = 0 to 15) = 1) for the port pin to switch to the alternative function.**
   $s \times 1/fs + 2 \times 1/f_{DNFATCKI}$
   where
   s = DNFAnCTL.DNFAnNFSTS[1:0] + 2
   fs: sampling clock
   $f_{DNFATCKI}$: input clock
2. **When a digital filter's output signal is used as an interrupt signal, only enable the digital filter (DNFAnEN.DNFAnNFENm (m = 0 to 15) = 1) while interrupts are disabled. Furthermore, only enable interrupts after enabling the digital filter, waiting for the time below to elapse, and then clearing the interrupt request flag.**
   $s \times 1/fs + 3 \times 1/f_{DNFATCKI}$

2.6.4 Port Filter Registers

2.6.4.1 FCLAnCTLm — Filter Control Register

This register controls the analog filter operation.

Access: This register can be read or written in 8-bit units.

Address: The allocation of input signals to FCLAnCTLm registers and the address of each register are shown in the tables in Table 2.155, List of Input Pins with a Noise Filter and Table 2.156, List of XOR Compare Unit Filters..

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	FCLAnINTLm	FCLAnINTFm	FCLAnINTRm
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 2.157 FCLAnCTLm register contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When written, write the initial value.
2	FCLAnINTLm	Level detection control 0: Level detection disabled 1: Level detection enabled
1	FCLAnINTFm	Falling edge detection control 0: Falling edge detection disabled 1: Falling edge detection enabled
0	FCLAnINTRm	Rising edge detection control 0: Rising edge detection disabled 1: Rising edge detection enabled

Following table summarizes the correct register set-up for the various detection functions.

CAUTION

Before setting this register, make sure that the connected modules are disabled. Attempting to change

may lead to unexpected output from the filter.

Table 2.158 Register set-up for the various detection functions

FCLAnINTLm	FCLAnINTFm	FCLAnINTRm	Function
0	0	0	No detection
		1	Rising edge detection
	1	0	Falling edge detection
		1	Both edges detection
1	0	0	Low level detection
		1	High level detection
	1	0	Low level detection
		1	High level detection

NOTE

For details about the relevant port groups and bits, see Section 2.6.1, Allocation of Port Filters.

2.6.4.2 DNFAnCTL — Digital Noise Elimination Control Register

This register specifies characteristics of the digital noise elimination filter.

Access: This register can be read or written in 8-bit units.

Address: For the correspondence between the DNFAnCTL register and input signals, and the addresses of individual registers, see Table 2.155, List of Input Pins with a Noise Filter and Table 2.156, List of XOR Compare Unit Filters.

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	DNFAnNFSTS[1:0]		—	—	DNFAnPRS[2:0]		
Initial value	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R	R	R/W	R/W	R/W

Table 2.159 DNFAnCTL register contents

Bit Position	Bit Name	Function
7	Reserved	When written, write the initial value.
6, 5	DNFAnNFSTS [1:0]	The DNFAnNFSTS[1:0] bits specify the number of samples used to judge whether an external signal pulse is valid. 00_B: 2 samples 01_B: 3 samples 10_B: 4 samples 11_B: 5 samples
4, 3	Reserved	When written, write the initial value.
2 to 0	DNFAnPRS [2:0]	Selects a digital filter sampling clock. The sampling clock depends on the input clock (see Table 2.155, List of Input Pins with a Noise Filter) and the DNFAnPRS[2:0] divider: sampling clock = input clock / $2^{DNFAnPRS[2:0]}$

NOTE

For details about the relevant port groups and bits, see Section 2.6.1, Allocation of Port Filters and Section 2.6.2, XOR Compare Unit Filters.

CAUTION

Before setting this register, make sure that the connected modules are disabled. Attempting to change may lead to unexpected output from the filter.

2.6.4.3 DNFAnEN — Digital Noise Elimination Enable Register

This register enables or disables digital noise elimination for an arbitrary input signal.

Access: This register can be read or written in 16-bit units.

Address: For the correspondence between the DNFAnEN register and input signals, and the addresses of individual registers, see Table 2.155, List of Input Pins with a Noise Filter and Table 2.156, List of XOR Compare Unit Filters.

Initial value: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DNFAn NFEN15	DNFAn NFEN14	DNFAn NFEN13	DNFAn NFEN12	DNFAn NFEN11	DNFAn NFEN10	DNFAn NFEN9	DNFAn NFEN8	DNFAn NFEN7	DNFAn NFEN6	DNFAn NFEN5	DNFAn NFEN4	DNFAn NFEN3	DNFAn NFEN2	DNFAn NFEN1	DNFAn NFEN0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 2.160 DNFAnEN register contents

Bit Position	Bit Name	Function
15 to 0	DNFAnNFEN [15:0]	Digital noise elimination control 0: Digital noise elimination disabled (filter is bypassed) 1: Digital noise elimination enabled

NOTE

For details about the relevant port groups and bits, see Section 2.6.1, Allocation of Port Filters and Section 2.6.2, XOR Compare Unit Filters.

CAUTION

Before setting this register, make sure that the connected modules are disabled. Attempting to change may lead to unexpected output from the filter.

2.7 Pin State

This section specifies the status of all pins in the different operating modes.

2.7.1 Pin state in normal operation mode

Table 2.161 Pin state in normal operation mode (1/2)

Port	Pin Category	$\overline{\text{DCUTRST}}$ = L			
		$\overline{\text{RESET}}$ pin = L		$\overline{\text{RESET}}$ pin = H	
		During power-up (POC0RES = H)	Internal reset active (POC0RES = L)	No reset active	During DEEPSTOP
Pn	P0	Hi-Z	Hi-Z	Operate	Hold/operate
	All D1x: P1, P2, P3, P42, P43, P46 and P47 D1L1, D1L2(H), D1M1(H): P44 and P45	Hi-Z	Hi-Z	Operate	Hold/Hi-Z
	All D1x: P10, P11, P21 and P40 D1M2(H) only: P44 and P45 D1M1A only: P22	Hi-Z	Hi-Z	Operate	Hi-Z
	P16, P17	L	L	Operate	Hold/L
JP0*3		Hi-Z	Hi-Z	Operate	Operate
$\overline{\text{RESET}}$ (open drain)		L	L → H input	H input	H input
FLMD0 (input with internal pull-down)		L	L	L	L
X1/X2		Stop	Stop *1	Operate	Operate
XT1/XT2		Stop	Stop *1	Operate	Operate
PWRGD		Operate	Operate	Operate	Operate
PWRCTL		L	H	H	L
ADCE0Im		Hi-Z	Hi-Z	Operate	Hi-Z
SDR-SDRAM I/F (SDRA)	SDRAA[12:11], SDRAA10, SDRAA[9:0]	L	L	Operate*2	Hi-Z
	SDRADQ[31:0]	Hi-Z	Hi-Z	Operate	Hi-Z
	SDRABA[1:0]	L	L	Operate*2	Hi-Z
	SDRACLK	L	L	Operate	Hi-Z
	SDRACKE	L	L	Operate	Hi-Z
	SDRADM[3:0]	L	H	Operate	Hi-Z
	$\overline{\text{SDRACS}}$, $\overline{\text{SDRACAS}}$, $\overline{\text{SDRARAS}}$, $\overline{\text{SDRAWE}}$	L	H	Operate	Hi-Z
DDR2-SDRAM I/F (SDRB)	SDRBA[13:11], SDRBA10, SDRBA[9:0]	L	L	Operate	Hi-Z
	SDRBDQ0	Hi-Z	Hi-Z	Operate	Hi-Z
	SDRBDQ1	H	H	Operate	Hi-Z
	SDRBDQ[31:2], $\overline{\text{SDRBDQS}}$[3:0], SDRBDQS[3:0]	Hi-Z	Hi-Z	Operate	Hi-Z

Table 2.161 Pin state in normal operation mode (2/2)

Port	Pin Category	DCUTRST = L			
		RESET pin = L		RESET pin = H	
		During power-up (POC0RES = H)	Internal reset active (POC0RES = L)	No reset active	During DEEPSTOP
DDR2-SDRAM I/F (SDRB)	SDRBBA[2:0]	L	L	Operate	Hi-Z
	SDRBCLK, SDRBCLK	L	L	Operate	Hi-Z
	SDRBCKE	L	L	Operate	Hi-Z
	SDRBDM[3:0]	L	H	Operate	Hi-Z
	SDRBODT	L	L	Operate	Hi-Z
	SDRBCS, SDRBCAS, SDRBRAS, SDRBWE	L	H	Operate	Hi-Z

Note 1. X1/X2 and XT1/XT2 are only affected by the Power-On-Clear 0 (POC0RES = H). Thus, once it starts operation, internal reset does not change the status of these pins.

Note 2. Unused pin of SDRAA[12:0] and SDRABA[1:0] are driven to low while the operation. Refer to Table 15.28, 1 x 16-bit SDRAM (DBCONF.BWIDTH[1:0] = 01B) and Table 15.29, 1 x 32-bit or 2 x 16-bit SDRAM (DBCONF.BWIDTH[1:0] = 10B) for their availability.

Note 3. JP0_4 should be pulled down on RESET pin = L state.
When SYSRES and the external RESET is active, an on-chip pull-down resistor is connected to JP0_4.

2.7.2 Pin state in debug mode

Table 2.162 Pin state in debug mode (1/2)

Port	Pin Category	DCUTRST = H			
		RESET pin = L		RESET pin = H	
		During power-up (POC0RES = H)	Internal reset active (POC0RES = L)	No reset active	During DEEPSTOP
Pn	P0	Hi-Z	Hi-Z	Operate	Hold/operate
	All D1x: P1, P2, P3, P42, P43, P46 and P47 D1L1, D1L2(H), D1M1(H): P44 and P45	Hi-Z	Hi-Z	Operate	Hold/Hi-Z
	All D1x: P10, P11, P21 and P40 D1M2(H) only: P44 and P45 D1M1A only: P22	Hi-Z	Hi-Z	Operate	Hi-Z
	P16, P17	L	L	Operate	Hold/L
JP0		Operate	Operate	Operate	Operate
RESET (open drain)		L	L → H input	H input	H input
FLMD0 (input with internal pull-down)		L	L	L	L
X1/X2		Stop	Stop *1*2	Operate	Operate
XT1/XT2		Stop	Stop *1	Operate	Operate
PWRGD		Operate	Operate	Operate	Operate
PWRCTL		L	H	H	H
ADCE0Im		Hi-Z	Hi-Z	Operate	Hi-Z

Table 2.162 Pin state in debug mode (2/2)

Port	Pin Category	DCUTRST = H			
		RESET pin = L		RESET pin = H	
		During power-up (POC0RES = H)	Internal reset active (POC0RES = L)	No reset active	During DEEPSTOP
SDR-SDRAM I/F (SDRA)	SDRAA[12:11], SDRAA10, SDRAA[9:0]	L	L	Operate*3	Hi-Z
	SDRADQ[31:0]	Hi-Z	Hi-Z	Operate	Hi-Z
	SDRABA[1:0]	L	L	Operate*3	Hi-Z
	SDRACLK	L	L	Operate	Hi-Z
	SDRACKE	L	L	Operate	Hi-Z
	SDRADM[3:0]	L	H	Operate	Hi-Z
	$\overline{\text{SDRACS}}$, $\overline{\text{SDRACAS}}$, $\overline{\text{SDRARAS}}$, $\overline{\text{SDRAWE}}$	L	H	Operate	Hi-Z
DDR2-SDRAM I/F (SDRB)	SDRBA[13:11], SDRBA10, SDRBA[9:0]	L	L	Operate	Hi-Z
	SDRBDQ0	Hi-Z	Hi-Z	Operate	Hi-Z
	SDRBDQ1	H	H	Operate	Hi-Z
	SDRBDQ[31:2], SDRBDQS[3:0], $\overline{\text{SDRBDQS[3:0]}}$	Hi-Z	Hi-Z	Operate	Hi-Z
	SDRBBA[2:0]	L	L	Operate	Hi-Z
	SDRBCLK, $\overline{\text{SDRBCLK}}$	L	L	Operate	Hi-Z
	SDRBCKE	L	L	Operate	Hi-Z
	SDRBDM[3:0]	L	H	Operate	Hi-Z
	SDRBODT	L	L	Operate	Hi-Z
	$\overline{\text{SDRBCS}}$, $\overline{\text{SDRBCAS}}$, $\overline{\text{SDRBRAS}}$, $\overline{\text{SDRBWE}}$	L	H	Operate	Hi-Z

Note 1. X1/X2 and XT1/XT2 are only affected by the Power-On-Clear 0 (POC0RES = H). Thus, once it starts operation, internal reset does not change the status of these pins.

Note 2. The X1/X2 pins are operating, if it is used for LPD 1-pin mode (OPJTAG[1:0] = 10_B).

Note 3. Unused pin of SDRAA[12:0] and SDRABA[1:0] are driven to low while the operation. Refer to Table 15.28, 1 x 16-bit SDRAM (DBCONF.BWIDTH[1:0] = 01B) and Table 15.29, 1 x 32-bit or 2 x 16-bit SDRAM (DBCONF.BWIDTH[1:0] = 10B) for their availability.

2.7.3 Pin state in serial programming mode

Table 2.163 Pin state in serial programming mode (1/2)

Port	Pin Category	$\overline{\text{DCUTRST}}$ = L			
		$\overline{\text{RESET}}$ pin = L		$\overline{\text{RESET}}$ pin = H	
		During power-up (POC0RES = H)	Internal reset active (POC0RES = L)	No reset active	During DEEPSTOP
Pn	P0	Hi-Z	Hi-Z	Operate	Hold/operate
	All D1x: P1, P2, P3, P42, P43, P46 and P47 D1L1, D1L2(H), D1M1(H): P44 and P45	Hi-Z	Hi-Z	Operate	Hold/Hi-Z
	All D1x: P10, P11, P21 and P40 D1M2(H) only: P44 and P45 D1M1A only: P22	Hi-Z	Hi-Z	Operate	Hi-Z
	P16, P17	L	L	Operate	Hold/L
JP0*2		Hi-Z	Hi-Z	Operate	Operate
$\overline{\text{RESET}}$ (open drain)		L	L → H input	H input	H input
FLMD0 (input with internal pull-down)		L	H	H	H
X1/X2		Stop	Stop	Operate	Operate
XT1/XT2		Stop	Stop	Operate	Operate
PWRGD		Operate	Operate	Operate	Operate
PWRCTL		L	H	H	L
ADCE0Im		Hi-Z	Hi-Z	Operate	Hi-Z
SDR-SDRAM I/F (SDRA)	SDRAA[12:11], SDRAA10, SDRAA[9:0]	L	L	Operate*1	Hi-Z
	SDRADQ[31:0]	Hi-Z	Hi-Z	Operate	Hi-Z
	SDRABA[1:0]	L	L	Operate*1	Hi-Z
	SDRACLK	L	L	Operate	Hi-Z
	SDRACKE	L	L	Operate	Hi-Z
	SDRADM[3:0]	L	H	Operate	Hi-Z
	$\overline{\text{SDRACS}}$, $\overline{\text{SDRACAS}}$, $\overline{\text{SDRARAS}}$, $\overline{\text{SDRAWE}}$	L	H	Operate	Hi-Z

Table 2.163 Pin state in serial programming mode (2/2)

Port	Pin Category	DCUTRST = L			
		RESET pin = L		RESET pin = H	
		During power-up (POC0RES = H)	Internal reset active (POC0RES = L)	No reset active	During DEEPSTOP
DDR2-SDRAM I/F (SDRB)	SDRBA[13:11], SDRBA10, SDRBA[9:0]	L	L	Operate	Hi-Z
	SDRBDQ0	Hi-Z	Hi-Z	Operate	Hi-Z
	SDRBDQ1	H	H	Operate	Hi-Z
	SDRBDQ[31:2], SDRBDQS[3:0], $\overline{\text{SDRBDQS}}$[3:0]	Hi-Z	Hi-Z	Operate	Hi-Z
	SDRBBA[2:0]	L	L	Operate	Hi-Z
	SDRBCLK, $\overline{\text{SDRBCLK}}$	L	L	Operate	Hi-Z
	SDRBCKE	L	L	Operate	Hi-Z
	SDRBDM[3:0]	L	H	Operate	Hi-Z
	SDRBODT	L	L	Operate	Hi-Z
	$\overline{\text{SDRBCS}}$, $\overline{\text{SDRBCAS}}$, $\overline{\text{SDRBRAS}}$, $\overline{\text{SDRBWE}}$	L	H	Operate	Hi-Z

Note 1. Unused pin of SDRAA[12:0] and SDRABA[1:0] are driven to low while the operation. Refer to Table 15.28, 1 x 16-bit SDRAM (DBCONF.BWIDTH[1:0] = 01B) and Table 15.29, 1 x 32-bit or 2 x 16-bit SDRAM (DBCONF.BWIDTH[1:0] = 10B) for their availability.

Note 2. JP0_4 should be pulled down on RESET pin = L state.
When SYSRES and the external RESET is active, an on-chip pull-down resistor is connected to JP0_4.

2.8 Recommend connection of unused pins

This section specifies how to connect pins, which are not used in the application system.

Table 2.164 Recommended connection of unused pins (1/3)

Pin	Recommended connection, if not used	
X1	Must always be connected to oscillator	
X2		
XT1	Connect to OSCVSS	
XT2	Open or connect to OSCVSS	
FLMD0	Must always be connected	
P0_0/MODE0	The function of these pins depend on the operation mode.	
P0_1/FLMD1		
P0_2/MODE1		
PWRCTL	Open	
PWRGD	• D1Lx, D1M1(H), D1M1A: Connect to PWRCTL • D1M2(H): must always be connected	
$\overline{\text{RESET}}$	Pull-up resistor to EVCC	
Port group JP0	The function of these pins depend on the operation mode. When JP0_4 is unused, connect it to VSS through resistor.	Note: If the I/O ports reset configuration is changed, set • Hi-Z (input disabled, when PIBCn_m = 0 and PMCn_m = 0): leave open • input enabled (when PIBCn_m = 1 and PMCn_m = 1): connect to VSS through resistor. • output enabled (PMn_m = 0): leave open
Port group P0	Open	
Port group P1	Open	
Port group P2	Open	
Port group P3	Open	
Port group P10	Open	
Port group P11	Open	
Port group P16	Open	
Port group P17	Open	
Port group P21	Open	
Port group P22	Open	
Port group P40	Open	
Port group P42	Open	
Port group P43	Open	
Port group P44	Open	
Port group P45	Open	
Port group P46	Open	
Port group P47	Open	
SDRAA[12:0]	Open	
SDRABA[1:0]		
SDRACK		
SDRACKE		
SDRADM[3:0]		
$\overline{\text{SDRACS}}$		
$\overline{\text{SDRACAS}}$		
$\overline{\text{SDRARAS}}$		
$\overline{\text{SDRAWE}}$		
SDRADQ[31:0]		

CAUTION

For partial operation of SDR-SDRAM Memory Controller (e.g. usage with 16-bit SDR-SDRAM) refer to Section 15, SDR-SDRAM Memory Controller (SDRA).

Table 2.164 Recommended connection of unused pins (2/3)

Pin	Recommended connection, if not used
SDRBA[13:0]	Connect to SDRBVSS
SDRBBA[2:0]	
SDRBCAS	
SDRBCK	
$\overline{\text{SDRBCK}}$	
SDRBCKE	
SDRBCS	
SDRBDM[3:0]	
SDRBODT	
SDRBRAS	
SDRBWE	
SDRBDQ[31:0]	Connect to SDRBVSS ground
SDRBDQS[3:0]	Connect to SDRBVSS ground
$\overline{\text{SDRBDQS[3:0]}}$	Connect to SDRBVSS ground
REG0VCC	These power supply pins • must always be connected • must always be active.
EVCC	
REG0C	Voltage stabilization capacitor
ISOVDD	These power supply pins • must always be connected • voltage may be powered down in DEEPSTOP mode.
REG1VCC	
SDRAVCC	
OSCVCC	
A0VCC	
A0VREF	
B0VCC	
B1VCC	
B2VCC	
B3VCC	
B4VCC	
B5VCC	
SFVCC	
ZPDVCC	
ISMVCC	
MVCC	
PLLVCC	
RVCC	
ZPDVREF	Connect to ZPDVCC or ZPDVSS

Table 2.164 Recommended connection of unused pins (3/3)

Pin	Recommended connection, if not used
SDRBVCC	These power supply pins • connect to SDRBVSS ground, if not used • voltage may be be powered down in DEEPSTOP mode.
SDRBCKVCC	
SDRBVREF	**CAUTION** Ensure that supplies SDRBVCC, SDRBCKVCC and SDRBVREF are connected to SDRBVSS at the same time. For partial operation of DDR2-SDRAM Memory Controller (e.g. usage with 16-bit DDR2-SDRAM) refer to Section 16, DDR2-SDRAM Memory Controller (SDRB).
A0VSS	All ground pins must always be connected.
B0VSS	
B1VSS	
B2VSS	
B3VSS	
B4VSS	
EVSS	
SFVSS	
ZPDVSS	
ISMVSS	
ISOVSS	
MVSS	
OSCVSS	
PLLVSS	
REG0VSS	
REG1VSS	
RVSS	
SDRBVSS	
SDRBCKVSS	

Section 3 CPU System

The RH850/D1L/D1M CPU is an RH850 CPU of the architecture class V850E3v5-V, as described in the

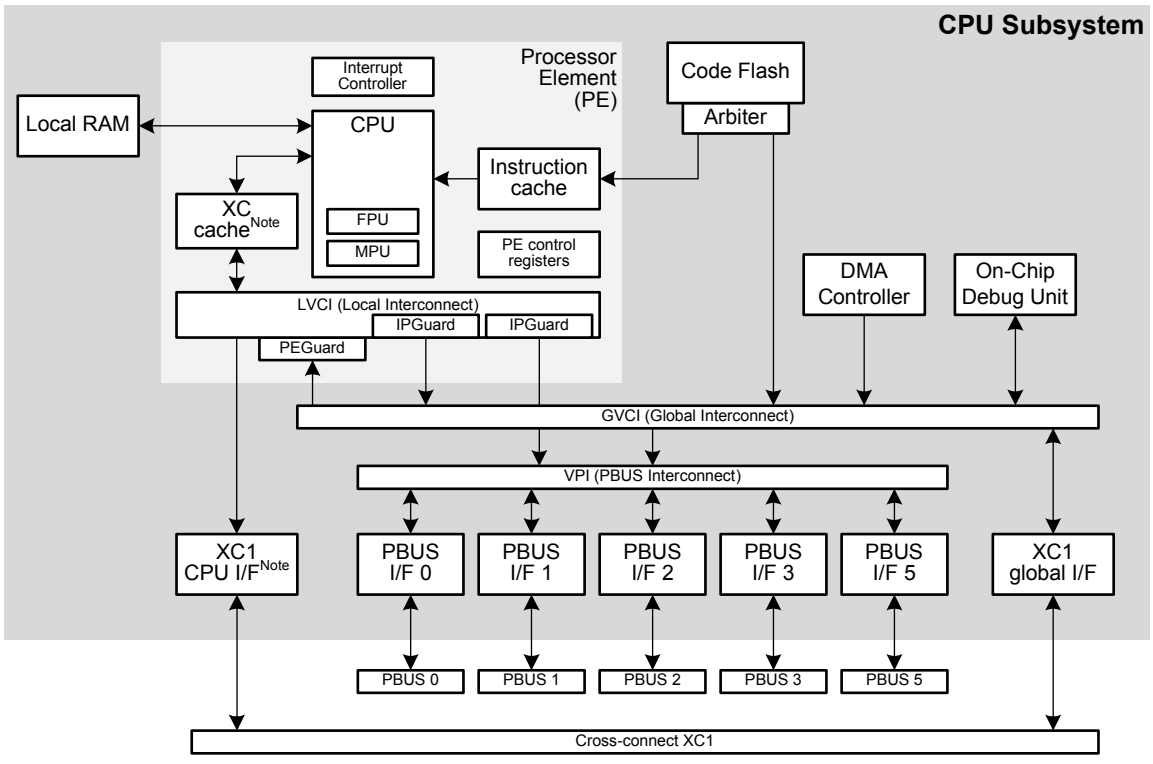
RH850 G3M
User's Manual: Software
Document number R01US0123EJxxxx (xxxx: revision number)

Of the functions as described in the above document the RH850/D1L/D1M products implement a subset as indicated in the following summary:

- MPU has 12 channels
- No MMU support
- No SIMD support

3.1 CPU Subsystem

The figure below shows the block diagram of the CPU Subsystem.



Note: The D1L1 devices do not have the XC cache and the XC1 CPU interface.

Figure 3.1 CPU Subsystem block diagram

3.1.1 Processor Element (PE) and CPU

The Processor Element (PE) comprises modules that are tightly coupled to the CPU. These modules operate with the CPU frequency and have dedicated protection facilities to prevent unwanted access.

The CPU features

- Single Processor Element with RH850G3M CPU core
- High performance 32-bit architecture for embedded control
- Separate 32-bit instruction and data bus
- 32-bit internal address bus: 4 GB linear space for both data and instructions
- 32 general-purpose registers, 32 bit wide
- Superscalar architecture with double 7 stage execution pipelines
- High speed branch structure with branch prediction and pre-fetch function
- Load/store units supports non-blocking load/store
- RISC type instruction set (backward compatible with V850, V850E1, and V850E2)
- Floating Point Unit (FPU)
  - Implemented as CPU coprocessor for execution of 32-bit single precision and 64-bit double precision floating-point operation instructions
  - Conforms to ANSI/IEEE standard 754 (IEEE binary floating-point operation standard)
- Code Flash cache
  - The cache resides in the instruction fetch pass of the CPU to the instruction flash memory
  - Size is 8 KB, 4-way associative, 128 lines for each way, with LRU replace algorithm
  - Cache operation is ECC protected
    - Double bit error detection, single bit error correction
- CPU protection functions:
  - supervisor and user mode
  - Memory Protection Unit (MPU)
- CPU snooze instruction:
  - Pauses CPU operation for 256 cycles per execution

Refer to Section 3.1.4, CPU protection functions for further details.

3.1.2 CPU Subsystem interfaces

All CPU Subsystem internal busses are multi-layered transaction based bus systems. Refer to Section 14.3.1, Transaction based busses for details.

3.1.2.1 XC1 interfaces

The CPU has access to all resources on the cross-connect XC1 via an exclusive CPU XC1 interface.

NOTE

This exclusive XC1 interface is not available for D1L1 devices.

The XC1 interfaces of following devices are equipped with a cache:

- D1L2(H), D1M1(H), D1M1-V2, D1M1A: 16 KB/4-way associative
- D1M2(H): 32 KB/4-way associative

All other CPU Subsystem master's XC1 accesses, e.g. the DMA Controller, are routed via the XC1 global interface.

A detailed description of the PBUS connections is given in Section 14.2, Cross-connect systems.

XC Cache

Refer to Section 3.3, XC Cache for description of the XC Cache.

3.1.2.2 PBUS interfaces

The PBUS is a single master bus, that is mainly used for accessing the control registers of most functional modules.

The PBUS modules are grouped into four functional segments, distributed to five separate PBUSes.

- System control PBUS segment
  connects modules to configure the system and the modules on the Always-On-Area, e.g.
  - System control: Interrupt Controller 2, Clock Controller, ICU-S, Error Control Module
  - Always-On-Area modules: Always-On-Area Timer, Real-Time Clock, Watchdog
- DMA PBUS segment
  connects modules mostly served by DMA e.g. like sound generator or IIS, e.g.
  - Audio interfaces: Sound Generator, I^2S interfaces, PCM-PWM Converter
  - LCD Bus Interface
- Graphics/communication PBUS segment
  connects to registers of modules, which are also attached to the cross-connections XC0, XC1, XC2, e.g.
  - video and graphics: Sprite Engine, Video Output Checker
  - advanced communication: Ethernet, Media Local Bus

- General PBUS segment
  connects all other modules, e.g.
  - timers: Timer Array Units, OS timers
  - standard communication: CAN, I2C, serial clocked and asynchronous interfaces (CSI, RLIN)
  - miscellaneous: A/D Converter, Stepper Motor Controller/Driver, PWM Generators and Diagnostic

Only the CPU and the DMA Controller can access the PBUS modules.

Each PBUS segment has its own guard, that allows to grant or prohibit certain access rights to dedicated PBUS masters.

A detailed description of the PBUS connections is given in Section 14.5, PBUS structure.

3.1.3 Bit manipulation

The RH850/G3M CPU core offers the following bit manipulation commands:

- CLR1, NOT1, SET1, TST1 → byte access used
- CAXI → word (32-bit) access used

Byte access can only be used if supported by

- the bus system
- the functional module

On bus level these commands result in a read-modify-write (RMW operation).

- Some buses can ensure this sequence is not interrupted by other access (bus is locked)
- If the bus or functional module cannot handle, the software needs to ensure that the register is not accessed in parallel by another master (e.g. CPU vs. DMA)

CAUTION

The application software must not use bit manipulation commands, if the register value can change as a result of operation of the functional module and read-modify-write access could overwrite such change.

Table 3.1 Bit manipulation support

	Target access via		
	PE internal buses*1	**APB**	**XC1 cross-connect**
Support of 8-bit, 16-bit access	Yes*2	Yes*2	Yes*2
Support of bus lock during RMW operations	Yes	Yes	No
Usage of SET1, CLR1, NOT1, TST1	Yes	Yes	No*3
Usage of CAXI	Yes	Yes	No*3

Note 1. All Processor Element internal buses.

Note 2. If the target register allows 8-bit/16-bit access.

Note 3. The usage of these commands is prohibited, when targeting registers via the XC1 cross-connect. The resulting read/write operations would be ignored.

CAUTION

The registers of the following modules are accessed via the XC1 cross-connect:

- **Serial Flash Memory Interface (SFMA)**
- **Video Data Controllers (VDCE)**
- **Video Output Warping Engine (VOWE)**
- **Display Output Comparators (DISCOM)**
- **JPEG Codec Unit A (JCUA)**
- **SDR-SDRAM Memory Controller (SDRA)**
- **HyperBus Controller (HYPB)**
- **OctaBus Controller (OCTA)**

Thus SET1, CLR1, NOT1, TST1 and CAXI operations must not be used for accessing the registers of these modules.

Note for memory access

The support of bit manipulation commands also affects data access in memory.

If the application software defines bit structures in memory, the compiler will access such data with bit manipulation. Consequently the bus connection of this memory must support bit manipulation commands.

In particular this means:

- Access to the Local RAM in the CPU Subsystem can use SET1, CLR1, TST1 and CAXI commands.
- Access to on-chip Video RAM (VRAM), Retention RAM (RRAM) and external SDRAM must not use SET1, CLR1, TST1 and CAXI commands. The read/write will be ignored.
  Note that no exception is issued in such case.

3.1.4 CPU protection functions

The CPU has the following means to protect the system against software malfunctions:

CPU Operating Modes

The CPU core provides three operating modes for the software:

- User mode
- Supervisor mode

The supervisor mode typically is granted to the operating system. Instructions required for modifications of the CPU core's system registers are only available in supervisor mode.

Applications managed by an operating system (or similar software instances) are run in user mode. Access to system resources or memory areas can be restricted. In case of a software malfunction non application related memory areas are protected against illegal access.

Memory Protection Unit (MPU)

The MPU is used to protect CPU memory regions against instruction fetches by illegal application software and or illegal data manipulation.

- The MPU supervises the whole address range of the CPU
  - Address range includes both memory and peripheral area
  - Allowed areas can be configured by software
  - Any unpermitted access will assert a MPU exception
- The MPU is integrated in the CPU core
  - Only CPU access (data and instruction) are supervised
  - Pipeline precise exceptions will be generated
  - Only if data/instruction are really used – i.e. speculative instructions fetch will not be handled if dropped
- MPU configuration
  - 12 protection windows configurable by CPU in supervisor mode
  - Granularity of windows is one word – areas may overlap

3.1.5 System Error Notification Control Function (SEG)

Errors due to an instruction fetch or data access can be the sources of system error exceptions. A SYSERR exception is an FE level exception from which return or recovery is not possible. SEG (SysErrGen) controls the notification and record of the error by the data access. Multiple error occurrence inputs are categorized according to error factors, and are processed sequentially from the highest-priority error factor, generating an FE-level asynchronous exception (SYSERR). The bit position of the SEGFLAG register becomes the priority of error factors. Error factors of lower bits take precedence over error factors of upper bits. Error information is recorded once regardless of error frequency. The error with the highest priority of error factor (in case errors occurred simultaneously) is valid. Recorded error information is not overwritten by subsequent errors.

NOTE

Be sure to place a SYNCP instruction at the start of handers for exceptions of set E.

- Exceptions of set E: FENMI, FEINT, EIINT (with the direct vector method), SYSERR, and FPI

For details, see "*RH850G3M User's Manual Software Edition*".

3.1.5.1 List of SEG Function Control Registers

Table 3.2 Base Address of SEG Register: FFFE E980$_H$

Address Offset	Size (Byte)	Register Name	Symbol	Right	R/W	OperableBit				Value after reset
						1	8	16	32	
0000_H	2	Errornotification control register	SEGCONT	—	R/W*1	—	—	√	—	0000_H
$+02_H$	2	Error occurrence retentionregister	SEGFLAG	—	R/W*1	—	—	√	—	0000_H
$+08_H$	4	Errorfactor retention register (address)	SEGADDR	—	R/W*1	—	—	√	√	Undefined (retained)

Note 1. Write accesses from other than the SV privilege (PSW.UM = 0) on the native machine are ignored.

NOTE

- If an access is made with an address offset or operable bits other than those specified above, an error response is returned.
- Write access is only possible in supervisor mode (UM = 0). Attempting to write in other modes leads to an error response being returned.
- No restriction is provided for read accesses.
- Read accesses to ranges permitted by other protection systems are enabled at any time.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> is defined as “SEG” for the registers in the table above.

<Symbol> is defined in the above table

3.1.5.2 Register set

(1) SEGCONT — Error Notification Control Register

This register is used to enable (= 1) or disable (= 0) notification of SYSERR request in response to error flags that store error occurrence status according to factors.

Access: This register can be read/written in 16-bit units.

Address: $+00_H$

Initial value: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	VPGE	VCRE	—	TCME	ROME	VCIE	—	ICCE	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R	R

Table 3.3 SEGCONT register contents (1/2)

Bit Position	Bit Name	Function
15to 10	Reserved	When read, the value after reset is read. When writing, write the value after reset.
9	VPGE	• This bit notifies of a response to an error in PBUS0-5. (excluding errors in readingfrom PBUS.) The error includes the followings: – When an unimplemented area is accessed – PBUS guard error(PBUS guarding also applies to the guarding of all registers in the modules for protection listed in Table 14.18)
8	VCRE	This bit notifies of IPG violation access detection and subsequent access blocking*2.
7	Reserved	When read, the value after reset is read. When writing, write the value after reset.
6	TCME	This bit notifies of an error during data accessing to its own local RAM. The error includes the following cases: • When an ECC error which cannot be corrected occurs • When an access to the RAM-unimplemented area in the local RAM is detected
5	ROME	This bit notifies of an error in access to the Code Flash when a table reference is read in response to a table reference interrupt. The error includes the following cases: • When an ECC error which cannot be corrected occurs

Table 3.3 SEGCONT register contents (2/2)

Bit Position	Bit Name	Function
4	VCIE	• This bit notifies of a response to an error in PBUS0-5.(excluding errors in writing to PBUS.) The error includes the following cases: – When an unimplemented areais accessed – PBUS guard error(PBUS guarding also applies to the guarding of all registers in the modules for protection listed in Table 14.18) • This bit notifies of a response to an error in the Code Flash The error includes the following cases: – When an ECC error which cannot be corrected occurs*1 • This bit notifies of a detection of access to an interconnect reserved area. – FFFF 0000_H to FFFF $4FFF_H$ – FFFE 0000_H to FFFE $BFFF_H$ – FB00 0000_H to FE9F $FFFF_H$ – F300 0000_H to F8FF $FFFF_H$ • This bit notifies of a IPG violation access detection and subsequent access blocking*2. • This bit notifies of an access privilege violation. – Read or write access to the IPG Protection Setting Registers by user mode (PSW.UM = 1) – Write access to the SEG Function Control Registers by user mode (PSW.UM = 1)
3	Reserved	When read, the value after reset is read. When writing, write the value after reset.
2	ICCE	Instruction cache error notification enable The error occurred in the instruction cache is handled when the instructioncache system register, ICCTRL.ICHEMK, is set to 0 (whose value after reset is 1): For instruction cache errors, see "*ICERR — Instruction Cache Error Register*" in "*RH850G3M User's Manual Software Edition*".
1 to 0	Reserved	When read, the value after reset is read. When writing, write the value after reset.

Note 1. While ECC error notification is enabled, an uncorrectable ECC error has occurred. Whether or not the error can be corrected depends on the ECC settings.

Note 2. For error factor address, see Section 14.6.3, PE's Internal Peripheral Guard (IPG), IPGADRUM register.

Table 3.4 Error Factor Codes and Handling of G3M Core SYSERR Exceptions

Factor Codes	Error Contents
10_H	Reserved
11_H	Instruction fetch errors (from code flash memory)
12_H	Errors in which their notification is enabled by the SEGCONT second bit
13_H	Instruction fetch errors (from other than code flash memory)
14_H	Errors in which their notification is enabled by the SEGCONT fourth bit
15_H	Errors in which their notification is enabled by the SEGCONT fifth bit
16_H	Errors in which their notification is enabled by the SEGCONT sixth bit
17_H	Reserved
18_H	Errors in which their notification is enabled by the SEGCONT eighth bit
19_H	Errors in which their notification is enabled by the SEGCONT ninth bit
$1A_H$	Reserved
$1B_H$	Reserved
$1C_H$	Reserved
$1D_H$	Reserved
$1E_H$	Reserved
$1F_H$	Reserved

3.1.5.3 SEGFLAG — Error Occurrence Retention Register

- This register sets the flags of error occurrence, that store each error occurrence status by factors. Those are not automatically cleared (return to 0).
- Writing to the register enables both setting and clearing.

Access: This register can be read/written in 16-bit units.

Address: $+02_H$

Initial value: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	VPGF	VCRF	—	TCMF	ROMF	VCIF	—	ICCF	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R	R

Table 3.5 SEGFLAG register contents

Bit Position	Bit Name	Function
15 to 10	Reserved	When read, the value after reset is read. When writing, write the value after reset.
9	VPGF	Flag corresponding to bit 9 of the SEGCONT register
8	VCRF	Flag corresponding to bit 8 of the SEGCONT register
7	Reserved	When read, the value after reset is read. When writing, write the value after reset.
6	TCMF	Flag corresponding to bit 6 of the SEGCONT register
5	ROMF	Flag corresponding to bit 5 of the SEGCONT register
4	VCIF	Flag corresponding to bit 4 of the SEGCONT register
3	Reserved	When read, the value after reset is read. When writing, write the value after reset.
2	ICCF	Flag corresponding to bit 2 of the SEGCONT register
1 to 0	Reserved	When read, the value after reset is read. When writing, write the value after reset.

3.1.5.4 SEGADDR — Error Factor Retention Register (Address)

This register records information of an error factor that notifies a SYSERR request (only one history is recorded).

It records the addresses where the error factors for the VCIF, ROMF and TCMF bits of the SEGFLAG register were found. The error factors except VCIF, ROMF and TCMF bits of the SEGFLAG register to be recorded as 00000000_H.

Access: This register can be read/written in 32-bit units.

Address: $+08_H$

Initial value: Undefined (retained)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDRESS[31:16]															
Value after reset	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDRESS[15:0]															
Value after reset	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: ×: Undefined (retained)

Table 3.6 SEGADDR register contents

Bit Position	Bit Name	Function
31 to 0	ADDRESS[31:0]	These bits retain the address at which SYSERR source occurred. (If an error occurred by accessing to the local RAM area, the addresses of the 19 lower-order bits are retained and the addresses of the 13 higher-order bits are cleared to 0.)

CAUTION

This cannot be modified when the error occurrence flag to enable notification is set.

3.1.5.5 SEG Function

(1) SEG function: Notifying a SYSERR request due to an error flag

- Setting an error flag takes precedence over clearing the same flag.
  - Simultaneous clearing operation is ignored.
- Priority of error factors
  - The bit position of the notification-enabled SEGFLAG register becomes the priority of error factors. Error factors of lower bits take precedence over error factors of upper bits. Notification is made from the highest-priority error factor.
  - The bit position of error factors is notified as a “SYSERR factor code”.
- Conditions for starting SYSERR request notification

- Even if a notification-disabled flag is set to 1, notification is not made.
- Notification is made immediately after a notification-enabled flag is set to 1.
- After clearing, notification is made depending on the flag state (re-arbitration).

- Finishing notification at a SYSERR request response
  - Even after notification is finished, the flag is not cleared automatically.
  - Notification is not made until re-arbitration is performed by setting or clearing the flag.
  - If an error flag that is prioritized higher than with an upper SYSERR factor code.

(2) SEG function: Recording error factor information

- When notification-enabled error occurrence is input, the error address is retained in the above register.
  - No information is retained by setting or clearing an error flag described in "(1) SEG function: Notifying a SYSERR request due to an error flag in **Section 3.1.5.5, SEG Function**" above.
  - When multiple error occurrence inputs are present simultaneously, information other than the prioritized error factor is not retained.
- While the notification-enabled error flag described in "(1) SEG function: Notifying a SYSERR request due to an error flag in Section 3.1.5.5, SEG Function" above is set to 1, overwrite to the above register is inhibited
  - If error occurrence input continues, information of subsequent error factors is not retained.
  - To reset the inhibition of overwrite to the register, clear either SEGCONT or SEGFLAG register (or both of them).

(3) Supplementary information on SYSERR exception

- The PSW.EBV bit retains its value and the base address for the exception handler is not changed even when a SYSERR exception is generated.
- Detection of errors in the instruction cache

  Detection of an error in the instruction cache does not lead to a SYSERR exception of the type that requires re-execution by the origin of the instruction fetch. Execution of the instruction by the CPU still proceeds; the instruction cache automatically invalidates the entry which lead to the error and fetching is repeated, this time from the code flash memory. The SEG block will be informed that a cache error has occurred if the setting of the ICHEMK bit of the ICCTRL cache-control system register is 0.

3.1.6 Product dependent registers

Table 3.7 List of product dependent registers

Register No. (reg ID, sel ID)	Register name	Symbol	Access Permission
SR5, 1	Machine control register	MCTL	SV (PSW.UM = 0)
SR6, 1	Processor ID register	PID	SV (PSW.UM = 0)

3.1.6.1 MCTL — Machine Control Register

This register is used to control the CPU.

Access: —

Address: —

Initial value: 8000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UIC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 3.8 MCTL register contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
30 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	UIC	This bit is used to control the interrupt enable/disable operation in user mode. When this bit is set to 1, executing the EI/DI instructionbecomes possible in user mode.

3.1.6.2 PID — Processor ID Register

The PID register retains a processor identifier that is unique to the CPU. The PID register is a read-only register.

CAUTION

The PID register indicates information used to identify the incorporated CPU core and CPU core configuration. Usage such that the software behavior varies dynamically according to the PID register information is not assumed.

Access: —

Address: —

Initial value: 0580 0F12$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PID[31:16]															
Value after reset	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PID[15:0]															
Value after reset	0	0	0	0	1	1	1	1	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.9 PID register contents

Bit Position	Bit Name	Function
31 to 24	PID	Architecture Identifier This identifier indicates the architecture of the processor.
23 to 8		Function Identifier This identifier indicates the functions of the processor. These bits indicate whether or not functions defined per bit are implemented (1: implemented, 0: not implemented). Bits 23 to 11:Reserved Bit 10: Double-precision floating-point operation function Bit 9: Single-precision floating-point operationfunction Bit 8: Memory protection unit function (MPU)
7to 0		Version Identifier This identifier indicates the version of the processor.

3.2 Instruction Cache ECC

The ECC of the Instruction Cache generates an interrupt request listed in the following table:

Table 3.10 Interrupt Request by ECC of Instruction Cache

Function	Connected to
Instruction cache data RAM ECC 2-bit error detection interrupt	These interrupts are logically OR combined and input to the Error Control Module INTECCI.
Instruction cache data RAM ECC 1-bit error detection interrupt	
Instruction cache tag RAM ECC 2-bit error detection interrupt	
Instruction cache tag RAM ECC 1-bit error detection interrupt	

Instruction caches ECC calculated following data size for each RAM.

- Instruction cache data RAM: The 8-bit ECC value is calculated for each 64-bit transfer.
- Instruction cache tag RAM: The 7-bit ECC value is calculated for each 32-bit transfer.

The ECC of the Instruction Cache and its registers are initialized by the following reset signal:

Table 3.11 Reset sources

ECC unit	Reset signal
Instruction Cache ECC	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

3.2.1 Overview

The instruction cache ECC is summarized in the table below.

Item	Description
ECC error detection	ECC error detection and correction can be either enabled or disabled. When enabled, • For instruction cache data RAM, ECC error detection and correction are carried out (2-bit error detection and 1-bit error detection and correction are carried out). • For instruction cache tag RAM, ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled, and notification are carried out.

Item	Description
Error notification	Upon occurrence of an ECC error, it is notified to the Error Control Module. **ECC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, notification of the ECC 2-bit error is disabled, and notification of the ECC 1-bit error is disabled.
Error status	A status register is provided, which indicates the status of ECC 2-bit error detection and ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Address capture	If an ECC error occurs while no error status is set, the address at which the associated error has occurred is captured. The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected. The error status serves as the enable bit of the capture address.

3.2.2 List of Registers

Table 3.12 List of Registers

Address	Symbol	Register Name	R/W	Value after reset	Access Size
FFC6 0400$_H$	IDECCCTL_PE1_OS	Instruction cache data RAM ECC control register	R/W	0000 0000$_H$	16/32
FFC6 0404$_H$	IDERRINT_PE1_OS	Instruction cache data RAM error information control register	R/W	0000 0000$_H$	8/16/32
FFC6 0408$_H$	IDSTCLR_PE1_OS	Instruction cache data RAM error status clear register	W	0000 0000$_H$	8/16/32
FFC6 040C$_H$	IDOVFSTR_PE1	Instruction cache data RAM error count overflow status register	R	0000 0000$_H$	8/16/32
FFC6 0410$_H$	ID1STERSTR_PE1_OS	Instruction cache data RAM 1st error status register	R	0000 0000$_H$	8/16/32
FFC6 0450$_H$	ID1STEADR0_PE1_OS	Instruction cache data RAM (Bank0) 1st error address register	R	0000 0000$_H$	8/16/32
FFC6 0454$_H$	ID1STEADR1_PE1_OS	Instruction cache data RAM (Bank1) 1st error address register	R	0000 0000$_H$	8/16/32
FFC6 1400$_H$	ITECCCTL_PE1_OS	Instruction cache tag RAM ECC control register	R/W	0000 0000$_H$	16/32
FFC6 1404$_H$	ITERRINT_PE1_OS	Instruction cache tag RAM error information control register	R/W	0000 0003$_H$	8/16/32
FFC6 1408$_H$	ITSTCLR_PE1_OS	Instruction cache tag RAM error status clear register	W	0000 0000$_H$	8/16/32
FFC6 140C$_H$	ITOVFSTR_PE1_OS	Instruction cache tag RAM error count overflow status	R	0000 0000$_H$	8/16/32
FFC6 1410$_H$	IT1STERSTR_PE1_OS	Instruction cache tag RAM 1st error status clear register	R	0000 0000$_H$	8/16/32
FFC6 1450$_H$	IT1STEADR0_PE1_OS	Instruction cache tag RAM (Bank0) 1st error address register	R	0000 0000$_H$	8/16/32

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> is defined as “ECCIC1” for the registers in the table above.

<Symbol> is defined in the above table.

3.2.3 Details of Registers

3.2.3.1 IDECCCTL_PE1_OS — Instruction cache data RAM data ECC control register

IDECCCTL enables or disables ECC error detection and correction and 1-bit error correction for cache data RAM. Set PROT[1:0] bits to 01_B when writing to IDECCCTL.

IDECCCTL is initialized by an internal reset or an external reset.

Access: This register can be read/written in 16-bit and 32-bit units.

Address: FFC6 0400_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT[1:0]		—	—	—	—	—	—	—	—	—	—	—	—	—	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 3.13 IDECCCTL_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, these bits are always read as 0. When writing, always write 0.
15, 14	PROT[1:0]	Enables or disables modification of the ECCDIS bits. The value written is not retained. These bits are always read as 0. Set PROT[1:0] = 01_B when writing to IDECCCTL.
13 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	ECCDIS	ECC disable bit Setting ECC error detection to enable/disable. 0:ECC error detection is enable 1:ECC error detection is disable

3.2.3.2 IDERRINT_PE1_OS — Instruction cache data RAM error information control register

IDERRINT enables or disables generation of the error notification signal to the ECM upon detection of an ECC 2-bit error or an ECC 1-bit error in cache data RAM.

IDERRINT is initialized by an internal reset or an external reset.

Access: This register can be read/written in 8-bit, 16-bit and 32-bit units.

Address: FFC6 0404_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 3.14 IDERRINT_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	DEDIE	ECC 2-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

3.2.3.3 IDSTCLR_PE1_OS — Instruction cache data RAM error status clear register

IDSTCLR clears the error flags in the error status register (ID1STERSTR), the overflow flag in the error overflow status register (IDOVFSTR), and the error address register (ID1STEADR). IDSTCLR is a write-only register and is always read as 0.

Access: This register can be written in 8-bit, 16-bit and 32-bit units.

Address: FFC6 0408$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SST CLR1	SST CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 3.15 IDSTCLR_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	SSTCLR1	Error Overflow Flag Clear (for bank 1) Writing 1 to this bit clears the DEDF1and SEDF1 flags in ID1STERSTR; ERROVF1 flag in IDOVFSTR; and ID1STEADR1.
0	SSTCLR0	Error Overflow Flag Clear (for bank 0) Writing 1 to this bit clears the DEDF0and SEDF0 flags in ID1STERSTR; ERROVF0 flag in IDOVFSTR; and ID1STEADR0.

3.2.3.4 IDOVFSTR_PE1_OS — Instruction cache data RAM error count overflow status register

IDOVFSTR monitors occurrence of error overflow in cache data RAM. If the second error occurs after the first error (= while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set. ERROVF is cleared by an internal reset, an external reset, or setting the STCLR bit to 1 in IDSTCLR.

Access: This register can be read in 8-bit, 16-bit and 32-bit units.

Address: FFC6 040C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR OVF1	ERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.16 IDOVFSTR_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	ERROVF1	Error Overflow Flag (for bank 1) ERROVF1 is set if the second error occurs while any of the error flags (DEDF1 and SEDF1) in the error status register is set, except when both of the error address and source of the second error are the same as those of the first error.
0	ERROVF0	Error Overflow Flag (for bank 0) ERROVF0 is set if the second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register is set, except when both of the error address and source of the second error are the same as those of the first error.

3.2.3.5 ID1STERSTR_PE1_OS — Instruction cache data RAM 1st error status register

ID1STERSTR monitors occurrence of the first error in cache data RAM. The error status is set if an error occurs while all the error flags for the relevant banks are 0. The error flag is overwritten if an ECC 2-bit error occurs while the ECC 1-bit error monitor flag is set.

ID1STERSTR is cleared by an internal reset, an external reset, or setting the STCLR bit to 1 in IDSTCLR.

Access: This register can be read in 8-bit, 16-bit and 32-bit units.

Address: FFC6 0410$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DEDF1	SEDF1	—	—	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.17 ID1STERSTR_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, these bits are always read as 0. When writing, always write 0.
9	DEDF1	ECC 2-bit error Monitor Flag (for bank 1) 0: Cleared to 0 by setting the STCLR1 bit to 1 in IDSTCLR. 1: Indicates that an ECC 2-bit error has occurred while the error flag DEDF1 is 0.
8	SEDF1	ECC 1-bit error Monitor Flag (for bank 1) 0: Cleared to 0 by setting the STCLR1 bit to 1 in IDSTCLR. 1: Indicates that an ECC 1-bit error has occurred while the error flags DEDF1 and SEDF1 are 0.
7 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	DEDF0	ECC 2-bit error Monitor Flag (for bank 0) 0: Cleared to 0 by setting the STCLR0 bit to 1 in IDSTCLR. 1: Indicates that an ECC 2-bit error has occurred while the error flag DEDF0 is 0.
0	SEDF0	ECC 1-bit error Monitor Flag (for bank 0) 0: Cleared to 0 by setting the STCLR0 bit to 1 in IDSTCLR. 1: Indicates that an ECC 1-bit error has occurred while the error flags DEDF0 and SEDF0 are 0.

3.2.3.6 ID1STEADRn_PE1_OS — Instruction cache data RAM (bank n) 1st error address register (n = 0, 1)

ID1STEADR holds the address at which an error has occurred in cache data RAM. The error address is set if an error occurs while all the error flags for the relevant banks are 0 in ID1STERSTR. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.

Since this register holds the internal address, add the base address of the associated memory to transform the internal address to the real address. ID1STEADR is cleared by an internal reset, an external reset, or setting the STCLR bit to 1 in IDSTCLR.

Access: This register can be read in 8-bit, 16-bit and 32-bit units.

Address: ID1STEADR0_PE1_OS: FFC6 0450$_H$
ID1STEADR1_PE1_OS: FFC6 0454$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	EADRn[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.18 ID1STEADRn_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, these bits are always read as 0. When writing, always write 0.
8 to 0	EADRn[8:0]	1st Error Address (for bank n) Monitors the address of the first error. The error address is set if an error occurs while all the error flags for bank n are 0 in ID1STERSTR. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.

3.2.3.7 ITECCCTL_PE1_OS — Instruction cache tag RAM ECC control register

ITECCCTL enables or disables ECC error detection in cache tag RAM. Set PROT[1:0] to 01_B when writing to ITECCCTL.

ITECCCTL is initialized by an internal reset or an external reset.

Access: This register can be read/written in 16-bit and 32-bit units.

Address: FFC6 1400_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT[1:0]		—	—	—	—	—	—	—	—	—	—	—	—	—	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 3.19 ITECCCTL_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, these bits are always read as 0. When writing, always write 0.
15, 14	PROT[1:0]	Enables or disables modification of the ECCDIS bit. The value written is not retained. These bits are always read as 0. Set PROT[1:0] = 01_B when writing to ITECCCTL.
13 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with setting PROT[1:0] = 01_B. 0: Enables ECC error detection. 1: Disables ECC error detection.

3.2.3.8 ITERRINT_PE1_OS — Instruction cache tag RAM error information control register

ITERRINT enables or disables generation of the error notification signal to the ECM upon detection of an ECC 2-bit error or an ECC 1-bit error in cache tag RAM.

ITERRINT is initialized by an internal reset or an external reset.

Access: This register can be read/written in 8-bit, 16-bit and 32-bit units.

Address: FFC6 1404_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 3.20 ITERRINT_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	DEDIE	ECC 2-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

3.2.3.9 ITSTCLR_PE1_OS — Instruction cache tag RAM error status clear register

ITSTCLR clears the error flags in the error status register (IT1STERSTR), the overflow flag in the error overflow status register (ITOVFSTR), and the error address register (IT1STEADR). ITSTCLR is a write-only register and is always read as 0.

Access: This register can be written in 8-bit, 16-bit and 32-bit units.

Address: FFC6 1408_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STCLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 3.21 ITSTCLR_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	STCLR0	Error Overflow Flag Clear Writing 1 to this bit clears the DEDF0 and SEDF0 flags in IT1STERSTR; ERROVF0 flag in ITOVFSTR; and IT1STEADR0.

3.2.3.10 ITOVFSTR_PE1_OS — Instruction cache tag RAM error count overflow status register

ITOVFSTR monitors occurrence of error overflow in cache tag RAM. If the second error occurs after the first error (= while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set. ERROVF is cleared by an internal reset, an external reset, or setting the STCLR bit to 1 in ITSTCLR.

Access: This register can be read in 8-bit, 16-bit and 32-bit units.

Address: FFC6 140C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.22 ITOVFSTR_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	ERROVF0	Error Overflow Flag ERROVF0 is set if the second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register is set, except when both of the error address and source of the second error are the same as those of the first error.

3.2.3.11 IT1STERSTR_PE1_OS — Instruction cache tag RAM 1st error status register

IT1STERSTR monitors occurrence of the first error in cache tag RAM.

The error flag is overwritten if an ECC 2-bit error occurs while the error flag is 0.

IT1STERSTR is cleared by an internal reset, an external reset, or setting the STCLR bit to 1 in ITSTCLR.

Access: This register can be read in 8-bit, 16-bit and 32-bit units.

Address: FFC6 1410$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.23 IT1STERSTR_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	DEDF0	ECC 2-bit error Monitor Flag 0: Cleared to 0 by setting the STCLR0 bit to 1 in ITSTCLR. 1: Indicates that an ECC 2-bit error has occurred while the error flag DEDF0 is 0.
0	SEDF0	ECC 1-bit error Monitor Flag 0: Cleared to 0 by setting the STCLR0 bit to 1 in ITSTCLR. 1: Indicates that an ECC 1-bit error has occurred while the error flags DEDF0 and SEDF0 are 0.

3.2.3.12 IT1STEADR0_PE1_OS — Instruction cache tag RAM 1st error address register

IT1STEADR holds the address at which an error has occurred in cache tag RAM. The error address is set if an error occurs while all the error flags for the relevant banks are 0 in IT1STERSTR. The address is updated if an ECC 1-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated. Since this register holds the internal address, add the base address of the associated memory to transform the internal address to the real address.

IT1STEADR is cleared by an internal reset, an external reset, or setting the STCLR bit to 1 in ITSTCLR.

Access: This register can be read in 8-bit, 16-bit and 32-bit units.

Address: FFC6 1450$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	EADR[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.24 IT1STEADR0_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, these bits are always read as 0. When writing, always write 0.
8 to 0	EADR[8:0]	1st Error Address Monitors the address of the first error. The error address is set if an error occurs while all the error flags are 0 in IT1STERSTR. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.

3.3 XC Cache

CPU access to the on-chip and external memories can be performed directly (un-cached) or via the XC Cache.

Therefore the

- Retention RAM (RRAM)
- on-chip Video RAMs (VRAM0 and VRAM1)
- external serial flash memory devices via the Serial Flash Memory Interface A (SFMA)
- external SDRAM memory devices via the SDR-SDRAM Memory Controller (SDRA) or DDR2-SDRAM Memory Controller (SDRB)
- external serial flash/RAM memory devices via the HyperBus (HYPB) Controller

can be accessed through different address windows.

For details about the direct and mirrored access areas refer to Section 5.3, Cross-connect address map and Section 54.5, Video RAM and SDRAM memory map.

3.3.1 XC Cache features summary

- The XC cache is not implemented in the D1L1 devices.
  - Unified cache i.e. both instruction and data cache
  - It allows caching of memories on the XC1 cross-connect, i.e. Video RAM (VRAM), external SDRAM, external serial flash
- Cache size:
  - D1L2(H), D1M1(H), D1M1-V2, D1M1A: 16 KB
  - D1M2(H): 32 KB
  - All: 4-way associative, 32 byte line size
- Least recently used (LRU) and write-back policy
- Cache operation is ECC protected
  - Double bit error detection, single bit error correction
  - XC caches ECC calculated following data size for each RAM.
    XC cache data RAM: The 7-bit ECC value is calculated for each 32-bit transfer.
    XC cache tag RAM: The 7-bit ECC value is calculated for each 19-bit transfer.

Item	Description
ECC error detection	ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected. • ECC error detection and correction are carried out (2-bit error detection and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled, and 1-bit error detection and correction and 2-bit error detection and notification are carried out.
Error notification	Upon occurrence of an ECC error, it is notified to the Error Control Module. **ECC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, notification of the ECC 1-bit and 2-bit errors is disabled.
Error status	A status register is provided, which indicates the statuses of ECC 2-bit error detection and ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Address capture	If an ECC error occurs while no error status is set, the address at which the associated error has occurred is captured. The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected. The error status serves as the enable bit of the capture address.

- Cached access configuration
  - Default mode is direct (un-cached) access
  - Two address windows can be defined for cached access
  - Minimum granularity is 1 MB
- Buffered mode available
  - Buffer area and Cache area cannot set to same address area. If address area are overlapped, XC cache issue the error response.
  - Read buffer: 64 bit x 8 entries
  - Write buffer: 64 bit x 2 entries
  - Buffer flash function available
- Cache operation commands available (by register access)

3.3.2 Cache/Buffer window set-up

- Base address of cached window = AXCCABASEn
- Window size of cached window = AXCCAMASKn

Example:

Window for cached access from address 2000 0000_H to 3FFF $FFFF_H$ (512 MByte window size):

- AXCCABASE0 = 2000 0000_H
- AXCCAMASK0 = 3FFF $FFFF_H$

Window for buffered access from address 4000 0000_H to 400F $FFFF_H$ (1 MByte window size):

- AXCBFBASE0 = 4000 0000_H
- AXCBFMASK0 = 000F $FFFF_H$

3.3.3 Cache initialization

After Power-On reset the cache needs to be reset as follows:

1. Cache RAM for data and tag initialization

- clear data and tag RAM by issuing cache commands for all index of the cache

2. Initialization sequence for error control registers AXCSYSERR, AXCERRCNT, AXCERRADR

- 1 st step (lock clear)
  - Write AXCSYSERR.SE = 0
  - Write AXCERRCNT.ERR = 0
- 2 nd step (initialize)
  - Write all zero to AXCSYSERR
  - Write all zero to AXCERRCNT
  - Write all zero to AXCERRADR

3.3.4 XC Cache latency

XC Cache latency is different in the hit case and miss case.

When a cache miss occure, XC cache fills the data from XC slaves to own cache. Therefore, the latency in the miss case depends on the latency to XC bus and slaves.

Cache latency

- Cache hit : 8 CPU clocks
- Cache miss : 25 CPU clocks (This is general value including XC bus latency.)

3.3.5 XC Cache control registers (D1L2(H), D1M1(H), D1M2(H), D1M1A, D1M1-V2)

CAUTION

Write access to the XC Cache registers is only available in supervisor mode.

In user mode the registers can only be read.

Table 3.25 List of XC Cache control registers

Register name	Symbol	Address
Cache configuration register		
Configuration register	AXCCFG	FFFE EE00$_H$
Cache error handling registers		
System error register	AXCSYSERR	FFFE EE0C$_H$
Error control register	AXCERRCNT	FFFE EE10$_H$
Error address register	AXCERRADR	FFFE EE14$_H$
Direct handling of cache tag and data content registers		
Cache synchronize register	AXCREQSYNC	FFFE EE18$_H$
Buffer flush register	AXCBUFFLUSH	FFFE EE1C$_H$
Cache command register	AXCCACHECMD	FFFE EE20$_H$
TAGLO register	AXCTAGLO	FFFE EE24$_H$
LRU register	AXCLRU	FFFE EE28$_H$
DATALO register	AXCDATALO	FFFE EE2C$_H$
DATAHI register	AXCDATAHI	FFFE EE30$_H$
TAGECC register	AXCTAGECC	FFFE EE34$_H$
DATAECC register	AXCDATAECC	FFFE EE38$_H$
Direct window handling registers		
Cache base 0 register	AXCCABASE0	FFFE EE40$_H$
Cache mask 0 register	AXCCAMASK0	FFFE EE44$_H$
Cache base 1 register	AXCCABASE1	FFFE EE48$_H$
Cache mask 1 register	AXCCAMASK1	FFFE EE4C$_H$
Buffer base 0 register	AXCBFBASE0	FFFE EE60$_H$
Buffer mask 0 register	AXCBFMASK0	FFFE EE64$_H$
Buffer base 1 register	AXCBFBASE1	FFFE EE68$_H$
Buffer mask 1 register	AXCBFMASK1	FFFE EE6C$_H$

NOTE

XC cache registers are reserved on D1L1.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> is defined as “AXC” for the registers in the table above.

<Symbol> is defined in the above table

3.3.5.1 AXCCFG — Configuration register

Access: This register can be read/written in 32-bit units.

Address: FFFE EE00$_H$

Initial value: 0002 030F$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DW	AUT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	BF1	BF0	0	0	0	0	CA1	CA0	0	1
Initial value	0	0	0	0	0	0	1	1	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R/W	R/W	R	R

Table 3.26 AXCCFG register contents

Bit Position	Bit Name	Function
31 to 18	Reserved	When written, write the initial value.
17	DW	Direct area WAW (write after write) Wait 0: no wait 1: wait
16	AUT	SEC (ECC single bit error correction) auto re-write enable 0: disable 1: enable
15 to 10	Reserved	When written, write the initial value.
9	BF1	Buffer area 1 enable 0: disable 1: enable
8	BF0	Buffer area 0 enable 0: disable 1: enable
7 to 2	Reserved	When written, write the initial value.
3	CA1	Cache area 1 enable 0: disable 1: enable
2	CA0	Cache area 0 enable 0: disable 1: enable
1 to 0	Reserved	When written, write the initial value.

3.3.5.2 AXCSYSERR — System error register

Access: This register can be read/written in 32-bit units.

Address: FFFE EE0C$_H$

Initial value: After power-up reset PURES: 0000 0000 0000 0000 0XXX XXXX 0000 000X$_B$
Any other reset does not change the value.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	ETYPE[14:8]							0	0	0	0	0	0	0	SE
Initial value	0	X	X	X	X	X	X	X	0	0	0	0	0	0	0	X
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 3.27 AXCSYSERR register contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When written, write the initial value.
14 to 8	ETYPE[14:8]	Error type (write mask by SE = 1) ETYPE14: cross-connect error response ETYPE13: direct control error ETYPE12: cache multi hit error ETYPE11: read buffer control error ETYPE10: read buffer ID error ETYPE9: write buffer ID error ETYPE8: write buffer post error
7 to 1	Reserved	When written, write the initial value.
0	SE	System error 0: no error 1: error

3.3.5.3 AXCERRCNT — Error control register

Access: This register can be read/written in 32-bit units.

Address: FFFE EE10$_H$

Initial value: After power-up reset PURES: 0000 0000 0000 0000 0000 XXXX 000X XXXX$_B$
Any other reset does not change the value.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	DV	TV	WAY[1:0]		0	0	0	DD	DS	TD	TS	ERR
Initial value	0	0	0	0	X	X	X	X	0	0	0	X	X	X	X	X
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 3.28 AXCERRCNT register contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When written, write the initial value.
11	DV	DRAM SEC log overflow. Overwrite masked by AXCERRCNT.ERR=1
10	TV	TRAM SEC log overflow. Overwrite masked by AXCERRCNT.ERR=1
9 to 8	WAY[1:0]	Error Way. Overwrite masked by AXCERRCNT.ERR=1
7 to 5	Reserved	When written, write the initial value.
4	DD	DRAM Uncorrectable Error. Overwrite masked by AXCERRCNT.ERR=1
3	DS	DRAM Correctable Error. Overwrite masked by AXCERRCNT.ERR=1
2	TD	TRAM Uncorrectable Error. Overwrite masked by AXCERRCNT.ERR=1
1	TS	TRAM Correctable Error. Overwrite masked by AXCERRCNT.ERR=1
0	ERR	Error Status 0: no error detect 1: error detect

An error reported in AXCERRCNT will assert INTECCACC connected to the Error Control Module (ECM), see Section 48, Error Control Module (ECM).

3.3.5.4 AXCERRADR — Error address register

Access: This register can be read/written in 32-bit units.

Address: FFFE EE14$_H$

Initial value: After power-up reset PURES: XXXX XXXX$_H$.
Any other reset does not change the value.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR[31:16]															
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR[15:0]															
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.29 AXCERRADR register contents

Bit Position	Bit Name	Function
31 to 0	ADR[31:0]	XC1 slave device error address. Overwrite masked by AXCERRCNT.ERR=1 **Note:** The AXCERRADR.ADR[31:0] register stores the address that the CPU side was accessing to through the XC Cache when an access error occur. Means the address stored is the address of the slave device in the XC1 Cross-connect area (SFMA, DDR2, SDRAM, VRAM0/1).

3.3.5.5 AXCREQSYNC — Cache synchronize register

Access: This register can be read in 32-bit units.

Address: FFFE EE18$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.30 AXCREQSYNC register contents

Bit Position	Bit Name	Function
31 to 0	–	Read access to this register will synchronize/write back of buffer of the XBUS cache bus interface.*1

Note 1. The cache RAM content is not written back - please use a suitable cache command for this purpose.

3.3.5.6 AXCBUFFLUSH — Buffer flush register

Access: This register can be read in 32-bit units.

Address: FFFE EE1C$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.31 AXCBUFFLUSH register contents

Bit Position	Bit Name	Function
31 to 0	–	Buffered domain all flush. Read buffer entries are all cleared. Write buffer entries are all written to XC cross-connect and invalidated.

Note 1. The cache RAM content is not written back - please use a suitable cache command for this purpose.

3.3.5.7 AXCCACHECMD — Cache command register

Access: This register can be read/written in 32-bit units.

Address: FFFE EE20$_H$

Initial value: 0000 0000$_H$

(1) Index type (IDX_*, CLEAR) format

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	BLK[1:0]		WAY[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	IDX[7:0]								TYP[3:0]				VLD
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.32 AXCCACHECMD register contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When written, write the initial value.
19 to 18	BLK[1:0]	DRAM block. Valid only IDX_ST_DATA. 00=DW0, 01=DW1, 10=DW2, 11=DW3.
17 to 16	WAY[1:0]	Cache way. 00=Way0, 01=Way1, 10=Way2, 11=Way3.
15 to 13	Reserved	When written, write the initial value.
12 to 5	IDX[7:0]	Cache logical index. Hardware automatically compensates corkscrew allocation.
4 to 1	TYP[3:0]	Command type.
0	VLD	Cache command valid. Automatically cleared to 0 when command done. In case of writing with VLD=0, operates as APB write. So cache command doesn't activate.

(2) Hit Type (HIT_WB_INV, FILL, CREATE_DIRTY) format

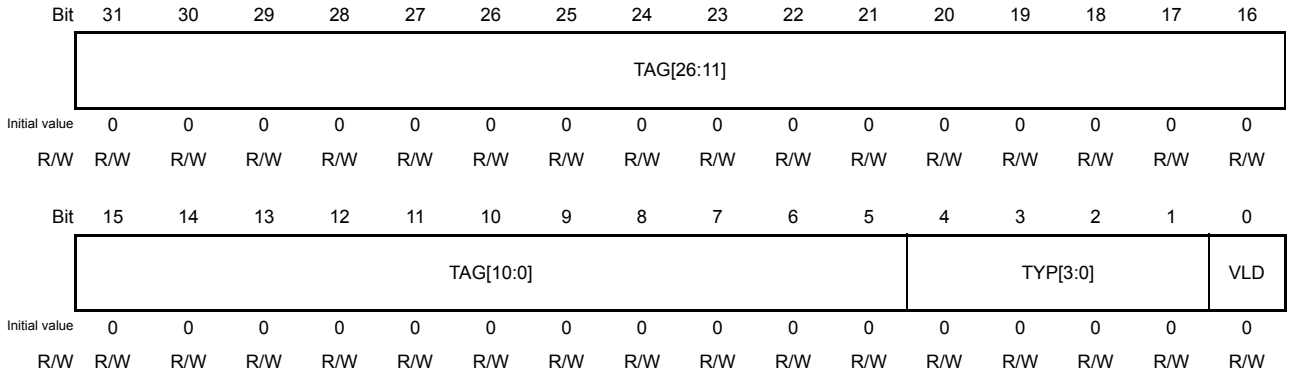


Table 3.33 AXCCACHECMD register contents

Bit Position	Bit Name	Function
31 to 5	TAG[26:0]	Physical cache tag. It is limited in cached domain. Access to other domain arises slave error.
4 to 1	TYP[3:0]	Command type.
0	VLD	Cache command valid. Automatically cleared to 0 when command done. In case of writing with VLD=0, operates as APB write. So cache command doesn't activate.

Table 3.34 Code assignment and permission of execution (1/2)

TYP[3:0]	Command	ECC detect	Note
0000_B	IDX_LD_TAG	ON(@TE=1) OFF(@TE=0)	Read from TRAM/Status-FF/LRU-FF of specified index/way. Write to AXCTAGLO, AXCLRU registers. Case of TE=0: write tag under ECC=Off. AXCTAGECC holds previous value. Case of TE=1 & TC=0: write tag/checkbits before correction. Case of TE=1 & TC=1: write tag/checkbits after correction.
0001_B	IDX_ST_TAG	No read	Read AXCTAGLO, AXCLRU registers. Case of AXCTAGECC.TE=0: Tag check bits are generated automatically. Case of AXCTAGECC.TE=1: AXCTAGECC.TAGECC[6:0] is used. Write to TRAM/Status-FF/LRU-FF of specified index/way.
0010_B	IDX_LD_DATA	ON(@TE=1) OFF(@TE=0)	Read from DRAM of specified index/double word. Write to AXCDATALO/AXCDATAHI/AXCECC registers. Case of DE=0: write data under ECC=Off. AXCDATAECC holds previous value. Case of DE=1 & DC=0 : write data/checkbits before correction. Case of DE=1 & DC=1 : write data/checkbits after correction. Hardware automatically compensates corkscrew allocation.
0011_B	IDX_ST_DATA	No read	Read from AXCDATALO/AXCDATAHI/AXCECC registers. Case of AXCDATAECC.DE=0 : Data check bits are generated automatically. Case of AXCDATAECC.DE=1 : AXCDATAECC.DATAECC{0/1} are used. Write to DRAM of specified index/doubleword. Hardware automatically compensates corkscrew allocation.
0100_B	IDX_WB_INV	ON	Check status of specified index/way. Case of Valid=1 & Dirty=1 : Writeback & Invalidate specified entry.
0101_B	FILL	ON	Fill data to cache way specified by LRU.

Table 3.34 Code assignment and permission of execution (2/2)

TYP[3:0]	Command	ECC detect	Note
0110_B	HIT_WB_INV	ON	Check status of specified address. Case of Hit & Dirty=1 : Writeback & Invalidate specified entry. Case of Hit & Dirty=0 : Invalidate specified entry Case of Miss : No operation.
0111_B	CREATE_DIRTY	OFF	Create dirty status. Case of Hit : Set D=1 Case of Miss : Write specified address to TRAM. Case of V=0 : Write 0 to DRAM Case of V=1 & D=1 : with writeback.
$1xxx_B$	CLEAR	No read	Clear target : All of {Valid, Dirty, LRU} bits.

3.3.5.8 AXCTAGLO — TAGLO register

Access: This register can be read/written in 32-bit units.

Address: FFFE $EE24_H$

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAG[18:3]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAG[2:0]			0	0	0	0	0	0	0	0	0	0	0	D	V
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 3.35 AXCTAGLO register contents

Bit Position	Bit Name	Function
31 to 13	TAG[18:0]	TAG
12 to 2	Reserved	When written, write the initial value.
1	D	Dirty
0	V	Valid

3.3.5.9 AXCLRU — LRU register

Access: This register can be read/written in 32-bit units.

Address: FFFE EE28$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LRU
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 3.36 AXCLRU register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	LRU	LRU

3.3.5.10 AXCDATALO — DATALO register

Access: This register can be read/written in 32-bit units.

Address: FFFE EE2C$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATALO[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATALO[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.37 AXCDATALO register contents

Bit Position	Bit Name	Function
31 to 0	DATALO[31:0]	Write data[31:0] for IDX_ST_DATA.

3.3.5.11 AXCDATAHI — DATAHI register

Access: This register can be read/written in 32-bit units.

Address: FFFE EE30$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATAHI[63:48]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATAHI[47:32]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 3.38 AXCDATAHI register contents

Bit Position	Bit Name	Function
31 to 0	DATAHI[63:32]	Write data[63:32] for IDX_ST_DATA.

3.3.5.12 AXCTAGECC — TAGECC register

Access: This register can be read/written in 32-bit units.

Address: FFFE EE34$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	TAGECC[6:0]							0	0	0	0	0	0	TC	TE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 3.39 AXCTAGECC register contents

Bit Position	Bit Name	Function
31 to 15	Reserved	When written, write the initial value.
14 to 8	TAGECC[6:0]	Tag ECC for IDX_ST_TAG/IDX_LD_TAG. TE=1 : Use TAGECC field TE=0 : Use generated syndrome
7 to 2	Reserved	When written, write the initial value.
1	TC	Tag Read Capture 0: before correction 1: after correction
0	TE	Error Syndrome Enable.

3.3.5.13 AXCDATAECC — DATAECC register

Access: This register can be read/written in 32-bit units.

Address: FFFE EE38$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	DATAECC1[6:0]						
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	DATAECC0[6:0]							0	0	0	0	0	0	DC	DE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Note 1. The initial value “0” of bit 16 must be changed to “1” and must not be changed afterwards.

Table 3.40 AXCDATAECC register contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When written, write the initial value.
22 to 16	DATAECC1[6:0]	Higher word Data ECC for IDX_ST_DATA/IDX_LD_DATA. DE=1 : Use DATAECC field, DE=0 : Use generated syndrome
15	Reserved	When written, write the initial value.
14 to 8	DATAECC0[6:0]	Lower word Data ECC for IDX_ST_DATA/IDX_LD_DATA. DE=1 : Use DATAECC field, DE=0 : Use generated syndrome
1	DC	Data Read Capture 0: before correction 1: after correction.
0	DE	Error Syndrome Enable

3.3.5.14 AXCCABASEn — Cache base n register (n = 0, 1)

Access: This register can be read/written in 32-bit units.

Address: AXCCABASE0: FFFE EE40$_H$
AXCCABASE1: FFFE EE48$_H$

Initial value: AXCCABASE0: E400 0000$_H$
AXCCABASE1: 2200 0000$_H$

AXCCABASE0:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BASE[11:0]												0	0	0	0
Initial value	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

AXCCABASE1:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BASE[11:0]												0	0	0	0
Initial value	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.41 AXCCABASEn register contents

Bit Position	Bit Name	Function
31 to 20	BASE[11:0]	Cache Base Address
19 to 0	Reserved	When written, write the initial value.

3.3.5.15 AXCCAMASKn — Cache mask n register (n = 0, 1)

Access: This register can be read/written in 32-bit units.

Address: AXCCAMASK0: FFFE EE44$_H$
AXCCAMASK1: FFFE EE4C$_H$

Initial value: AXCCAMASK0: 03FF FFFF$_H$
AXCCAMASK1: 01FF FFFF$_H$

AXCCAMASK0:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MASK[11:0]												1	1	1	1
Initial value	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

AXCCAMASK1:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MASK[11:0]												1	1	1	1
Inivtial value	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.42 AXCCAMASKn register contents

Bit Position	Bit Name	Function
31 to 20	MASK[11:0]	Cache Address Mask. 0=No Mask, 1=Mask. Minimum size (Mask[31:20]=12'h000) is 1M-bytes. Effective in the range which specified 1 continuously from the lower bit. 1 after 0 appeared ignores.
19 to 0	Reserved	When written, write the initial value.

3.3.5.16 AXCBFBASEn — Buffer base n register (n = 0, 1)

Access: This register can be read/written in 32-bit units.

Address: AXCBFBASE0: FFFE EE60$_H$
AXCBFBASE1: FFFE EE68$_H$

Initial value: AXCBFBASE0: E800 0000$_H$
AXCBFBASE1: 2600 0000$_H$

AXCBFBASE0:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BASE[11:0]												0	0	0	0
Initial value	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

AXCBFBASE1:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BASE[11:0]												0	0	0	0
Initial value	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.43 AXCBFBASEn register contents

Bit Position	Bit Name	Function
31 to 20	BASE[11:0]	Buffer Base Address
19 to 0	Reserved	When written, write the initial value.

3.3.5.17 AXCBFMASKn — Buffer mask n register (n = 0, 1)

Access: This register can be read/written in 32-bit units.

Address: AXCBFMASK0: FFFE EE64$_H$
AXCBFMASK1: FFFE EE6C$_H$

Initial value: AXCBFMASK0: 03FF FFFF$_H$
AXCBFMASK1: 01FF FFFF$_H$

AXCBFMASK0:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MASK[11:0]												1	1	1	1
Initial value	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

AXCBFMASK1:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MASK[11:0]												1	1	1	1
Initial value	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.44 AXCBFMASKn register contents

Bit Position	Bit Name	Function
31 to 20	MASK[11:0]	Buffer Address Mask. 0=No Mask, 1=Mask. Minimum size (Mask[31:20]=12'h000) is 1M-bytes. Effective in the range which specified 1 continuously from the lower bit. 1 after 0 appeared ignores.
19 to 0	Reserved	When written, write the initial value.

3.3.6 XC Cache ECC control registers (D1L2(H), D1M1(H), D1M2(H), D1M1A, D1M1-V2)

Table 3.45 List of XC Cache ECC control registers

Register name	Symbol	Address
XC cache data/tag RAM ECC Control Register	AXCDECCCTL_PE1_OS	FFC6 6400$_H$
XC cache data RAM ECC Error Information Register	AXCDERRINT_PE1_OS	FFC6 6404$_H$
XC cache data RAM Error Status Clear Register	AXCDSTCLR_PE1_OS	FFC6 6408$_H$
XC cache data RAM Error Overflow Register	AXCDOVFSTR_PE1_OS	FFC6 640C$_H$
XC cache data RAM 1st Error Status Register 0	AXCD1STERSTR0_PE1_OS	FFC6 6410$_H$
XC cache data RAM 1st Error Status Register 1	AXCD1STERSTR1_PE1_OS	FFC6 6414$_H$
XC cache data RAM 1st Error Status Register 2	AXCD1STERSTR2_PE1_OS	FFC6 6418$_H$
XC cache data RAM 1st Error Status Register 3	AXCD1STERSTR3_PE1_OS	FFC6 641C$_H$
XC cache data RAM (Bank 0) 1st Error Address Register 0	AXCD1STEADR0_PE1_OS	FFC6 6450$_H$
XC cache data RAM (Bank 1) 1st Error Address Register 1	AXCD1STEADR1_PE1_OS	FFC6 6454$_H$
XC cache data RAM (Bank 2) 1st Error Address Register 2	AXCD1STEADR2_PE1_OS	FFC6 6458$_H$
XC cache data RAM (Bank 3) 1st Error Address Register 3	AXCD1STEADR3_PE1_OS	FFC6 645C$_H$
XC cache data RAM (Bank 4) 1st Error Address Register 4	AXCD1STEADR4_PE1_OS	FFC6 6460$_H$
XC cache data RAM (Bank 5) 1st Error Address Register 5	AXCD1STEADR5_PE1_OS	FFC6 6464$_H$
XC cache data RAM (Bank 6) 1st Error Address Register 6	AXCD1STEADR6_PE1_OS	FFC6 6468$_H$
XC cache data RAM (Bank 7) 1st Error Address Register 7	AXCD1STEADR7_PE1_OS	FFC6 646C$_H$
XC cache data RAM (Bank 8) 1st Error Address Register 8	AXCD1STEADR8_PE1_OS	FFC6 6470$_H$
XC cache data RAM (Bank 9) 1st Error Address Register 9	AXCD1STEADR9_PE1_OS	FFC6 6474$_H$
XC cache data RAM (Bank 10) 1st Error Address Register 10	AXCD1STEADR10_PE1_OS	FFC6 6478$_H$
XC cache data RAM (Bank 11) 1st Error Address Register 11	AXCD1STEADR11_PE1_OS	FFC6 647C$_H$
XC cache data RAM (Bank 12) 1st Error Address Register 12	AXCD1STEADR12_PE1_OS	FFC6 6480$_H$
XC cache data RAM (Bank 13) 1st Error Address Register 13	AXCD1STEADR13_PE1_OS	FFC6 6484$_H$
XC cache data RAM (Bank 14) 1st Error Address Register 14	AXCD1STEADR14_PE1_OS	FFC6 6488$_H$
XC cache data RAM (Bank 15) 1st Error Address Register 15	AXCD1STEADR15_PE1_OS	FFC6 648C$_H$

Table 3.46 List of XC Cache Tag ECC control registers

Register name	Symbol	Address
XC cache tag RAM ECC Error Information Register	AXCTERRINT_PE1_OS	FFC6 7404$_H$
XC cache tag RAM Error Status Clear Register	AXCTSTCLR_PE1_OS	FFC6 7408$_H$
XC cache tag RAM Error Overflow Register	AXCTOVFSTR_PE1_OS	FFC6 740C$_H$
XC cache tag RAM 1st Error Status Register 0	AXCT1STERSTR_PE1_OS	FFC6 7410$_H$
XC cache tag RAM (Bank 0) 1st Error Address Register 0	AXCT1STEADR0_PE1_OS	FFC6 7450$_H$
XC cache tag RAM (Bank 1) 1st Error Address Register 1	AXCT1STEADR1_PE1_OS	FFC6 7454$_H$

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> is defined as “ECCAXC” for the registers in the table above.

<Symbol> is defined in the above table

3.3.6.1 AXCDECCCTL_PE1_OS – XC cache data/tag RAM ECC control register

AXCDECCCTL enables or disables ECC error detection and correction and 1-bit error correction for cache data RAM and tag RAM. Set PROT[1:0] bits to 01B when writing to AXCDECCCTL.

AXCDECCCTL is initialized by an internal reset or an external reset.

Access: This register can be read/written in 16-bit and 32-bit units.

Address: FFC6 6400$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT[1:0]		–	–	–	–	–	–	–	–	–	–	–	–	–	ECCDIS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 3.47 AXCDECCCTL_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, these bits are always read as 0. When writing, always write 0.
15, 14	PROT[1:0]	Enables or disables modification of the ECCDIS bits. The value written is not retained. These bits are always read as 0. Set PROT[1:0] = 01$_B$ when writing to AXCDECCCTL.
13 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	ECCDIS	ECC disable bit Setting ECC error detection to enable/disable. 0: ECC error detection is enable 1: ECC error detection is disable

3.3.6.2 AXCDERRINT_PE1_OS – XC cache data RAM error information control register

AXCDERRINT enables or disables generation of the error notification signal to the ECM upon detection of an ECC 2-bit error or an ECC 1-bit error in cache data RAM.

AXCDERRINT is initialized by an internal reset or an external reset.

Access: This register can be read/written in 8-bit, 16-bit and 32-bit units.

Address: FFC6 6404_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	DEDIE	SEDIE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 3.48 AXCDERRINT_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	DEDIE	ECC 2-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

3.3.6.3 AXCDSTCLR_PE1_OS – XC cache data RAM error status clear register

AXCDSTCLR clears the error flags in the error status register (AXCD1STERSTRm, m = 0 to 3), the overflow flag in the error overflow status register (AXCDOVFSTR), and the error address register (AXCD1STEADRn, n = 0 to 15). AXCDSTCLR is a write-only register and is always read as 0.

Access: This register can be read/written in 8-bit, 16-bit and 32-bit units.

Address: FFC6 6408$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SST CLR15	SST CLR14	SST CLR13	SST CLR12	SST CLR11	SST CLR10	SST CLR9	SST CLR8	SST CLR7	SST CLR6	SST CLR5	SST CLR4	SST CLR3	SST CLR2	SST CLR1	SST CLR0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 3.49 AXCDSTCLR_PE1_OS register contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, these bits are always read as 0. When writing, always write 0.
15	SSTCLR15	Error Overflow Flag Clear (for bank 15) Writing 1 to this bit clears the DEDF15 and SEDF15 flags in AXCD1STERSTR3 and ERROVF15 flag in AXCDOVFSTR and AXCD1STEADR15.
14	SSTCLR14	Error Overflow Flag Clear (for bank 14) Writing 1 to this bit clears the DEDF14 and SEDF14 flags in AXCD1STERSTR3 and ERROVF14 flag in AXCDOVFSTR and AXCD1STEADR14.
13	SSTCLR13	Error Overflow Flag Clear (for bank 13) Writing 1 to this bit clears the DEDF13 and SEDF13 flags in AXCD1STERSTR3 and ERROVF13 flag in AXCDOVFSTR and AXCD1STEADR13.
12	SSTCLR12	Error Overflow Flag Clear (for bank 12) Writing 1 to this bit clears the DEDF12 and SEDF12 flags in AXCD1STERSTR3 and ERROVF12 flag in AXCDOVFSTR and AXCD1STEADR12.
11	SSTCLR11	Error Overflow Flag Clear (for bank 11) Writing 1 to this bit clears the DEDF11 and SEDF11 flags in AXCD1STERSTR2 and ERROVF11 flag in AXCDOVFSTR and AXCD1STEADR11.
10	SSTCLR10	Error Overflow Flag Clear (for bank 10) Writing 1 to this bit clears the DEDF10 and SEDF10 flags in AXCD1STERSTR2 and ERROVF10 flag in AXCDOVFSTR and AXCD1STEADR10.
9	SSTCLR9	Error Overflow Flag Clear (for bank 9) Writing 1 to this bit clears the DEDF9 and SEDF9 flags in AXCD1STERSTR2 and ERROVF9 flag in AXCDOVFSTR and AXCD1STEADR9.
8	SSTCLR8	Error Overflow Flag Clear (for bank 8) Writing 1 to this bit clears the DEDF8 and SEDF8 flags in AXCD1STERSTR2 and ERROVF8 flag in AXCDOVFSTR and AXCD1STEADR8.
7	SSTCLR7	Error Overflow Flag Clear (for bank 7) Writing 1 to this bit clears the DEDF7 and SEDF7 flags in AXCD1STERSTR1 and ERROVF7 flag in AXCDOVFSTR and AXCD1STEADR7.
6	SSTCLR6	Error Overflow Flag Clear (for bank 6) Writing 1 to this bit clears the DEDF6 and SEDF6 flags in AXCD1STERSTR1 and ERROVF6 flag in AXCDOVFSTR and AXCD1STEADR6.

Table 3.49 AXCDSTCLR_PE1_OS register contents (2/2)

Bit Position	Bit Name	Function
5	SSTCLR5	Error Overflow Flag Clear (for bank 5) Writing 1 to this bit clears the DEDF5 and SEDF5 flags in AXCD1STERSTR1 and ERROVF5 flag in AXCDOVFSTR and AXCD1STEADR5.
4	SSTCLR4	Error Overflow Flag Clear (for bank 4) Writing 1 to this bit clears the DEDF4 and SEDF4 flags in AXCD1STERSTR1 and ERROVF4 flag in AXCDOVFSTR and AXCD1STEADR4.
3	SSTCLR3	Error Overflow Flag Clear (for bank 3) Writing 1 to this bit clears the DEDF3 and SEDF3 flags in AXCD1STERSTR0 and ERROVF3 flag in AXCDOVFSTR and AXCD1STEADR3.
2	SSTCLR2	Error Overflow Flag Clear (for bank 2) Writing 1 to this bit clears the DEDF2 and SEDF2 flags in AXCD1STERSTR0 and ERROVF2 flag in AXCDOVFSTR and AXCD1STEADR2.
1	SSTCLR1	Error Overflow Flag Clear (for bank 1) Writing 1 to this bit clears the DEDF1 and SEDF1 flags in AXCD1STERSTR0 and ERROVF1 flag in AXCDOVFSTR and AXCD1STEADR1.
0	SSTCLR0	Error Overflow Flag Clear (for bank 0) Writing 1 to this bit clears the DEDF0 and SEDF0 flags in AXCD1STERSTR0 and ERROVF0 flag in AXCDOVFSTR and AXCD1STEADR0.

3.3.6.4 AXCDOVFSTR_PE1_OS – XC cache data RAM error count overflow status

AXCDOVFSTR monitors occurrence of error overflow in cache data RAM. If the second error occurs after the first error (= while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set. ERROVFn is cleared by an internal reset, an external reset, or setting the STCLRn bit to 1 in AXCDSTCLR.

Access: This register can be read in 8-bit, 16-bit and 32-bit units.

Address: FFC6 640C$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ERR OVF15	ERR OVF14	ERR OVF13	ERR OVF12	ERR OVF11	ERR OVF10	ERR OVF9	ERR OVF8	ERR OVF7	ERR OVF6	ERR OVF5	ERR OVF4	ERR OVF3	ERR OVF2	ERR OVF1	ERR OVF0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.50 AXCDOVFSTR_PE1_OS register contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, these bits are always read as 0. When writing, always write 0.
15	ERROVF15	Error Overflow Flag (for bank 15) ERROVF15 is set if the second error occurs while any of the error flags (DEDF15 and SEDF15) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
14	ERROVF14	Error Overflow Flag (for bank 14) ERROVF14 is set if the second error occurs while any of the error flags (DEDF14 and SEDF14) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
13	ERROVF13	Error Overflow Flag (for bank 13) ERROVF13 is set if the second error occurs while any of the error flags (DEDF13 and SEDF13) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
12	ERROVF12	Error Overflow Flag (for bank 12) ERROVF12 is set if the second error occurs while any of the error flags (DEDF12 and SEDF12) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
11	ERROVF11	Error Overflow Flag (for bank 11) ERROVF11 is set if the second error occurs while any of the error flags (DEDF11 and SEDF11) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
10	ERROVF10	Error Overflow Flag (for bank 10) ERROVF10 is set if the second error occurs while any of the error flags (DEDF10 and SEDF10) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.

Table 3.50 AXCDOVFSTR_PE1_OS register contents (2/2)

Bit Position	Bit Name	Function
9	ERROVF9	Error Overflow Flag (for bank 9) ERROVF9 is set if the second error occurs while any of the error flags (DEDF9 and SEDF9) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
8	ERROVF8	Error Overflow Flag (for bank 8) ERROVF8 is set if the second error occurs while any of the error flags (DEDF8 and SEDF8) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
7	ERROVF7	Error Overflow Flag (for bank 7) ERROVF7 is set if the second error occurs while any of the error flags (DEDF7 and SEDF7) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
6	ERROVF6	Error Overflow Flag (for bank 6) ERROVF6 is set if the second error occurs while any of the error flags (DEDF6 and SEDF6) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
5	ERROVF5	Error Overflow Flag (for bank 5) ERROVF5 is set if the second error occurs while any of the error flags (DEDF5 and SEDF5) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
4	ERROVF4	Error Overflow Flag (for bank 4) ERROVF4 is set if the second error occurs while any of the error flags (DEDF4 and SEDF4) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
3	ERROVF3	Error Overflow Flag (for bank 3) ERROVF3 is set if the second error occurs while any of the error flags (DEDF3 and SEDF3) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
2	ERROVF2	Error Overflow Flag (for bank 2) ERROVF2 is set if the second error occurs while any of the error flags (DEDF2 and SEDF2) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
1	ERROVF1	Error Overflow Flag (for bank 1) ERROVF1 is set if the second error occurs while any of the error flags (DEDF1 and SEDF1) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
0	ERROVF0	Error Overflow Flag (for bank 0) ERROVF0 is set if the second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.

3.3.6.5 AXCD1STERSTRn_PE1_OS – XC cache data RAM 1st error status register (n = 0 - 3)

AXCD1STERSTRn monitors occurrence of the first error in cache data RAM. The error status is set if an error occurs while all the error flags for the relevant banks are 0. The error flag is overwritten if an ECC 2-bit error occurs while the ECC 1-bit error monitor flag is set.

SEDFn and DEDFn is cleared by an internal reset, an external reset, or setting the STCLRn bit to 1 in AXCDSTCLR.

Access: This register can be read in 8-bit, 16-bit and 32-bit units.

Address: FFC6 6410$_H$ + 4 × n

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	DEDF 4n+3	SEDF 4n+3	–	–	–	–	–	–	DEDF 4n+2	SEDF 4n+2
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	DEDF 4n+1	SEDF 4n+1	–	–	–	–	–	–	DEDF 4n	SEDF 4n
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.51 AXCD1STERSTRn_PE1_OS register contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	When read, these bits are always read as 0. When writing, always write 0.
25	DEDF4n+3	ECC 2-bit error Monitor Flag (for bank 4n+3) 0: Cleared to 0 by setting the STCLR4n+3 bit to 1 in AXCDSTCLR. 1: Indicates that an ECC 2-bit error has occurred while the error flag DEDF4n+3 is 0.
24	SEDF4n+3	ECC 1-bit error Monitor Flag (for bank 4n+3) 0: Cleared to 0 by setting the STCLR4n+3 bit to 1 in AXCDSTCLR. 1: Indicates that an ECC 1-bit error has occurred while the error flags DEDF4n+3 and SEDF4n+3 are 0.
23 to 18	Reserved	When read, these bits are always read as 0. When writing, always write 0.
17	DEDF4n+2	ECC 2-bit error Monitor Flag (for bank 4n+2) 0: Cleared to 0 by setting the STCLR4n+2 bit to 1 in AXCDSTCLR. 1: Indicates that an ECC 2-bit error has occurred while the error flag DEDF4n+2 is 0.
16	SEDF4n+2	ECC 1-bit error Monitor Flag (for bank 4n+2) 0: Cleared to 0 by setting the STCLR4n+2 bit to 1 in AXCDSTCLR. 1: Indicates that an ECC 1-bit error has occurred while the error flags DEDF4n+2 and SEDF4n+2 are 0.
16 to 10	Reserved	When read, these bits are always read as 0. When writing, always write 0.
9	DEDF4n+1	ECC 2-bit error Monitor Flag (for bank 4n+1) 0: Cleared to 0 by setting the STCLR4n+1 bit to 1 in AXCDSTCLR. 1: Indicates that an ECC 2-bit error has occurred while the error flag DEDF4n+1 is 0.
8	SEDF4n+1	ECC 1-bit error Monitor Flag (for bank 4n+1) 0: Cleared to 0 by setting the STCLR4n+1 bit to 1 in AXCDSTCLR. 1: Indicates that an ECC 1-bit error has occurred while the error flags DEDF4n+1 and SEDF4n+1 are 0.
7 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.

Table 3.51 AXCD1STERSTRn_PE1_OS register contents (2/2)

Bit Position	Bit Name	Function
1	DEDF4n	ECC 2-bit error Monitor Flag (for bank 4n) 0: Cleared to 0 by setting the STCLR4n bit to 1 in AXCDSTCLR. 1: Indicates that an ECC 2-bit error has occurred while the error flag DEDF4n is 0.
0	SEDF4n	ECC 1-bit error Monitor Flag (for bank 4n) 0: Cleared to 0 by setting the STCLR4n bit to 1 in AXCDSTCLR. 1: Indicates that an ECC 1-bit error has occurred while the error flags DEDF4n and SEDF4n are 0.

3.3.6.6 AXCD1STEADRn_PE1_OS – XC cache data RAM (bank n) 1st error address register (n = 0 - 15)

AXCD1STEADR holds the address at which an error has occurred in cache data RAM. The error address is set if an error occurs while all the error flags for the relevant banks are 0 in AXCD1STERSTRm (m = 0 to 3). The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.

Since this register holds the internal address, add the base address of the associated memory to transform the internal address to the real address. AXCD1STEADRn is cleared by an internal reset, an external reset, or setting the STCLRn bit to 1 in AXCDSTCLR.

Access: This register can be read in 8-bit, 16-bit and 32-bit units.

Address: FFC6 6450_H + 4 × n

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EADRn[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADRn[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.52 AXCD1STEADRn_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 0	EADRn[31:0]	1st Error Address (for bank n) Monitors the address of the first error. The error address is set if an error occurs while all the error flags for bank n are 0 in AXCD1STERSTRm. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.

3.3.6.7 AXCTERRINT_PE1_OS – XC cache tag RAM error information control register

AXCTERRINT enables or disables generation of the error notification signal to the ECM upon detection of an ECC 2-bit error or an ECC 1-bit error in cache tag RAM.

AXCTERRINT is initialized by an internal reset or an external reset.

Access: This register can be read/written in 8-bit, 16-bit and 32-bit units.

Address: FFC6 7404$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	DEDIE	SEDIE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 3.53 AXCTERRINT_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	DEDIE	ECC 2-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

3.3.6.8 AXCTSTCLR_PE1_OS – XC cache tag RAM error status clear register

AXCTSTCLR clears the error flags in the error status register (AXCT1STERSTR), the overflow flag in the error overflow status register (AXCTOVFSTR), and the error address register (AXCT1STEADRn, n = 0, 1). AXCTSTCLR is a write-only register and is always read as 0.

Access: This register can be read/written in 8-bit, 16-bit and 32-bit units.

Address: FFC6 7408$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	ST CLR1	ST CLR0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 3.54 AXCTSTCLR_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	STCLR1	Error Overflow Flag Clear (for bank 1) Writing 1 to this bit clears the DEDF1 and SEDF1 flags in AXCT1STERSTR and ERROVF1 flag in AXCTOVFSTR and AXCT1STEADR1.
0	STCLR0	Error Overflow Flag Clear (for bank 0) Writing 1 to this bit clears the DEDF0 and SEDF0 flags in AXCT1STERSTR and ERROVF0 flag in AXCTOVFSTR and AXCT1STEADR0.

3.3.6.9 AXCTOVFSTR_PE1_OS – XC cache tag RAM error count overflow status register

AXCTOVFSTR monitors occurrence of error overflow in cache tag RAM. If the second error occurs after the first error (= while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set. ERROVFn is cleared by an internal reset, an external reset, or setting the STCLRn bit to 1 in AXCTSTCLR.

Access: This register can be read in 8-bit, 16-bit and 32-bit units.

Address: FFC6 740C$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	ERR OVF1	ERR OVF0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.55 AXCTOVFSTR_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	ERROVF1	Error Overflow Flag (for bank 1) ERROVF1 is set if the second error occurs while any of the error flags (DEDF1 and SEDF1) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.
0	ERROVF0	Error Overflow Flag (for bank 0) ERROVF0 is set if the second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register is set,except when both of the error address and source of the second error are the same as those of the first error.

3.3.6.10 AXCT1STERSTR_PE1_OS – XC cache tag RAM 1st error status register

AXCT1STERSTR monitors occurrence of the first error in cache tag RAM.

The error flag is overwritten if an ECC 2-bit error occurs while the error flag is 0.

AXCT1STERSTR is cleared by an internal reset, an external reset, or setting the STCLRn bit to 1 in AXCTSTCLR.

Access: This register can be read in 8-bit, 16-bit and 32-bit units.

Address: FFC6 7410$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	DEDF1	SEDF1	–	–	–	–	–	–	DEDF0	SEDF0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.56 AXCT1STERSTR_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When read, these bits are always read as 0. When writing, always write 0.
9	DEDF1	ECC 2-bit error Monitor Flag (for bank 1) 0: Cleared to 0 by setting the STCLR1 bit to 1 in AXCTSTCLR. 1: Indicates that an ECC 2-bit error has occurred while the error flag DEDF1 is 0.
8	SEDF1	ECC 1-bit error Monitor Flag (for bank 1) 0: Cleared to 0 by setting the STCLR1 bit to 1 in AXCTSTCLR. 1: Indicates that an ECC 1-bit error has occurred while the error flags DEDF1 and SEDF1 are 0.
7 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	DEDF0	ECC 2-bit error Monitor Flag (for bank 0) 0: Cleared to 0 by setting the STCLR0 bit to 1 in AXCTSTCLR. 1: Indicates that an ECC 2-bit error has occurred while the error flag DEDF0 is 0.
0	SEDF0	ECC 1-bit error Monitor Flag (for bank 0) 0: Cleared to 0 by setting the STCLR0 bit to 1 in AXCTSTCLR. 1: Indicates that an ECC 1-bit error has occurred while the error flags DEDF0 and SEDF0 are 0.

3.3.6.11 AXCT1STEADRn_PE1_OS – XC cache tag RAM 1st error address register (n = 0, 1)

AXCT1STEADRn holds the address at which an error has occurred in cache tag RAM. The error address is set if an error occurs while all the error flags for the relevant banks are 0 in AXCT1STERSTR. The address is updated if an ECC 1-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated. Since this register holds the internal address, add the base address of the associated memory to transform the internal address to the real address.

AXCT1STEADRn is cleared by an internal reset, an external reset, or setting the STCLRn bit to 1 in AXCTSTCLR.

Access: This register can be read in 8-bit, 16-bit and 32-bit units.

Address: FFC6 7450_H + 4 × n

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EADRn[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADRn[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 3.57 AXCD1STEADRn_PE1_OS register contents

Bit Position	Bit Name	Function
31 to 0	EADRn[31:0]	1st Error Address (for bank n) Monitors the address of the first error. The error address is set if an error occurs while all the error flags for bank n are 0 in AXCT1STESTR. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.

3.4 Usage Notes

3.4.1 Synchronization of Store Instruction Completion and Subsequent Instruction Generation

When a control register is updated by a store instruction, there is a time lag between CPU implementation of the store instruction and actual updating of the control register. Therefore, adequate synchronization processing is needed to ensure the control register reflects updated contents before generation of a subsequent instruction. How to perform synchronization processing is shown below.

Regarding the procedure of system register update by LDSR instruction and synchronizing subsequent instruction generation, see

APPENDIX A HAZARD RESOLUTION PROCEDURE FOR SYSTEM REGISTERS
in RH850G3M User's Manual: Software.

When updated results in the control registers are reflected in the implementation of a subsequent instruction:

Example1: an interrupt is enabled by implementation of an EI instruction after an interrupt request is cleared by access from the control register in the INTC2 and the peripheral circuits. Proceed as follows in this case.

(1) Store instruction to update a control register (ST.W, etc.)

(2) Dummy read of the above-mentioned control register (LD.W etc.)

(3) SYNCP

(4) Subsequent instruction (EI)

Example 2: Implement the same processing even when the access required after waiting to secure the updating of a given control register (register A) is to another control register (register B). This includes the following cases: the interlinked operation of different peripheral modules and when releasing the interrupt mask in INTC after making peripheral module settings.

However, this processing is unnecessary if control registers A and B are in the same PBUS group.

(1) Issue the instruction for storage to update control register A (ST.W, etc.)

(2) Dummy-read the above control register (LD.W, etc.)

(3) Issue SYNCP.

(4) Issue the instruction to access control register B (ST.W, LD.W, etc.)

The same processing is also required when access to control registers and memory within the scope of protection starts after waiting for the completion of settings for safety functions such as memory protection, ECC checking, and so on.

When the updated results in the control registers and memories are reflected in the instruction fetch of a subsequent instruction:

(Case a) After writing an instruction in a RAM, CPU fetch the written instruction from the same RAM. Proceed as follows.

(1) Store instruction to update a memory (ST.W, etc.)

(2) Dummy read of the above-mentioned memory (LD.W, etc.)

(3) SYNCP

(4) SYNCI

(5) Subsequent instruction (branch instruction, etc.)

(Case b) After completion of MPU function or ECC register updating, CPU fetch the instruction from the same RAM.
Proceed as follows.

(1) Store instruction to update a memory (ST.W, etc.)

(2) Dummy read of the above-mentioned memory (LD.W, etc.)

(3) SYNCP

(4) SYNCI

(5) Subsequent instruction (branch instruction, etc.)

When switching Code Flash area:

In this case, see (7) Update of FCUFAREA register in the RH850/D1x Flash Memory User's Manual: Hardware Interface

3.4.2 Accesses to Registers by Bit-Manipulation Instructions

Processing of a bit-manipulation instruction takes the form of atomic reading, modification, and writing of an eight-bit unit. Therefore, registers which enable reading and writing in 8 bits can be basically accessed by bit-manipulation instructions. However, take care in the cases of registers that contain multiple flag bits, since the read-modify-write cycle may also clear flags other than that which was the target for clearing.

NOTE

Bit-manipulation instructions cannot be used for VDCE, DISCOM, JCUA, SFMA, SDRA, VOWE.

3.4.3 Ensuring Coherency after Code Flash Programming

The CPU has an efficient instruction cache and data buffer for the code flash area.

Therefore, after using self-programming to program the code flash memory, clear the instruction cache and data buffer to ensure coherency. The instruction cache and data buffer can be cleared by using the ICCTRL register and the CDBCR register, respectively.

3.4.4 Overwriting Context when Acknowledging Multiple Exceptions

Exceptions may be acknowledged regardless of the states of the ID and NP bits of the PSW register. This depends on the type of exception. When multiple exceptions occur, the contents of the system registers which hold the context information are overwritten. Regarding the conditions for acknowledging exceptions from each source and the possibility of return and recovery, see the list of exception sources in the *RH850G3M User's Manual:Software*.

3.4.5 Usage Notes on Prefetching

CPU executes speculative instruction fetching from locations later than the current value of the program counter to retain maintain the throughput of instruction fetches. Reading from memory due to such prefetching may proceed even from locations to which instruction codes have not been assigned (note 1 in **Figure 3.2**). Please note the following. The CPU does not execute values read in such cases.

These notes apply to instruction fetching from memory in general.

- Occurrence of ECC errors due to values in memory being undefined
  This prefetching may lead to an ECC error in case of reading from the code flash memory after it has been erased or from the local RAM before initialization. When instruction codes are assigned to memory, initialize said area with values as desired (note 1 in **Figure 3.2**).
- Detection of illegal access by the IPG
  The IPG may detect such prefetching as illegal access. To prevent prefetching being detected as an illegal access, do not allow any region of overlap area with said areas (note 1 in **Figure 3.2**) and areas to which access is prohibited by the IPG. Reading from an area protected by the MPU does not cause a memory protection exception.
- Access to Access Prohibited Area
  Assign instruction codes to memory without allowing any overlap between said area (note 1 in **Figure 3.2**) and an access-prohibited area.

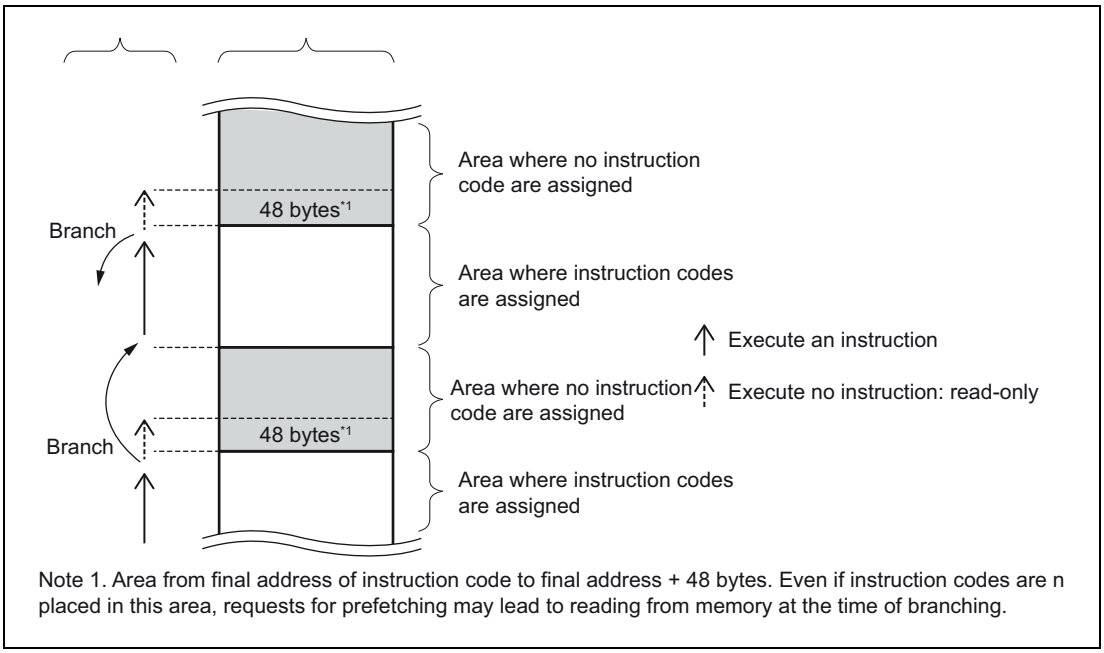


Figure 3.2 Area that Requires Attention Regarding Prefetching

3.4.6 Initialization of Register Set

For a register in the register set, which contains an undefined value after reset, write a value and then read it.

Section 4 Write-Protected Registers

Write-protected registers are protected from inadvertent write access due to erroneous program execution, etc.

Writing to a write-protected register requires a special register protection unlock sequence.

4.1 Register protection clusters

Protected registers are bundled in some register protection clusters.

The protection equipment treats all registers in the same cluster as a single protection unit.

If the protection unlock sequence for a register is initiated, no access to any other register of the same protection cluster is allowed. Otherwise the unlock sequence is disrupted and the register write fails. The diagram below shows a disruption of the unlock sequence by an access to the same cluster within an interrupt service routine.

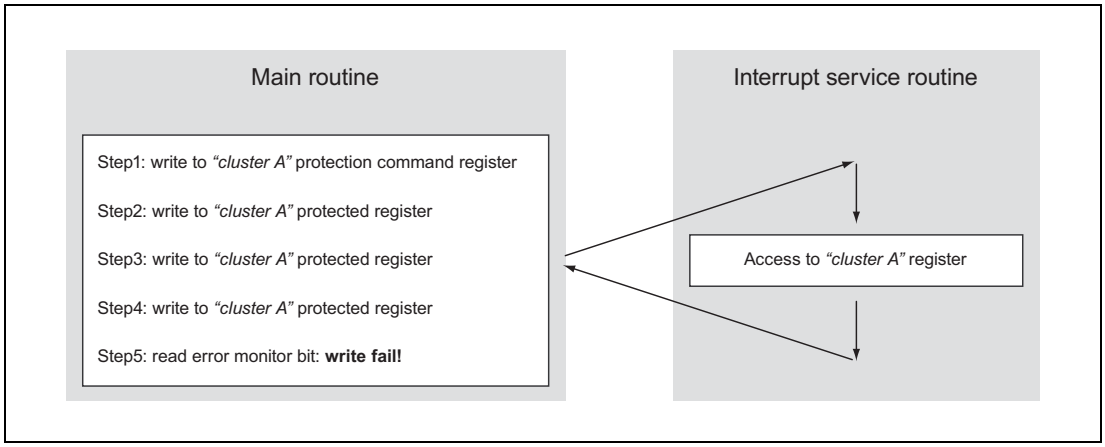


Figure 4.1 Interruption of Register Protection Unlock Sequence

Access to a register of another protection cluster during the unlock sequence does not disrupt the unlock sequence and the register write can be completed successfully.
The following diagram below shows such situation.

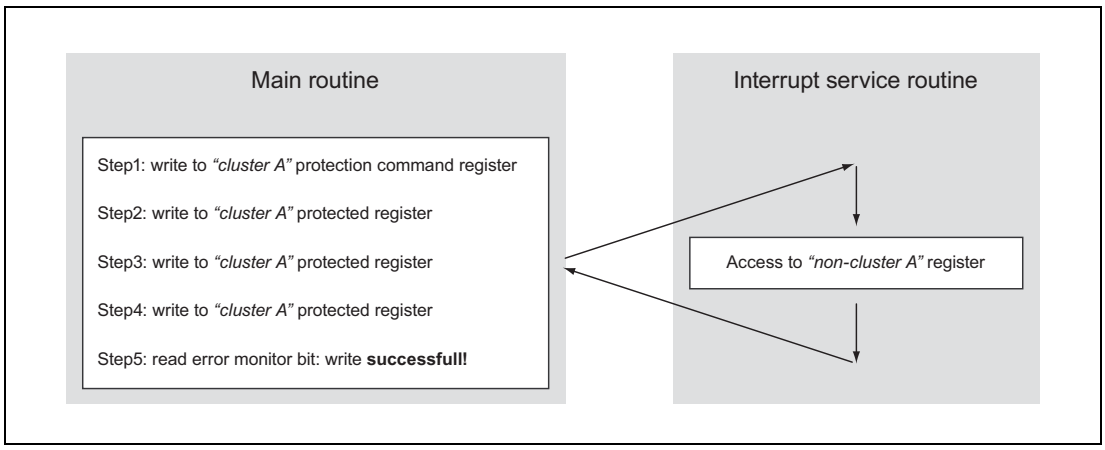


Figure 4.2 Example of Successful Protection Unlock Sequence

For more information on registers of RH850/D1L/D1M register protection clusters, see Section 4.2, RH850/D1L/D1M write-protected registers.

4.1.1 Register protection unlock sequence

Write access to a write protected register is enabled by using a special protection unlock sequence:

1. Write the fixed value 0000 $00A5_H$ to the protection command register
2. Write the desired value to the protected register.
3. Write the bit-wise inversion of the desired value to the protected register.
   (To the reserved bits, write the bit-wise inversion of the initial value.)
4. Write the desired value to the protected register.
5. Verify successful write of the desired value to the protected register by verifying that the error monitor bit in the protection status register is “0”.
   In case the write was not successful, indicated by the error monitor bit set to “1”, the entire sequence has to be restarted at step 1.

In case of any access to another register between step 1 to step 4 of the above sequence, the protection mechanism behaves as follows:

- If the second register belongs to the same cluster, the write to the protected register fails (the error monitor bit set to “1”). The entire sequence has to be restarted at step 1.
- If the second register does not belong to the same cluster, the protection unlock sequence is not disrupted and the write to the first register can be completed successfully.

For the port regisetr protection sequence, see Section 2.2.3.5, Port Register Protection.

4.1.2 Register protection and interrupt/emulation break

If an interrupt/emulation break occurs during the protection unlock sequence, the protection mechanism behaves as follows:

(1) Interrupt during the protection unlock sequence

If an interrupt is acknowledged during the above protection unlock sequence and the interrupt service routine does not access any register of the same register protection cluster, the protection unlock sequence is not disrupted and the write to the protected register can be successfully completed after returning from the interrupt service routine.

(2) Emulation break during the protection unlock sequence

If an emulation break occurs during the above protection unlock sequence, e.g. because of a breakpoint hit, the debugger does not access any register of the same register protection cluster, the protection unlock sequence is not disrupted and the write to the protected register can be successfully completed after returning from the break.

4.2 RH850/D1L/D1M write-protected registers

The following table lists the RH850/D1L/D1M write-protected registers.

Table 4.1 Write-Protection Target Registers (1/2)

Protection Target	Protected Register	Protection Register		Protection Cluster
		Command Register	Status Register	
Clock Controller	MOSCE	PROTCMD0	PROTS0	Control protection cluster 0
	SOSCE			
	ROSCE			
	CKSC_AAWOTS_CTL			
	CKSC_AAWOTD_CTL			
	CKSC_AWDTA0D_CTL			
	CKSC_ARTCAS_CTL			
	CKSC_ARTCAD_CTL			
	CKSC_AFOUTS_CTL			
Stand-by Controller	STBC0PSC			
	IOHOLD			
Reset function	SWRESA			
Clock Controller	PLL0E	PROTCMD1	PROTS1	Isolated-Area clock control protection cluster 1
	PLL1E			
	CKSC_IPLL0S_CTL			
	CKSC_IPLL1S_CTL			
	CKSC_ICPUCLKS_CTL			
	CKSC_ICPUCLKD_CTL			
Clock Controller	PLL2E	PROTCMDD1	PROTSD1	Isolated-Area clock control protection cluster 2
	CKDV_ICLKJITD_CTL			
	CKSC_ICLKJITS_CTL			
	CKDV_ICLKFIXD_CTL			
	CKSC_ICLKFIXS_CTL			
	CKSC_IPLLFIXS_CTL			
	CKSC_IXCCLKS_CTL			
	CKSC_ISDRBS_CTL			
	CKSC_IPCETNBS_CTL			
	CKSC_IXCETNBS_CTL			
	CKSC_IPCMLBBS_CTL			
	CKSC_IXCMLBBS_CTL			
	CKSC_IPCRSCANS_CTL			
	CKSC_IMLBBS_CTL			
	CKSC_ISFMAS_CTL			
	CKSC_ISFMAD_CTL			
	CKSC_IRSCAND_CTL			
	CKSC_IRSCANXINS_CTL			
	CKDV_ISSIFD_CTL			
	CKSC_ITAUB01S_CTL			
	CKSC_ITAUB2S_CTL			

Table 4.1 Write-Protection Target Registers (2/2)

Protection Target	Protected Register	Protection Register		Protection Cluster
		Command Register	Status Register	
Clock Controller	CKSC_ITAUJS_CTL	PROTCMDD1	PROTSD1	Isolated-Area clock control protection cluster 2
	CKSC_IOSTMS_CTL			
	CKSC_ILCBIS_CTL			
	CKSC_IADCED_CTL			
	CKSC_IISMS_CTL			
	CKSC_IRLINS_CTL			
Video channels clock generators	CKSC_IPLL0PIXS_CTL			
	CKDV_IPLL2IND_CTL			
	CKSC_IPLL2INS_CTL			
	CKSC_IPLL2CLKS_CTL			
	CKSC_IDOTCLK0S_CTL			
	CKSC_IDOTCLK1S_CTL			
	CKDV_IDOTCLK0D_CTL			
	CKDV_IDOTCLK1D_CTL			
	CKSC_IVDCE0VOS_CTL			
	CKSC_IVDCE1VOS_CTL			
	CKSC_IVOEXS_CTL			
	CKSC_IRSDSS_CTL			
	CKSC_IVDCE0VIS_CTL			
	CKSC_IMIPIPLLS_CTL			
	CKSC_IMIPIPIXD_CTL			
Clock Monitors	CLMAnCTL0	CLMAnPCMD	CLMAnPS	Clock Monitors control protection clusters
	CLMATEST	PROTCMDCLMA	PROTSCLMA	
	CLMATEST2	PROTCMDCLMA2	PROTSCLMA2	
	CLMATEST3	PROTCMDCLMA3	PROTSCLMA3	
Reset function	MRSTC	PROTCMDMRST	PROTSMRST	System control protection clusters
Wake-up timing control	PWRGD_CNT	PROTCMDPWRGD	PROTSPWRGD	
Clock Controller	APB_CLK_RATIO			
On-Chip Debug control	IDMODI	PROTCMDIDMODI	PROTSIDMODI	
Port control*1	PINVn, PODCn, PDSCn of AWO ports	PPCMDn	PPROTSn	Port control protection clusters
	PINVn, PODCn, PDSCn of ISO ports	PPCMDn	PPROTSn	
	JPINVn, JPODCn, JPDSCn of JP0 ports	JPPCMDn	JPPROTSn	
Self-programming function	FLMDCNT	FLMDPCMD	FLMDPS	Self-programming protection clusters

Note 1. Each port group has its own protection command register and status register. For details, see **Section 4.2.1, Port protection clusters** on the next page.

4.2.1 Port protection clusters

The following port control registers have write protection:

- Port output level inversion register (PINVn, JPINV0)
- Port open drain control registers (PODCn, JPODC0)
- Port drive strength control registers (PDSCn)

For the port regisetr protection sequence, see Section 2.2.3.5, Port Register Protection.

4.2.2 Overview of RH850/D1L/D1M protection registers

Register write protection is controlled and operated by using the following registers.

Table 4.2 Overview of Protection Command Registers (1/2)

Module Name	Register Name	Symbol	Address
Control protection clusters			
SYS	Protection command register 0	PROTCMD0	FFF8 0000$_H$
SYS	Protection status register 0	PROTS0	FFF8 0004$_H$
Isolated-Area clock control protection clusters			
SYS	Protection command register 1	PROTCMD1	FFF8 8000$_H$
SYS	Protection status register 1	PROTS1	FFF8 8004$_H$
SYS	Protection command register	PROTCMDD1	FFF8 7000$_H$
SYS	Protection status register	PROTSD1	FFF8 7004$_H$
Clock Monitors control protection cluster			
	Protection command register 0	CLMA0PCMD	FFF8 C010$_H$
	Protection status register 0	CLMA0PS	FFF8 C014$_H$
	Protection command register 1	CLMA1PCMD	FFF8 F010$_H$
	Protection status register 1	CLMA1PS	FFF8 F014$_H$
	Protection command register 2	CLMA2PCMD	FFF8 F090$_H$
	Protection status register 2	CLMA2PS	FFF8 F094$_H$
	Protection command register 3	CLMA3PCMD	FFF8 F110$_H$
	Protection status register 3	CLMA3PS	FFF8 F114$_H$
	Protection command register 4	CLMA4PCMD	FFF8 F190$_H$
	Protection status register 4	CLMA4PS	FFF8 F194$_H$
	Protection command register 5	CLMA5PCMD	FFF8 F210$_H$
	Protection status register 5	CLMA5PS	FFF8 F214$_H$
	Protection command register 6	CLMA6PCMD	FFF8 F290$_H$
	Protection status register 6	CLMA6PS	FFF8 F294$_H$
CLMAC	Protection command register	PROTCMDCLMA	FFF8 C200$_H$
CLMAC	Protection status register	PROTSCLMA	FFF8 C204$_H$
CLMAC	Protection command register	PROTCMDCLMA2	FFF8 F680$_H$
CLMAC	Protection status register	PROTSCLMA2	FFF8 F684$_H$
CLMAC	Protection command register	PROTCMDCLMA3	FFF8 F380$_H$
CLMAC	Protection status register	PROTSCLMA3	FFF8 F384$_H$
System control protection clusters			
SYS	Protection command register	PROTCMDMRST	FFF8 F480$_H$
SYS	Protection status register	PROTSMRST	FFF8 F484$_H$

Table 4.2 Overview of Protection Command Registers (2/2)

Module Name	Register Name	Symbol	Address
PWRG	Protection command register	PROTCMDPWRGD	FFF8 F580$_H$
PWRG	Protection status register	PROTSPWRGD	FFF8 F584$_H$
IDMP	Protection command register	PROTCMDIDMODI	FFF9 2000$_H$
IDMP	Protection status register	PROTSIDMODI	FFF9 2004$_H$
Port protection clusters			
PORT_AWO	Protection command registers for P0	PPCMD0	FFC1 C02C$_H$
PORT_AWO	Protection status registers for P0	PPROTS0	FFC1 C034$_H$
PORT_ISO	Protection command registers for P1	PPCMD1	FFC1 406C$_H$
PORT_ISO	Protection status registers for P1	PPROTS1	FFC1 4074$_H$
PORT_ISO	Protection command registers for P2	PPCMD2	FFC1 40AC$_H$
PORT_ISO	Protection status registers for P2	PPROTS2	FFC1 40B4$_H$
PORT_ISO	Protection command registers for P3	PPCMD3	FFC1 40EC$_H$
PORT_ISO	Protection status registers for P3	PPROTS3	FFC1 40F4$_H$
PORT_ISO	Protection command registers for P10	PPCMD10	FFC1 42AC$_H$
PORT_ISO	Protection status registers for P10	PPROTS10	FFC1 42B4$_H$
PORT_ISO	Protection command registers for P11	PPCMD11	FFC1 42EC$_H$
PORT_ISO	Protection status registers for P11	PPROTS11	FFC1 42F4$_H$
PORT_ISO	Protection command registers for P16	PPCMD16	FFC1 442C$_H$
PORT_ISO	Protection status registers for P16	PPROTS16	FFC1 4434$_H$
PORT_ISO	Protection command registers for P17	PPCMD17	FFC1 446C$_H$
PORT_ISO	Protection status registers for P17	PPROTS17	FFC1 4474$_H$
PORT_ISO	Protection command registers for P21	PPCMD21	FFC1 456C$_H$
PORT_ISO	Protection status registers for P21	PPROTS21	FFC1 4574$_H$
PORT_ISO	Protection command registers for P22	PPCMD22	FFC1 45AC$_H$
PORT_ISO	Protection status registers for P22	PPROTS22	FFC1 45B4$_H$
PORT_ISO	Protection command registers for P40	PPCMD40	FFC1 4A2C$_H$
PORT_ISO	Protection status registers for P40	PPROTS40	FFC1 4A34$_H$
PORT_ISO	Protection command registers for P42	PPCMD42	FFC1 4AAC$_H$
PORT_ISO	Protection status registers for P42	PPROTS42	FFC1 4AB4$_H$
PORT_ISO	Protection command registers for P43	PPCMD43	FFC1 4AEC$_H$
PORT_ISO	Protection status registers for P43	PPROTS43	FFC1 4AF4$_H$
PORT_ISO	Protection command registers for P44	PPCMD44	FFC1 4B2C$_H$
PORT_ISO	Protection status registers for P44	PPROTS44	FFC1 4B34$_H$
PORT_ISO	Protection command registers for P45	PPCMD45	FFC1 4B6C$_H$
PORT_ISO	Protection status registers for P45	PPROTS45	FFC1 4B74$_H$
PORT_ISO	Protection command registers for P46	PPCMD46	FFC1 4BAC$_H$
PORT_ISO	Protection status registers for P46	PPROTS46	FFC1 4BB4$_H$
PORT_ISO	Protection command registers for P47	PPCMD47	FFC1 4BEC$_H$
PORT_ISO	Protection status registers for P47	PPROTS47	FFC1 4BF4$_H$
PORTJ_AWO	Protection command registers for JP0	JPPCMD0	FFC2 402C$_H$
PORTJ_AWO	Protection status registers for JP0	JPPROTS0	FFC2 4034$_H$
Self-programming protection clusters			
	Protection command register	FLMDPCMD	FFA0 0004$_H$
	Protection status register	FLMDPS	FFA0 0008$_H$

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

4.2.3 Control protection cluster registers

4.2.3.1 PROTCMD0 — Protection command register

This register is used to initiate the write protection unlock sequence for write-protected registers.

Access: This register can be written in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: Reading this register returns 0000 0000_H.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PCMD0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 4.3 PROTCMD0 Register Contents

Bit Position	Bit Name	Function
7 to 0	PCMD0[7:0]	Protection commands to enable writing to control protection cluster registers

4.2.3.2 PROTS0 — Protection status register

This register indicates the status of the protection unlock sequence performed by PROTCMD0.

Access: This register is a read-only register that can be read in 32-bit units.
Write operation is ignored.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PROT ERR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4.4 PROTS0 Register Contents

Bit Position	Bit Name	Function
0	PROTERR	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

4.2.4 Isolated-Area clock control protection clusters

4.2.4.1 PROTCMD1 — Protection command register

This register is used to initiate the write protection unlock sequence for write-protected registers.

Access: This register can be written in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: Reading this register returns 0000 0000_H.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PCMD1[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 4.5 PROTCMD1 Register Contents

Bit Position	Bit Name	Function
7 to 0	PCMD1[7:0]	Protection commands to enable writing to control protection cluster registers

4.2.4.2 PROTS1 — Protection status register

This register indicates the status of the protection unlock sequence performed by PROTCMD1.

Access: This register is a read-only register that can be read in 32-bit units.
Write operation is ignored.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PROT ERR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4.6 PROTCMD1 Register Contents

Bit Position	Bit Name	Function
0	PROTERR	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

4.2.4.3 PROTCMDD1 — Protection command register

This register is used to initiate the write protection unlock sequence for write-protected registers.

Access: This register can be written in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: Reading this register returns 0000 0000_H.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PCMDD1[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 4.7 PROTCMDD1 Register Contents

Bit Position	Bit Name	Function
7 to 0	PCMDD1[7:0]	Protection commands to enable writing to control protection cluster registers

4.2.4.4 PROTSD1 — Protection status register

This register indicates the status of the protection unlock sequence performed by PROTCMDn.

Access: This register is a read-only register that can be read in 32-bit units.
Write operation is ignored.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PROTD ERR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4.8 PROTCMSD1 Register Contents

Bit Position	Bit Name	Function
0	PROTD ERR	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

4.2.5 Clock Monitors protection cluster registers

4.2.5.1 CLMAnPCMD — CLMAn protection command register

This register is a protection command register for the CLMAnCTL0 register.

Index n

An index “n” denotes the number of protection command registers. For details, see Table 4.1, Write-Protection Target Registers.

Access: This register can be written in 8-bit units.
Address: See Table 4.2, Overview of Protection Command Registers.
Initial value: Reading this register returns an undefined value.

Bit	7	6	5	4	3	2	1	0
	CLMAnREG[7:0]							
Initial value								
R/W	W	W	W	W	W	W	W	W

Table 4.9 CLMAnPCMD Register Contents

Bit Position	Bit Name	Function
7 to 0	CLMAn REG[7:0]	Protection command that enables writing to the CLMAnCTL0 register

4.2.5.2 CLMAnPS — CLMAn protection status register

This register is used to verify whether the write-protected register (CLMAnCTL0) has been successfully written or not.

Index n

An index “n” denotes the number of protection command registers. For details, see Table 4.1, Write-Protection Target Registers.

Access: This register can be read in 8-bit units.
Address: See Table 4.2, Overview of Protection Command Registers.
Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	CLMAnPRERR
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 4.10 CLMAnPS Register Contents

Bit Position	Bit Name	Function
0	CLMAn PRERR	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

4.2.5.3 PROTCMDCLMA — Clock monitor test protection command register

This register is a protection command register for the CLMATEST register.

Access: This register can be written in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: Reading this register returns an undefined value.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLMATREG[7:0]							
Initial value																
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 4.11 PROTCMDCLMA Register Contents

Bit Position	Bit Name	Function
7 to 0	CLMATREG[7:0]	Protection command that enables writing to CLMATEST

4.2.5.4 PROTSCLMA — Clock monitor test protection status register

This register is used to verify whether the write-protected register (CLMATEST) has been successfully written or not.

Access: This register can be read in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLMAT PRERR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4.12 PROTSCLMA Register Contents

Bit Position	Bit Name	Function
0	CLMATPRERR	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

4.2.5.5 PROTCMDCLMA2 — Clock monitor test protection command register 2

This register is a protection command register for the CLMATEST2 register.

Access: This register can be written in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: Reading this register returns an undefined value.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLMATREG2[7:0]							
Initial value																
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 4.13 PROTCMDCLMA2 Register Contents

Bit Position	Bit Name	Function
7 to 0	CLMA TREG2[7:0]	Protection command that enables writing to CLMATEST2

4.2.5.6 PROTSCLMA2 — Clock monitor test protection status register 2

This register is used to verify whether the write-protected register (CLMATEST2) has been successfully written or not.

Access: This register can be read in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLMA TPR ERR2
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4.14 PROTSCLMA2 Register Contents

Bit Position	Bit Name	Function
0	CLMATPRERR2	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

4.2.5.7 PROTCMDCLMA3 — Clock monitor test protection command register3

This register is a protection command register for the CLMATEST3 register.

Access: This register can be written in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: Reading this register returns an undefined value.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLMATREG3[7:0]							
Initial value																
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 4.15 PROTCMDCLMA3 Register Contents

Bit Position	Bit Name	Function
7 to 0	CLMA TREG3[7:0]	Protection command that enables writing to CLMATEST3

4.2.5.8 PROTSCLMA3 — Clock monitor test protection status register

This register is used to verify whether the write-protected register (CLMATEST3) has been successfully written or not.

Access: This register can be read in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLMA TPR ERR3
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4.16 PROTSCLMA3 Register Contents

Bit Position	Bit Name	Function
0	CLMATPRERR3	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

4.2.6 System control protection register

4.2.6.1 PROTCMDMRST — Reset control protection command register

This register is a protection command register for the MRSTC register.

Access: This register can be written in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: Reading this register returns an undefined value.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MRSTCPREG[7:0]							
Initial value																
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 4.17 PROTCMDMRST Register Contents

Bit Position	Bit Name	Function
7 to 0	MRSTC PREG[7:0]	Protection command that enables writing to MRSTC

4.2.6.2 PROTSMRST — Reset protection status register

This register is used to verify whether the write-protected register (MRSTC) has been successfully written or not.

Access: This register can be read in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MRSTC PERR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4.18 PROTSMRST Register Contents

Bit Position	Bit Name	Function
0	MRSTC PERR	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

4.2.6.3 PROTCMDPWRGD — PWR_GD protection command register

This register is a protection command register for the PWR_GD and APB_CLK_RATIO register.

Access: This register can be written in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: Reading this register returns an undefined value.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PWRGDPRREG[7:0]							
Initial value																
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 4.19 PROTCMDPWRGD Register Contents

Bit Position	Bit Name	Function
7 to 0	PWRGD PRREG[7:0]	Protection command that enables writing to PWR_GD and APB_CLK_RATIO

4.2.6.4 PROTSPWRGD — PWR_GD protection status register

This register is used to verify whether the write-protected register (PWR_GD and APB_CLK_RATIO) has been successfully written or not.

Access: This register can be read in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWRGD PRERR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4.20 PROTSPWRGD Register Contents

Bit Position	Bit Name	Function
0	PWRGD PRERR	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

4.2.6.5 PROTCMDIDMODI — On-Chip Debug control protection command register

This register is a protection command register for the IDMODI register.

Access: This register can be written in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: Reading this register returns an undefined value.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value																
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	IDMODIPREG[7:0]							
Initial value																
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 4.21 PROTCMDIDMODI Register Contents

Bit Position	Bit Name	Function
7 to 0	IDMODI PREG[7:0]	Protection command that enables writing to IDMODI

4.2.6.6 PROTSIDMODI — On-Chip Debug control protection status register

This register is used to verify whether the write-protected register (IDMODI) has been successfully written or not.

Access: This register can be read in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IDMODI PERR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4.22 PROTSIDMODI Register Contents

Bit Position	Bit Name	Function
0	IDMODI PERR	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

4.2.7 Port protection cluster registers

4.2.7.1 PPCMDn — Port protection command register

PPCMDn is a protection command register for port group n.

Index n

An index “n” denotes the number of protection command registers. For details, see Table 4.1, Write-Protection Target Registers.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: Reading this register returns 0000 0000_H.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PPCMDn[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

NOTE

The protection command register for port group JP0 is JPPCMD0. Its bits are JPPCMD0[7:0].

Table 4.23 PPCMDn Register Contents

Bit Position	Bit Name	Function
7 to 0	PPCMDn[7:0]	Protection command that enables writing to port protection cluster registers

4.2.7.2 PPROTSn — Port protection status register

PPROTSn is a protection status register for write-protected registers of port group n. It indicates the status of the protection sequence operated by PPCMDn.

Index n

An index “n” denotes the number of protection command registers. For details, see Table 4.1, Write-Protection Target Registers.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PPROTSnPRERR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

NOTE

The protection status register for port group JP0 is JPPROTS0. Its bit is JPPROTS0PRERR.

Table 4.24 PPROTSn Register Contents

Bit Position	Bit Name	Function
0	PPROTSn PRERR	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

4.2.8 Details of self-programming protection cluster registers

4.2.8.1 FLMDPCMD — FLMD protection command register

FLMDPCMD is a protection command register for the FLMDCNT register.

Access: This register is a write-only register that can be written in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: Reading this register returns 0000 0000_H.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	FLMDPC[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 4.25 FLMDPCMD Register Contents

Bit Position	Bit Name	Function
7 to 0	FLMDPC[7:0]	Protection command that enables writing to self-programming protection cluster registers

4.2.8.2 FLMDPS — FLMD protection error status register

This register is used to verify whether the write-protected register (FLMDCNT) has been successfully written or not.

Access: This register is a read-only register that can be read in 32-bit units.

Address: See Table 4.2, Overview of Protection Command Registers.

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FLMDPRERR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4.26 FLMDPS Register Contents

Bit Position	Bit Name	Function
0	FLMDPRERR	Protection write sequence error monitor 0: No protection error 1: Protection error occurred

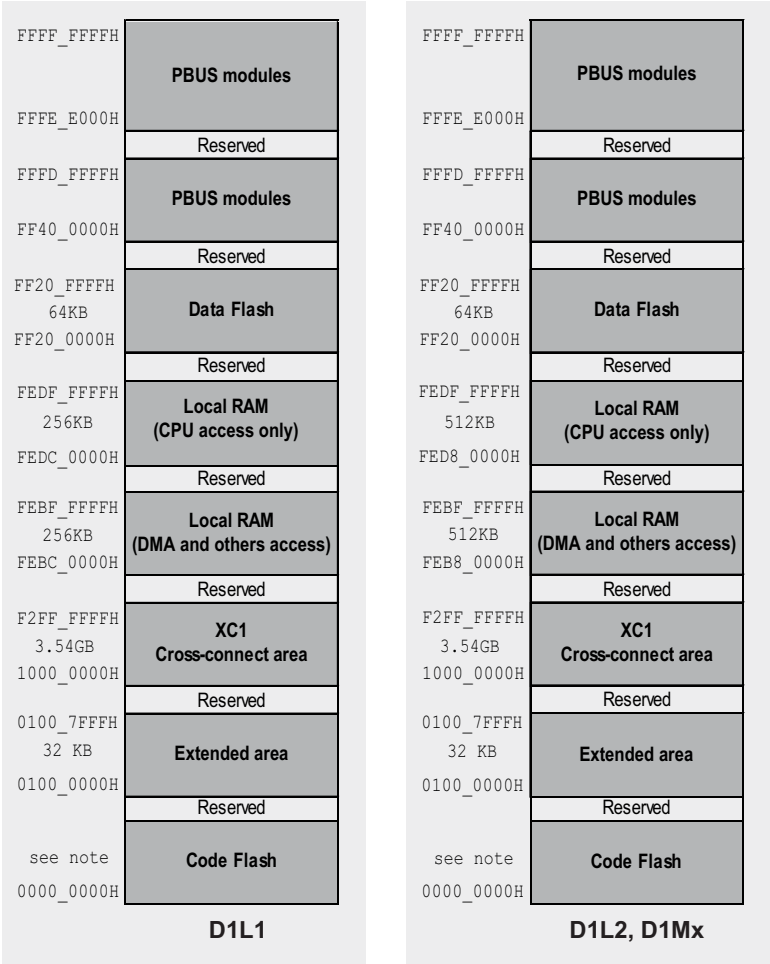
Section 5 Address Spaces

CAUTION

When making an access to the on-chip I/O register space, access the addresses shown in this manual. Do not access an address that is reserved or not specified in this manual. Otherwise, operation is not guaranteed.

5.1 CPU address map

The following diagram shows the address map from the CPU's perspective.



Note: The Code Flash memory is device dependent:

- D1L1: 2 MB, address space 0000 0000_H to 001F $FFFF_H$
- D1L2(H), D1M1-V2, D1M1A: 4 MB, address space 0000 0000_H to 003F $FFFF_H$
- D1M1(H)_3.75MB, D1M2(H)_3.75MB: 3.75 MB, address space 0000 0000_H to 003B $FFFF_H$
- D1M1(H)_5MB, D1M2(H)_5MB: 5 MB, address space 0000 0000_H to 004F $FFFF_H$

NOTE

Initial value of CPU reset vector

Reset vector is depended RBASE register settings. And, RBASE initial value is 0000 0000H.

RBASE register is described by "RH850 Family User's Manual: Software".

Figure 5.1 CPU address map

Local RAM

The Local RAM is accessible through two address areas in the CPU address map:

- The Local RAM area starting from address FEDC 0000_H (D1L1) or FED8 0000_H (D1L2, D1Mx) is accessible only by the CPU.
- The Local RAM area starting from address FEBC 0000_H (D1L1) or FEB8 0000_H (D1L2, D1Mx) is accessible also by all other masters, like the DMA Controller and all bus masters of the cross-connect system.

NOTE

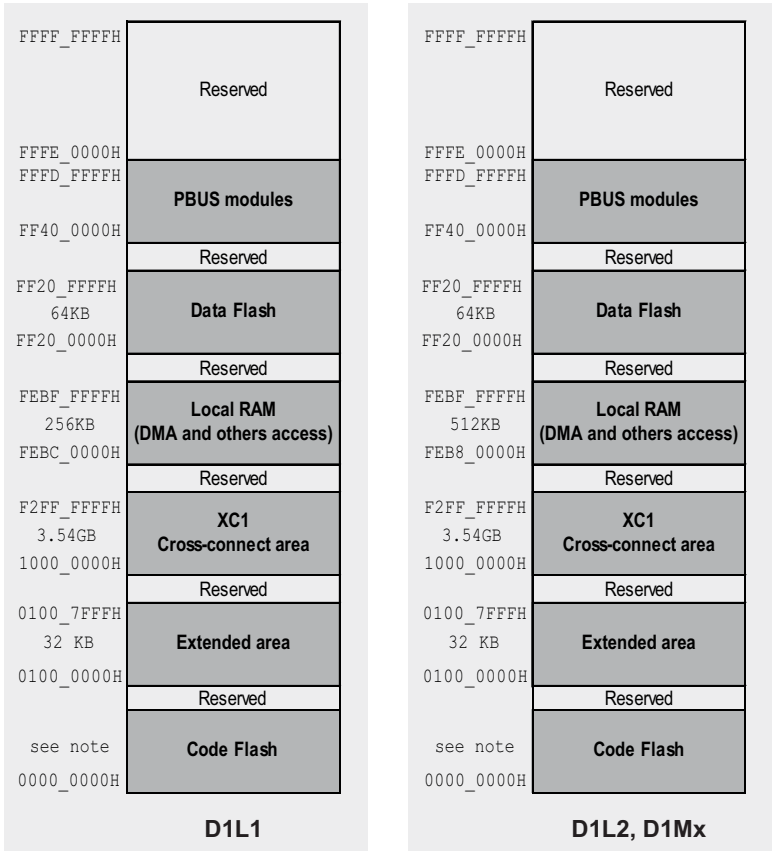
Though the CPU can access the Local RAM via both address areas, it is highly recommended to use the exclusive area at FEDC 0000_H (D1L1) or FED8 0000_H (D1L2, D1Mx).

CPU access to on-chip and external memories

The CPU can access the video RAMs and all external memories via the 3.54 GB large memory window to the XC1 cross-connect.

5.2 DMA and XC1 master's address map for the CPU Subsystem

The following diagram shows the address map of the CPU Subsystem from the DMA Controllers and the other XC1 cross-connect master's perspective.



Note: The Code Flash memory is device dependent:

- D1L1: 2 MB, address space 0000 0000_H to 001F $FFFF_H$
- D1L2(H), D1M1-V2, D1M1A: 4 MB, address space 0000 0000_H to 003F $FFFF_H$
- D1M1(H)_3.75MB, D1M2(H)_3.75MB: 3.75 MB, address space 0000 0000_H to 003B $FFFF_H$
- D1M1(H)_5MB, D1M2(H)_5MB: 5 MB, address space 0000 0000_H to 004F $FFFF_H$

Figure 5.2 DMA address map

5.3 Cross-connect address map

The following diagram shows the address map of the cross-connect bus system from all master's perspective.

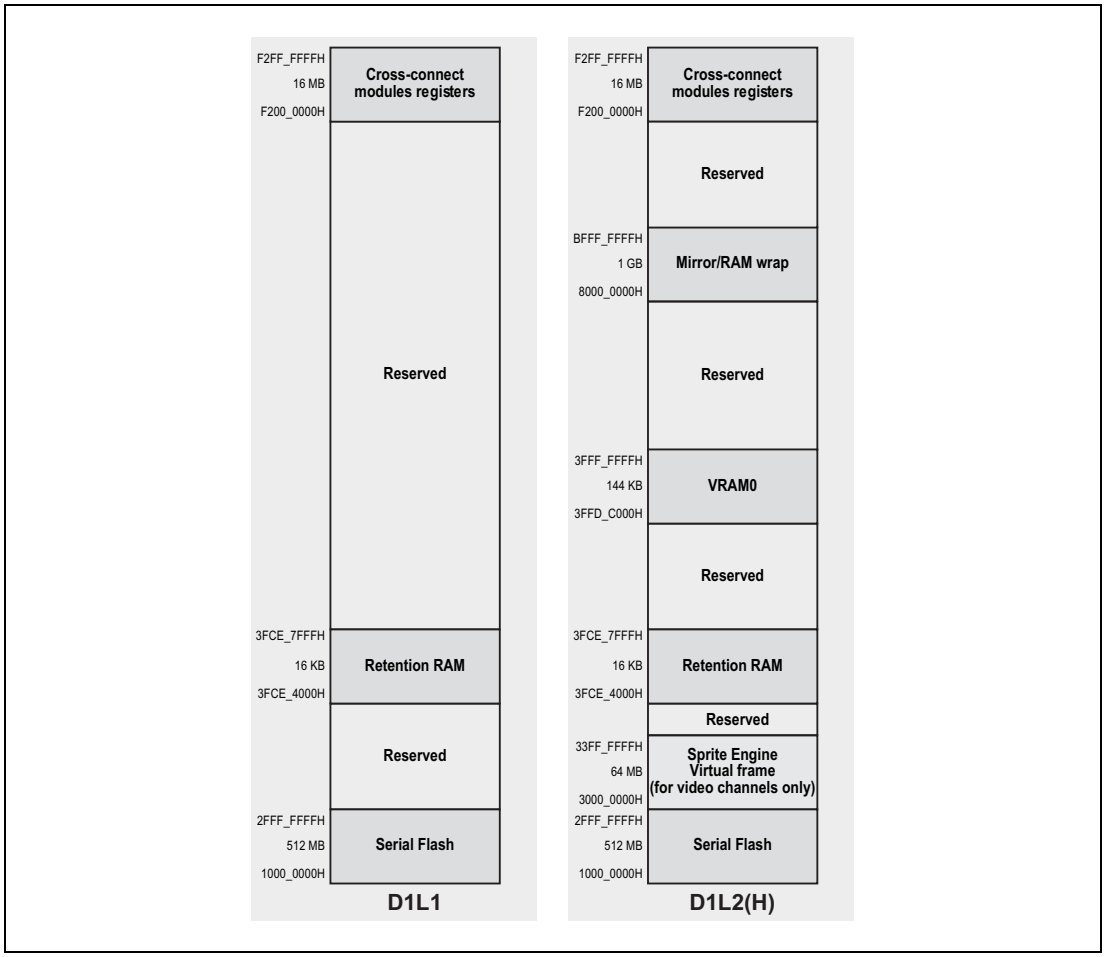
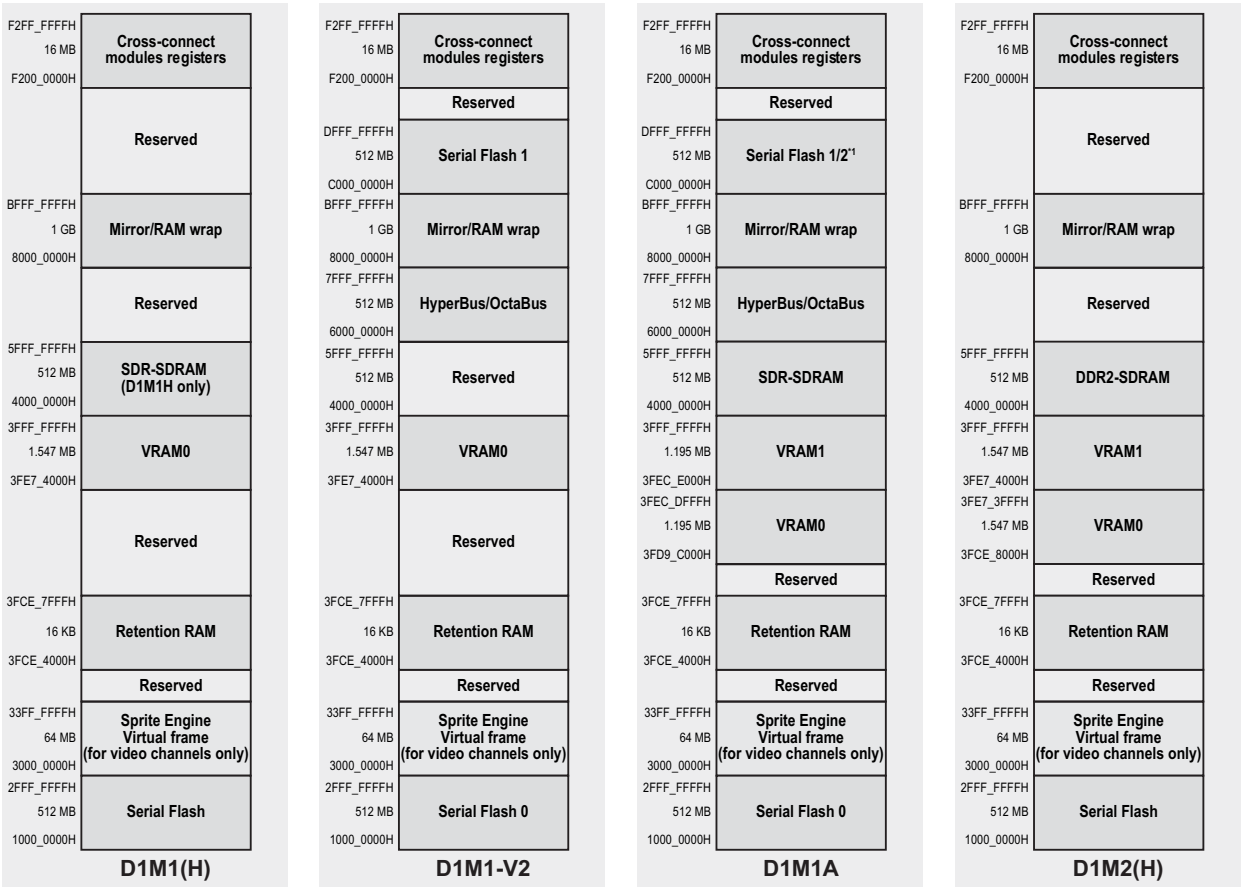


Figure 5.3 D1Lx cross-connect address map



Note 1. SFMA1 and SFMA2 cannot be used concurrently. One of them must be selected by XCRAMCFG0.SFMASEL. Refer to Section 14.4, Bus Switch for external memory interfaces (D1M1-V2, D1M1A only) for details.

Figure 5.4 D1Mx cross-connect address map

The Mirror/RAM wrap area contains different address mapping images of the Video RAMs VRAM0/VRAM1, the SDRAM and the serial flash area.

Purpose of the mirrored areas are

- optimized memory mapping for different colour modes
- possibility for cached/uncached access from CPU.

Refer to Section 54, Video RAM and Video RAM Wrapper (VRAM) for details.

Section 6 Operating Modes

This section describes the operating mode and mode selection of the RH850/D1L/D1M.

The RH850/D1L/D1M has the operating modes as shown below.

Operating modes

- Normal operating mode

  This mode is for execution of application software. The on-chip debug capabilities also use this mode. If FLMD0 is activated while the operation is in this mode, writing to the code flash memory through self programming is enabled.

- Flash serial programming mode

  The flash programmer enables erasing/writing to the on-chip flash memory.

- Boundary scan mode

  This mode allows boundary scan tests compliant with IEEE Standard 1149.1.

When an external reset or power-on reset *1 is generated, RH850/D1L/D1M latches the state of the FLMD0, FLMD1, MODE0, and MODE1 pins and decides the operating mode after the reset is canceled. **Table 6.1** lists the relationship between the pin settings and the operating mode.

Note 1. The external reset includes all SYSRES in normal operation mode because RESET output is available.

Table 6.1 Selection of Operating Modes

Pins				
FLMD0*1	FLMD1 (P0_1)	MODE0 (P0_0)	MODE1 (P0_2)	Operating Mode
0	x	x	x	Normal operation mode*2
1	0	x	x	Flash programming mode
1	1	0	1	Boundary scan mode
Other than above				Settings are prohibited

Note 1. FLMD0 is not allowed to change its state during operation.
Note 2. Except the case using self-programming function.

CAUTION

To change operating mode, restart from POC0RES (remove the power supply once and apply it again). In the case of only by the external reset, some functions are not initialized after the mode transitions. For details of functions not initialized by the external reset, see Section 9.2.2, Reset sources and targets.

Section 7 Interrupt

The interrupt controller (INTC) determines priority of interrupt sources and controls interrupt requests to the CPU. The INTC has a register to set priority of each interrupt. Interrupt requests are processed according to the priority set in this register.

7.1 Overview

- One non-maskable interrupts (FENMI), one FE level interrupt (FEINT), and 256 EI level interrupts
  - FENMI interrupt: FE level non-maskable interrupt from ECM
  - FEINT interrupt: FE level maskable interrupt from ECM
  - Maskable interrupts: 256
    – 32 high-speed interrupts
    – 224 low-speed interrupts
    Refer to Section 7.6, Interrupt Response Times for details concerning the different interrupt response times.
- Sixteen interrupt priority levels can be set for the EI level interrupts.
  Up to 16 priority levels of the EI level interrupts can be set in EI level interrupt control registers.
- Three detection methods
  A method of detecting the pin NMI interrupt and external interrupt can be selected from rising edge, falling edge, and both edges.
- Direct branching method or table referencing method is selectable by setting two types of interrupt handler address setting registers.

7.2 Register Specifications

The INTC has registers listed in tables below. These registers are mainly used to set interrupt priority and to control detection of external interrupt input signals.

Table 7.1 Index

Index	Meaning
n	Indicating EI level interrupt source number (n = 0 - 255)
m	Indicating EI level interrupt mask register number (m = 0 - 7)
x	Indicating INTP number (x = 0 - 10)

7.2.1 Register Configuration

Table 7.2 Interrupt Control

Address	Symbol	Register Name	R/W	Initial Value	Access Size
FFFE EA00$_H$ - FFFE EA3E$_H$	EIC0 to EIC31	EI level interrupt control register	R/W	008F$_H$*1 808F$_H$*2	8 bit, 16 bit
FFFF B040$_H$ - FFFF B1FE$_H$	EIC32 to EIC255				
FFFE EAF0$_H$	IMR0	EI level interrupt mask register	R/W	FFFF FFFF$_H$	8 bit, 16 bit, 32 bit
FFFF B404$_H$ - FFFF B41C$_H$	IMR1 to IMR7				
FFFE EA78$_H$	FNC	FE level NMI status register	R	0000$_H$	8 bit, 16 bit
FFFE EA7A$_H$	FIC	FE level interrupt status register	R	0000$_H$	8 bit, 16 bit

Note 1. When a synchronous edge is detected.
Note 2. When a high-level signal is detected.

INTC1 registers

Among the registers shown in **Table 7.2**, the EIC0 to 31, IMR0, FNC, and FIC are located in INTC1 of the CPU-specific Peripheral. Each of these registers only can be accessed from the CPU which includes it.Writing is only possible in supervisor mode (PSW.UM = 0).

INTC2 registers

Of the registers listed in **Table 7.2**, EIC32 to EIC255, and IMR1 to IMR7 are located in INTC2, the controller for interrupts from peripheral IP group 0. Writing to these registers is only possible for the PE in supervisor mode. When writing to IMR1 to IMR7, only the bits corresponding to the conditions described above are overwritten; other bits are not updated.

CAUTION

The INTC2 registers are not accessible per default as these registers are guarded by the PBUS Guard PBG0B.

Before accessing any of these registers the PBG0B protection register FSGD0BPROT0 has to be set to 07FF FFFF$_H$.

See Section 14.6.4, PBUS Guards (PBG) for further details.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> is defined as "INTC1" or “INTC2” for the registers in the table above.

<Symbol> is defined in the above table.

7.2.2 EI Level Interrupt Control Registers 0 to 255 (EIC0 - EIC255)

These registers are used to set interrupt control conditions for each EI level interrupt source. One register is provided for each source of this type. Writing to these registers requires care when a register is set for detection in synchronous edge mode. If 0 is written to the EIRFn bit immediately after a peripheral module generates the corresponding interrupt request, the request may be inadvertently lost. On the other hand, writing 1 to the EIRFn bit immediately after an interrupt is accepted by the CPU may lead to incorrect re-issuing of the request. Writing to any of these registers, therefore, should only proceed while peripheral modules are not generating interrupt requests and the CPU has not accepted any interrupt.

Writing to an EICn register includes writing by using the bit operation instructions (set1, clr1, and not1). The bit operation instructions proceed in the following sequence (1) reading the register, (2) processing the specified bit, and (3) writing the new value back to the register. When a bit operation instruction has a bit other than EIRFn as the operand, the value read in step (1) will be written back in step (3). Accordingly, the issuance of interrupt requests from peripheral modules or the acceptance of an interrupt request by the CPU between steps (1) and (3) may lead to the problems described above.

The addresses of the registers corresponding to the interrupt channels that are listed as reserved in the tables in 7.4, Interrupt Exception Handler and Priority Operations, are also reserved. Therefore, access to these addresses is prohibited. Operation cannot be guaranteed if they are accessed.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EICTn	—	—	EIRFn	—	—	—	—	EIMKn	EITBn	—	—	EIP3n	EIP2n	EIP1n	EIP0n
Initial value	*1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1
R/W	R	R	R	R/W	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W

Note 1. 0: Synchronous edge detection, 1: High-level detection

Table 7.3 EIC0 - EIC255 register contents (1/2)

Bit Position	Bit Name	Function
15	EICTn	Interrupt Channel Type The following values are read according to the interrupt input interface. This is a read-only bit. 0: Synchronous edge detection 1: High-level detection
14, 13	—	Reserved These bits are always read as 0. The write value should always be 0.
12	EIRFn	Interrupt Request Flag Operation varies with the interrupt input interface. 0: No interrupt request (Initial value) 1: Interrupt request present • Synchronous edge detection This flag is automatically cleared to 0 when an interrupt request of the self-channel is accepted by the CPU core. This bit can be set or cleared by the software. • High level detection This bit cannot be set or cleared by the software. This is a read-only bit.
11 to 8	—	Reserved These bits are always read as 0. The write value should always be 0.

Table 7.3 EIC0 - EIC255 register contents (2/2)

Bit Position	Bit Name	Function
7	EIMKn	Interrupt Mask While this bit is set to 1, interrupt requests set in the interrupt request flag (EIRFn) are masked to inhibit interrupt requests from the channel to the CPU core. Notification of presence of unprocessed interrupts is not made and the PMEI bit in ICSR is not set from channels for which this bit is set to 1. Even when interrupt processing is disabled by the setting of this bit, an input of interrupt signal is not masked and the interrupt request flag is set. Setting this bit is also reflected in the setting of corresponding bit of the interrupt mask register (IMR). 0: Interrupt processing is enabled. 1: Interrupt processing is disabled. (Initial value)
6	EITBn	Interrupt Vector Method Select 0: Direct branching method based on priority 1: Table referencing method
5, 4	—	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	EIP3n-0n	These bits specify 16 interrupt priority levels (0: highest priority, 15: lowest priority). If two or more EI level interrupt requests are generated simultaneously, a source with higher priority specified by these bits is selected and is sent to the CPU core. If the priority specified by these bits is equal, a source of smaller channel number is selected as fixed priority.

Note: n = 0 to 255

7.2.3 EI Level Interrupt Mask Registers 0 to 7 (IMR0 - IMR7)

These registers are aggregation of the EIMK bit in the EIC register. Setting of the corresponding EIMK bit is reflected in each bit in the IMRm register. Furthermore, setting of the IMRm register is reflected in the corresponding EIMK bit.

The bits corresponding to the interrupt channels that are listed as reserved in in the tables in **7.4, Interrupt Exception Handler and Priority Operations**, are also reserved. When read, an initial value is returned. When written, write an initial value.

IMR0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMR0H	EIMK31	EIMK30	EIMK29	EIMK28	EIMK27	EIMK26	EIMK25	EIMK24	EIMK23	EIMK22	EIMK21	EIMK20	EIMK19	EIMK18	EIMK17	EIMK16
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMR0L	EIMK15	EIMK14	EIMK13	EIMK12	EIMK11	EIMK10	EIMK9	EIMK8	EIMK7	EIMK6	EIMK5	EIMK4	EIMK3	EIMK2	EIMK1	EIMK0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IMR1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMR1H	EIMK63	EIMK62	EIMK61	EIMK60	EIMK59	EIMK58	EIMK57	EIMK56	EIMK55	EIMK54	EIMK53	EIMK52	EIMK51	EIMK50	EIMK49	EIMK48
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMR1L	EIMK47	EIMK46	EIMK45	EIMK44	EIMK43	EIMK42	EIMK41	EIMK40	EIMK39	EIMK38	EIMK37	EIMK36	EIMK35	EIMK34	EIMK33	EIMK32
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

:

:

IMR7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMR7H	EIMK 255	EIMK 254	EIMK 253	EIMK 252	EIMK 251	EIMK 250	EIMK 249	EIMK 248	EIMK 247	EIMK 246	EIMK 245	EIMK 244	EIMK 243	EIMK 242	EIMK 241	EIMK 240
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMR7L	EIMK 239	EIMK 238	EIMK 237	EIMK 236	EIMK 235	EIMK 234	EIMK 233	EIMK 232	EIMK 231	EIMK 230	EIMK 229	EIMK 228	EIMK 227	EIMK 226	EIMK 225	EIMK 224
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7.2.4 FE Level NMI Status Register (FNC)

This register indicates the status of FE level NMI conditions.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FNRF	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.4 FNC register contents

Bit Position	Bit Name	Function
15 to 13	—	Reserved. The write value should always be 0.
12	FNRF	Interrupt Request Flag 0: No interrupt request (Initial value) 1: An interrupt request present This bit is automatically cleared to 0 when an FE level NMI interrupt request is accepted by the CPU core.
11 to 0	—	Reserved. The write value should always be 0.

7.2.5 FE Level Interrupt Status Register (FIC)

This register indicates the status of FE level interrupt conditions.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	FIRF	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.5 FIC register contents

Bit Position	Bit Name	Function
15 to 13	—	Reserved. The write value should always be 0.
12	FIRF	Interrupt Request Flag 0: No interrupt request (Initial value) 1: An interrupt request present This bit is automatically cleared to 0 when an FE level interrupt request is accepted by the CPU core.
11 to 0	—	Reserved. The write value should always be 0.

7.3 Interrupt Sources

Interrupt sources are grouped into three: NMI, INTPx, and on-chip peripheral modules. Priority of each interrupt is represented by an interrupt priority level value between 0 (highest level) and 15 (lowest level).

7.3.1 NMI Interrupts

NMI interrupts are input from the Error Control Module. For details see Section 48, Error Control Module (ECM).

7.3.2 INTPx Interrupts

INTPx interrupts are input from INTPx pins. A method of detecting the INTPx interrupt can be selected from rising edge, falling edge, and both edges. For details on the detection method, see Section 2, Pins.

Furthermore, interrupt priority levels 0 to 15 can be set for each pin in the interrupt control register (EICn).

When an interrupt request is detected by a change in the INTPx pin, and an interrupt request signal is sent to the INTC. The INTPx interrupt request detection result is retained until the interrupt request is accepted. Whether an INTPx interrupt request has been detected or not can be checked by reading the EIRFn bit in the corresponding EI level interrupt control register n (EICn) for INTPx. Writing 0 clears the INTPx interrupt request detection result.

When exiting the INTPx interrupt exception handler, confirm that the EIRFn bit has been cleared in the corresponding EI level interrupt control register n (EICn) for INTPx to prevent re-acceptance by mistake, and then issue an instruction to return from interrupt.

7.3.3 On-Chip Peripheral Module Interrupts

For the on-chip peripheral module that generates an interrupt, see 7.4, Interrupt Exception Handler and Priority Operations.

Since different interrupt vectors are assigned to each source, the interrupt exception handler need not decide sources. Priority can be set within priority levels 0 to 15 for each interrupt source.

7.4 Interrupt Exception Handler and Priority Operations

The following tables list interrupt sources, source codes, exception handler offset addresses, and interrupt priority.

There are two specifications for exception handler addresses: (1) standard specifications where exception handler addresses are determined by the PSW.EBV bit in the CPU core, RBASE register, and EBASE register and (2) extended specifications where exception handler addresses for interrupts are specified individually for each channel.

In the standard specifications, an offset address is added to the base address (RBASE register /EBASE register) in the CPU core to generate an exception handler address. The following two methods are provided for giving an interrupt offset address. For channels other than the interrupt channel, the specified offset address is given.

- An offset address is determined within a range of +100H to +1F0H according to the priority level (0 to 15) specified for each channel, regardless of interrupt channels.
- Every offset address is +100H regardless of the priority level. This is a function to reduce the memory occupation size of the exception handler. To utilize this function, set the RINT bit in the RBASE/EBASE register to 1. The initial value of RINT is 0.

In the extended specifications, a table for reading exception handler addresses for each interrupt channel is provided. The handler address is extracted by referencing the table. The table reference position is obtained by the following calculation formula. The INTBP register exists in the CPU core.

Exception handler address read position = INTBP + channel number * 4-byte

For details of handler addresses, refer to "*RH850G3M User's Manual Software Edition*."

Priority levels 0 to 15 (0: highest) of INTPx (external interrupts) and on-chip peripheral module interrupts can be set for each channel. If specified priority levels are equal, an interrupt source of smaller channel number is selected as fixed priority.

NOTE

Be sure to place a SYNCP instruction at the start of handers for exceptions of set E.

- Exceptions of set E: FENMI, FEINT, EIINT (with the direct vector method), SYSERR, and FPI

For details, see "*RH850G3M User's Manual Software Edition*".

7.4.1 D1L1 Interrupt Exception Handler and Priority

Table 7.6 D1L1 Interrupt Exception Handler and Priority (1/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
Non-maskable interrupt			(FENMI)	$E0_H$	$+0E0_H$	$+0E0_H$	—		High
FE level interrupt			(FEINT)	$F0_H$	$+0F0_H$	$+0F0_H$	—		
WDTA0	75 % interrupt (INTWDTA0)		0	1000	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+000_H$	0 to 15 (15)	
WDTA1	75 % interrupt (INTWDTA1)		1	1001			$+004_H$	0 to 15 (15)	
RTCA0	1 second interrupt (INTRTCA01S)		2	1002			$+008_H$	0 to 15 (15)	
	Alarm interrupt (INTRTCA0AL)		3	1003			$+00C_H$	0 to 15 (15)	
	Fixed interval interrupt (INTRTCA0R)		4	1004			$+010_H$	0 to 15 (15)	
AWOT0	Timer interrupt (INTAWOT0)		5	1005			$+014_H$	0 to 15 (15)	
Port	External interrupt 0 (INTP0)		6	1006			$+018_H$	0 to 15 (15)	
	External interrupt 1 (INTP1)		7	1007			$+01C_H$	0 to 15 (15)	
	External interrupt 2 (INTP2)		8	1008			$+020_H$	0 to 15 (15)	
	External interrupt 3 (INTP3)		9	1009			$+024_H$	0 to 15 (15)	
	External interrupt 4 (INTP4)		10	100A			$+028_H$	0 to 15 (15)	
	External interrupt 5 (INTP5)		11	100B			$+02C_H$	0 to 15 (15)	
	External interrupt 6 (INTP6)		12	100C			$+030_H$	0 to 15 (15)	
	External interrupt 7 (INTP7)		13	100D			$+034_H$	0 to 15 (15)	
	External interrupt 8 (INTP8)		14	100E			$+038_H$	0 to 15 (15)	
	External interrupt 9 (INTP9)		15	100F			$+03C_H$	0 to 15 (15)	
	External interrupt 10 (INTP10)		16	1010			$+040_H$	0 to 15 (15)	
TAUB0	Channel 0 interrupt (INTTAUB0I0)		17	1011			$+044_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB0I1)		18	1012			$+048_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB0I2)		19	1013			$+04C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB0I3)		20	1014			$+050_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB0I4)		21	1015			$+054_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB0I5)		22	1016			$+058_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB0I6)		23	1017			$+05C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB0I7)		24	1018			$+060_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB0I8)		25	1019			$+064_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB0I9)		26	101A			$+068_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB0I10)		27	101B			$+06C_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB0I11)		28	101C			$+070_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB0I12)		29	101D			$+074_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB0I13)		30	101E			$+078_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB0I14)		31	101F			$+07C_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB0I15)		32	1020			$+080_H$	0 to 15 (15)	

Table 7.6 D1L1 Interrupt Exception Handler and Priority (2/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
TAUB1	Channel 0 interrupt (INTTAUB1I0)		33	1021	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+084_H$	0 to 15 (15)	High
	Channel 1 interrupt (INTTAUB1I1)		34	1022			$+088_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB1I2)		35	1023			$+08C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB1I3)		36	1024			$+090_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB1I4)		37	1025			$+094_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB1I5)		38	1026			$+098_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB1I6)		39	1027			$+09C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB1I7)		40	1028			$+0A0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB1I8)		41	1029			$+0A4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB1I9)		42	102A			$+0A8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB1I10)		43	102B			$+0AC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB1I11)		44	102C			$+0B0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB1I12)		45	102D			$+0B4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB1I13)		46	102E			$+0B8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB1I14)		47	102F			$+0BC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB1I15)		48	1030			$+0C0_H$	0 to 15 (15)	
TAUB2	Channel 0 interrupt (INTTAUB2I0)		49	1031			$+0C4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB2I1)		50	1032			$+0C8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB2I2)		51	1033			$+0CC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB2I3)		52	1034			$+0D0_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB2I4)		53	1035			$+0D4_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB2I5)		54	1036			$+0D8_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB2I6)		55	1037			$+0DC_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB2I7)		56	1038			$+0E0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB2I8)		57	1039			$+0E4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB2I9)		58	103A			$+0E8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB2I10)		59	103B			$+0EC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB2I11)		60	103C			$+0F0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB2I12)		61	103D			$+0F4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB2I13)		62	103E			$+0F8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB2I14)		63	103F			$+0FC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB2I15)		64	1040			$+100_H$	0 to 15 (15)	
ADCE0	Temperature Sensor interrupt (INTADCE0TSN)		65	1041			$+104_H$	0 to 15 (15)	
	Scan group 1 interrupt (INTADCE0I1)		66	1042			$+108_H$	0 to 15 (15)	
	Scan group 2 interrupt (INTADCE0I2)		67	1043			$+10C_H$	0 to 15 (15)	
	Scan group 3 interrupt (INTADCE0I3)		68	1044			$+110_H$	0 to 15 (15)	
RSCAN	Channel 0 to 2 global error interrupt (INTRCANGERR)	√	69	1045			$+114_H$	0 to 15 (15)	
	Channel 0 to 2 RX FIFO interrupt (INTRCANGRECC)	√	70	1046			$+118_H$	0 to 15 (15)	
	Channel 0 error interrupt (INTRCAN0ERR)	√	71	1047			$+11C_H$	0 to 15 (15)	
	Channel 0 COM RX FIFO interrupt (INTRCAN0REC)	√	72	1048			$+120_H$	0 to 15 (15)	
	Channel 0 TX interrupt (INTRCAN0TRX)	√	73	1049			$+124_H$	0 to 15 (15)	
CSIG0	Reception error interrupt (INTCSIGT0IRE)		74	104A			$+128_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG0IR)		75	104B			$+12C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG0IC)		76	104C			$+130_H$	0 to 15 (15)	
CSIH0	Reception error interrupt (INTCSIH0IRE)		77	104D			$+134_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH0IR)		78	104E			$+138_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH0IC)		79	104F			$+13C_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH0IJC)		80	1050			$+140_H$	0 to 15 (15)	

Table 7.6 D1L1 Interrupt Exception Handler and Priority (3/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RLIN30	Status interrupt (INTRLIN30UR2)		81	1051	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+144_H$	0 to 15 (15)	High
	Reception complete interrupt (INTRLIN30UR1)		82	1052			$+148_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN30UR0)		83	1053			$+14C_H$	0 to 15 (15)	
RLIN31	Status interrupt (INTRLIN31UR2)		84	1054			$+150_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN31UR1)		85	1055			$+154_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN31UR0)		86	1056			$+158_H$	0 to 15 (15)	
SG0	Threshold interrupt (INTSG0TI)		87	1057			$+15C_H$	0 to 15 (15)	
SG1	Threshold interrupt (INTSG1TI)		88	1058			$+160_H$	0 to 15 (15)	
SG2	Threshold interrupt (INTSG2TI)		89	1059			$+164_H$	0 to 15 15)	
SG3	Threshold interrupt (INTSG3TI)		90	105A			$+168_H$	0 to 15 (15)	
SG4	Threshold interrupt (INTSG4TI)		91	105B			$+16C_H$	0 to 15 (15)	
OCD/DCU	Debugger hot-attach interrupt (INTDCUTDI)	√	92	105C			$+170_H$	0 to 15 (15)	
DMAC	DMA Transfer Error (INTDMAERR)		93	105D			$+174_H$	0 to 15 (15)	
	DMA channel 0 transfer completion/transfer count compare match interrupt (INTDMA0)		94	105E			$+178_H$	0 to 15 (15)	
	DMA channel 1 transfer completion/transfer count compare match interrupt (INTDMA1)		95	105F			$+17C_H$	0 to 15 (15)	
	DMA channel 2 transfer completion/transfer count compare match interrupt (INTDMA2)		96	1060			$+180_H$	0 to 15 (15)	
	DMA channel 3 transfer completion/transfer count compare match interrupt (INTDMA3)		97	1061			$+184_H$	0 to 15 (15)	
	DMA channel 4 transfer completion/transfer count compare match interrupt (INTDMA4)		98	1062			$+188_H$	0 to 15 (15)	
	DMA channel 5 transfer completion/transfer count compare match interrupt (INTDMA5)		99	1063			$+18C_H$	0 to 15 (15)	
	DMA channel 6 transfer completion/transfer count compare match interrupt (INTDMA6)		100	1064			$+190_H$	0 to 15 (15)	
	DMA channel 7 transfer completion/transfer count compare match interrupt (INTDMA7)		101	1065			$+194_H$	0 to 15 (15)	
	DMA channel 8 transfer completion/transfer count compare match interrupt (INTDMA8)		102	1066			$+198_H$	0 to 15 (15)	
	DMA channel 9 transfer completion/transfer count compare match interrupt (INTDMA9)		103	1067			$+19C_H$	0 to 15 (15)	
	DMA channel 10 transfer completion/transfer count compare match interrupt (INTDMA10)		104	1068			$+1A0_H$	0 to 15 (15)	
	DMA channel 11 transfer completion/transfer count compare match interrupt (INTDMA11)		105	1069			$+1A4_H$	0 to 15 (15)	
	DMA channel 12 transfer completion/transfer count compare match interrupt (INTDMA12)		106	106A			$+1A8_H$	0 to 15 (15)	
	DMA channel 13 transfer completion/transfer count compare match interrupt (INTDMA13)		107	106B			$+1AC_H$	0 to 15 (15)	
	DMA channel 14 transfer completion/transfer count compare match interrupt (INTDMA14)		108	106C			$+1B0_H$	0 to 15 (15)	
	DMA channel 15 transfer completion/transfer count compare match interrupt (INTDMA15)		109	106D			$+1B4_H$	0 to 15 (15)	
RIIC0	Receive error/event interrupt (INTRIIC0LEE)	√	110	106E			$+1B8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC0RI)		111	106F			$+1BC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC0TI)		112	1070			$+1C0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC0TEI)	√	113	1071			$+1C4_H$	0 to 15 (15)	
RIIC1	Receive error/event interrupt (INTRIIC1LEE)	√	114	1072			$+1C8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC1RI)		115	1073			$+1CC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC1TI)		116	1074			$+1D0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC1TEI)	√	117	1075			$+1D4_H$	0 to 15 (15)	

Table 7.6 D1L1 Interrupt Exception Handler and Priority (4/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RSCAN1	Channel 1 error interrupt (INTRCAN1ERR)	√	118	1076	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+1D8_H$	0 to 15 (15)	High
	Channel 1 COM RX FIFO interrupt (INTRCAN1REC)	√	119	1077			$+1DC_H$	0 to 15 (15)	
	Channel 1 TX interrupt (INTRCAN1TRX)	√	120	1078			$+1E0_H$	0 to 15 (15)	
TAUJ0	Channel 0 interrupt (INTTAUJ0I0)		121	1079			$+1E4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUJ0I1)		122	107A			$+1E8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUJ0I2)		123	107B			$+1EC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUJ0I3)		124	107C			$+1F0_H$	0 to 15 (15)	
OSTM0	Timer interrupt (INTOSTM0)		125	107D			$+1F4_H$	0 to 15 (15)	
OSTM1	Timer interrupt (INTOSTM1)		126	107E			$+1F8_H$	0 to 15 (15)	
CSIG1	Reception error interrupt (INTCSIGT1IRE)		127	107F			$+1FC_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG1IR)		128	1080			$+200_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG1IC)		129	1081			$+204_H$	0 to 15 15)	
CSIG2	Reception error interrupt (INTCSIGT2IRE)		130	1082			$+208_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG2IR)		131	1083			$+20C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG2IC)		132	1084			$+210_H$	0 to 15 (15)	
CSIG3	Reception error interrupt (INTCSIGT3IRE)		133	1085			$+214_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG3IR)		134	1086			$+218_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG3IC)		135	1087			$+21C_H$	0 to 15 (15)	
CSIH1	Reception error interrupt (INTCSIH1IRE)		136	1088			$+220_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH1IR)		137	1089			$+224_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH1IC)		138	108A			$+228_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH1IJC)		139	108B			$+22C_H$	0 to 15 (15)	

Table 7.6 D1L1 Interrupt Exception Handler and Priority (5/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
PWM Generators and Diagnostic	A/D Converter trigger queue full interrupt (INTQFULL)		140	108C	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+230_H$	0 to 15 (15)	High
	PWM Generator channel 0 interrupt (INTPWGA0)		141	108D			$+234_H$	0 to 15 (15)	
	PWM Generator channel 1 interrupt (INTPWGA1)		142	108E			$+238_H$	0 to 15 (15)	
	PWM Generator channel 2 interrupt (INTPWGA2)		143	108F			$+23C_H$	0 to 15 (15)	
	PWM Generator channel 3 interrupt (INTPWGA3)		144	1090			$+240_H$	0 to 15 (15)	
	PWM Generator channel 4 interrupt (INTPWGA4)		145	1091			$+244_H$	0 to 15 (15)	
	PWM Generator channel 5 interrupt (INTPWGA5)		146	1092			$+248_H$	0 to 15 (15)	
	PWM Generator channel 6 interrupt (INTPWGA6)		147	1093			$+24C_H$	0 to 15 (15)	
	PWM Generator channel 7 interrupt (INTPWGA7)		148	1094			$+250_H$	0 to 15 (15)	
	PWM Generator channel 8 interrupt (INTPWGA8)		149	1095			$+254_H$	0 to 15 (15)	
	PWM Generator channel 9 interrupt (INTPWGA9)		150	1096			$+258_H$	0 to 15 (15)	
	PWM Generator channel 10 interrupt (INTPWGA10)		151	1097			$+25C_H$	0 to 15 (15)	
	PWM Generator channel 11 interrupt (INTPWGA11)		152	1098			$+260_H$	0 to 15 (15)	
	PWM Generator channel 12 interrupt (INTPWGA12)		153	1099			$+264_H$	0 to 15 (15)	
	PWM Generator channel 13 interrupt (INTPWGA13)		154	109A			$+268_H$	0 to 15 (15)	
	PWM Generator channel 14 interrupt (INTPWGA14)		155	109B			$+26C_H$	0 to 15 (15)	
	PWM Generator channel 15 interrupt (INTPWGA15)		156	109C			$+270_H$	0 to 15 (15)	
	PWM Generator channel 16 interrupt (INTPWGA16)		157	109D			$+274_H$	0 to 15 (15)	
	PWM Generator channel 17 interrupt (INTPWGA17)		158	109E			$+278_H$	0 to 15 (15)	
	PWM Generator channel 18 interrupt (INTPWGA18)		159	109F			$+27C_H$	0 to 15 (15)	
	PWM Generator channel 19 interrupt (INTPWGA19)		160	10A0			$+280_H$	0 to 15 (15)	
	PWM Generator channel 20 interrupt (INTPWGA20)		161	10A1			$+284_H$	0 to 15 (15)	
	PWM Generator channel 21 interrupt (INTPWGA21)		162	10A2			$+288_H$	0 to 15 (15)	
	PWM Generator channel 22 interrupt (INTPWGA22)		163	10A3			$+28C_H$	0 to 15 (15)	
	PWM Generator channel 23 interrupt (INTPWGA23)		164	10A4			$+290_H$	0 to 15 (15)	
ICU-S2	CMD registers ready to write interrupt (INTICUSTWRDY)		165	10A5			$+294_H$	0 to 15 (15)	
	CMD registers ready to read interrupt (INTICUSTRRDY)		166	10A6			$+298_H$	0 to 15 (15)	
–	Reserved	√	167	10A7			$+29C_H$	0 to 15 (15)	
–	Reserved		168	10A8			$+2A0_H$	0 to 15 (15)	
–	Reserved		169	10A9			$+2A4_H$	0 to 15 (15)	
–	Reserved	√	170	10AA			$+2A8_H$	0 to 15 (15)	
Flash control	Flash access error (INTFLERR)	√	171	10AB			$+2AC_H$	0 to 15 (15)	
Flash control	Flash sequencer ready (INTFLENDNM)		172	10AC			$+2B0_H$	0 to 15 (15)	
–	Reserved	√	173	10AD			$+2B4_H$	0 to 15 (15)	
LCBI0	Read data ready interrupt (INTLCBI0RDY)		174	10AE			$+2B8_H$	0 to 15 (15)	
	Write buffer empty interrupt (INTLCBI0EMPTY)		175	10AF			$+2BC_H$	0 to 15 (15)	
	Write buffer half full interrupt (INTLCBI0HALF)		176	10B0			$+2C0_H$	0 to 15 (15)	
	Write buffer full interrupt (INTLCBI0FULL)		177	10B1			$+2C4_H$	0 to 15 (15)	
	Write buffer quarter full interrupt (INTLCBI0QTR)		178	10B2			$+2C8_H$	0 to 15 (15)	
	Write buffer three quarters full interrupt (INTLCBI03QTR)		179	10B3			$+2CC_H$	0 to 15 (15)	
RSCAN2	Channel 2 error interrupt (INTRCAN2ERR)	√	180	10B4			$+2D0_H$	0 to 15 (15)	
	Channel 2 COM RX FIFO interrupt (INTRCAN2REC)	√	181	10B5			$+2D4_H$	0 to 15 (15)	
	Channel 2 TX interrupt (INTRCAN2TRX)	√	182	10B6			$+2D8_H$	0 to 15 (15)	
PCMP0	FIFO buffer fill interrupt (INTPCMP0FFIL)		183	10B7			$+2DC_H$	0 to 15 (15)	
	Error interrupt (INTPCMP0FERR)		184	10B8			$+2E0_H$	0 to 15 (15)	

Table 7.6 D1L1 Interrupt Exception Handler and Priority (6/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
ISM0	Target position reached interrupt (INTISM0REACHED)		185	10B9	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+2E4_H$	0 to 15 (15)	High
	Idle state interrupt (INTISM0DONE)		186	10BA			$+2E8_H$	0 to 15 (15)	
	ZPD measurement started interrupt (INTISM0ZPDAD)		187	10BB			$+2EC_H$	0 to 15 (15)	
–	Reserved	√	188	10BC			$+2F0_H$	0 to 15 (15)	
SSIF0	Multi-purpose interrupt (INTSSIF0)	√	189	10BD			$+2F4_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF0RX)	√	190	10BE			$+2F8_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF0TX)	√	191	10BF			$+2FC_H$	0 to 15 (15)	
SSIF1	Multi-purpose interrupt (INTSSIF1)	√	192	10C0			$+300_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF1RX)	√	193	10C1			$+304_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF1TX)	√	194	10C2			$+308_H$	0 to 15 (15)	
–	Reserved	√	195	10C3			$+30C_H$	0 to 15 (15)	
–	Reserved	√	196	10C4			$+310_H$	0 to 15 (15)	
–	Reserved	√	197	10C5			$+314_H$	0 to 15 (15)	
–	Reserved	√	198	10C6			$+318_H$	0 to 15 (15)	
–	Reserved	√	199	10C7			$+31C_H$	0 to 15 (15)	
–	Reserved		200	10C8			$+320_H$	0 to 15 (15)	
–	Reserved	√	201	10C9			$+324_H$	0 to 15 (15)	
–	Reserved	√	202	10CA			$+328_H$	0 to 15 (15)	
–	Reserved	√	203	10CB			$+32C_H$	0 to 15 (15)	
–	Reserved	√	204	10CC			$+330_H$	0 to 15 (15)	
–	Reserved	√	205	10CD			$+334_H$	0 to 15 (15)	
–	Reserved	√	206	10CE			$+338_H$	0 to 15 (15)	
–	Reserved	√	207	10CF			$+33C_H$	0 to 15 (15)	
–	Reserved	√	208	10D0			$+340_H$	0 to 15 (15)	
–	Reserved	√	209	10D1			$+344_H$	0 to 15 (15)	
–	Reserved	√	210	10D2			$+348_H$	0 to 15 (15)	
–	Reserved	√	211	10D3			$+34C_H$	0 to 15 (15)	
–	Reserved	√	212	10D4			$+350_H$	0 to 15 (15)	
–	Reserved	√	213	10D5			$+354_H$	0 to 15 (15)	
–	Reserved	√	214	10D6			$+358_H$	0 to 15 (15)	
RLIN32	Status interrupt interrupt (INTRLIN32UR2)		215	10D7			$+35C_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN32UR1)		216	10D8			$+360_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN32UR0)		217	10D9			$+364_H$	0 to 15 (15)	
RLIN33	Status interrupt (INTRLIN33UR2)		218	10DA			$+368_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN33UR1)		219	10DB			$+36C_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN33UR0)		220	10DC			$+370_H$	0 to 15 (15)	
–	Reserved	√	221	10DD			$+374_H$	0 to 15 (15)	
–	Reserved		222	10DE			$+378_H$	0 to 15 (15)	
–	Reserved	√	223	10DF			$+37C_H$	0 to 15 (15)	
–	Reserved	√	224	10E0			$+380_H$	0 to 15 (15)	
–	Reserved	√	225	10E1			$+384_H$	0 to 15 (15)	
–	Reserved	√	226	10E2			$+388_H$	0 to 15 (15)	
–	Reserved	√	227	10E3			$+38C_H$	0 to 15 (15)	
–	Reserved	√	228	10E4			$+390_H$	0 to 15 (15)	
–	Reserved	√	229	10E5			$+394_H$	0 to 15 (15)	
–	Reserved	√	230	10E6			$+398_H$	0 to 15 (15)	
–	Reserved	√	231	10E7			$+39C_H$	0 to 15 (15)	

Table 7.6 D1L1 Interrupt Exception Handler and Priority (7/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
–	Reserved	√	232	10E8	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+3A0_H$	0 to 15 (15)	High
–	Reserved	√	233	10E9			$+3A4_H$	0 to 15 (15)	
–	Reserved	√	234	10EA			$+3A8_H$	0 to 15 (15)	
–	Reserved	√	235	10EB			$+3AC_H$	0 to 15 (15)	
–	Reserved		236	10EC			$+3B0_H$	0 to 15 (15)	
–	Reserved		237	10ED			$+3B4_H$	0 to 15 (15)	
–	Reserved		238	10EE			$+3B8_H$	0 to 15 (15)	
–	Reserved	√	239	10EF			$+3BC_H$	0 to 15 (15)	
–	Reserved	√	240	10F0			$+3C0_H$	0 to 15 (15)	
–	Reserved	√	241	10F1			$+3C4_H$	0 to 15 (15)	
–	Reserved	√	242	10F2			$+3C8_H$	0 to 15 (15)	
–	Reserved		243	10F3			$+3CC_H$	0 to 15 (15)	
–	Reserved	√	244	10F4			$+3D0_H$	0 to 15 (15)	
–	Reserved	√	245	10F5			$+3D4_H$	0 to 15 (15)	
–	Reserved	√	246	10F6			$+3D8_H$	0 to 15 (15)	
–	Reserved	√	247	10F7			$+3DC_H$	0 to 15 (15)	
–	Reserved		248	10F8			$+3E0_H$	0 to 15 (15)	
–	Reserved		249	10F9			$+3E4_H$	0 to 15 (15)	
–	Reserved		250	10FA			$+3E8_H$	0 to 15 (15)	
–	Reserved		251	10FB			$+3EC_H$	0 to 15 (15)	
–	Reserved	√	252	10FC			$+3F0_H$	0 to 15 (15)	
–	Reserved		253	10FD			$+3F4_H$	0 to 15 (15)	
–	Reserved	√	254	10FE			$+3F8_H$	0 to 15 (15)	
Internal bus errors	This interrupt is a logical OR combination of various internal bus error indications. The exact cause of this error interrupt can be evaluated by use of the bus error status registers. Refer to Section 7.4.9, Bus error status registers for details.		255	10FF			$+3FC_H$	0 to 15 (15)	

Note 1. For a level interrupt, the status register in each module should be cleared during interrupt processing by software. The EICTn bit in EICn is set to 1. The EIRFn bit in EICn cannot be cleared by software.

7.4.2 D1L2(H) Interrupt Exception Handler and Priority

Table 7.7 D1L2(H) Interrupt Exception Handler and Priority (1/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
Non-maskable interrupt			(FENMI)	$E0_H$	$+0E0_H$	$+0E0_H$	—		High
FE level interrupt			(FEINT)	$F0_H$	$+0F0_H$	$+0F0_H$	—		
WDTA0	75 % interrupt (INTWDTA0)		0	1000	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+000_H$	0 to 15 (15)	
WDTA1	75 % interrupt (INTWDTA1)		1	1001			$+004_H$	0 to 15 (15)	
RTCA0	1 second interrupt (INTRTCA01S)		2	1002			$+008_H$	0 to 15 (15)	
	Alarm interrupt (INTRTCA0AL)		3	1003			$+00C_H$	0 to 15 (15)	
	Fixed interval interrupt (INTRTCA0R)		4	1004			$+010_H$	0 to 15 (15)	
AWOT0	Timer interrupt (INTAWOT0)		5	1005			$+014_H$	0 to 15 (15)	
Port	External interrupt 0 (INTP0)		6	1006			$+018_H$	0 to 15 (15)	
	External interrupt 1 (INTP1)		7	1007			$+01C_H$	0 to 15 (15)	
	External interrupt 2 (INTP2)		8	1008			$+020_H$	0 to 15 (15)	
	External interrupt 3 (INTP3)		9	1009			$+024_H$	0 to 15 (15)	
	External interrupt 4 (INTP4)		10	100A			$+028_H$	0 to 15 (15)	
	External interrupt 5 (INTP5)		11	100B			$+02C_H$	0 to 15 (15)	
	External interrupt 6 (INTP6)		12	100C			$+030_H$	0 to 15 (15)	
	External interrupt 7 (INTP7)		13	100D			$+034_H$	0 to 15 (15)	
	External interrupt 8 (INTP8)		14	100E			$+038_H$	0 to 15 (15)	
	External interrupt 9 (INTP9)		15	100F			$+03C_H$	0 to 15 (15)	
	External interrupt 10 (INTP10)		16	1010			$+040_H$	0 to 15 (15)	
TAUB0	Channel 0 interrupt (INTTAUB0I0)		17	1011			$+044_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB0I1)		18	1012			$+048_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB0I2)		19	1013			$+04C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB0I3)		20	1014			$+050_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB0I4)		21	1015			$+054_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB0I5)		22	1016			$+058_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB0I6)		23	1017			$+05C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB0I7)		24	1018			$+060_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB0I8)		25	1019			$+064_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB0I9)		26	101A			$+068_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB0I10)		27	101B			$+06C_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB0I11)		28	101C			$+070_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB0I12)		29	101D			$+074_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB0I13)		30	101E			$+078_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB0I14)		31	101F			$+07C_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB0I15)		32	1020			$+080_H$	0 to 15 (15)	

Table 7.7 D1L2(H) Interrupt Exception Handler and Priority (2/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
TAUB1	Channel 0 interrupt (INTTAUB1I0)		33	1021	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+084_H$	0 to 15 (15)	High
	Channel 1 interrupt (INTTAUB1I1)		34	1022			$+088_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB1I2)		35	1023			$+08C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB1I3)		36	1024			$+090_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB1I4)		37	1025			$+094_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB1I5)		38	1026			$+098_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB1I6)		39	1027			$+09C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB1I7)		40	1028			$+0A0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB1I8)		41	1029			$+0A4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB1I9)		42	102A			$+0A8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB1I10)		43	102B			$+0AC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB1I11)		44	102C			$+0B0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB1I12)		45	102D			$+0B4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB1I13)		46	102E			$+0B8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB1I14)		47	102F			$+0BC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB1I15)		48	1030			$+0C0_H$	0 to 15 (15)	
TAUB2	Channel 0 interrupt (INTTAUB2I0)		49	1031			$+0C4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB2I1)		50	1032			$+0C8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB2I2)		51	1033			$+0CC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB2I3)		52	1034			$+0D0_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB2I4)		53	1035			$+0D4_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB2I5)		54	1036			$+0D8_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB2I6)		55	1037			$+0DC_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB2I7)		56	1038			$+0E0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB2I8)		57	1039			$+0E4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB2I9)		58	103A			$+0E8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB2I10)		59	103B			$+0EC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB2I11)		60	103C			$+0F0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB2I12)		61	103D			$+0F4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB2I13)		62	103E			$+0F8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB2I14)		63	103F			$+0FC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB2I15)		64	1040			$+100_H$	0 to 15 (15)	
ADCE0	Temperature Sensor interrupt (INTADCE0TSN)		65	1041			$+104_H$	0 to 15 (15)	
	Scan group 1 interrupt (INTADCE0I1)		66	1042			$+108_H$	0 to 15 (15)	
	Scan group 2 interrupt (INTADCE0I2)		67	1043			$+10C_H$	0 to 15 (15)	
	Scan group 3 interrupt (INTADCE0I3)		68	1044			$+110_H$	0 to 15 (15)	
RSCAN	Channel 0 to 2 global error interrupt (INTRCANGERR)	√	69	1045			$+114_H$	0 to 15 (15)	
	Channel 0 to 2 RX FIFO interrupt (INTRCANGRECC)	√	70	1046			$+118_H$	0 to 15 (15)	
	Channel 0 error interrupt (INTRCAN0ERR)	√	71	1047			$+11C_H$	0 to 15 (15)	
	Channel 0 COM RX FIFO interrupt (INTRCAN0REC)	√	72	1048			$+120_H$	0 to 15 (15)	
	Channel 0 TX interrupt (INTRCAN0TRX)	√	73	1049			$+124_H$	0 to 15 (15)	
CSIG0	Reception error interrupt (INTCSIGT0IRE)		74	104A			$+128_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG0IR)		75	104B			$+12C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG0IC)		76	104C			$+130_H$	0 to 15 (15)	
CSIH0	Reception error interrupt (INTCSIH0IRE)		77	104D			$+134_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH0IR)		78	104E			$+138_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH0IC)		79	104F			$+13C_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH0IJC)		80	1050			$+140_H$	0 to 15 (15)	

Table 7.7 D1L2(H) Interrupt Exception Handler and Priority (3/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RLIN30	Status interrupt (INTRLIN30UR2)		81	1051	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+144_H$	0 to 15 (15)	High
	Reception complete interrupt (INTRLIN30UR1)		82	1052			$+148_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN30UR0)		83	1053			$+14C_H$	0 to 15 (15)	
RLIN31	Status interrupt (INTRLIN31UR2)		84	1054			$+150_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN31UR1)		85	1055			$+154_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN31UR0)		86	1056			$+158_H$	0 to 15 (15)	
SG0	Threshold interrupt (INTSG0TI)		87	1057			$+15C_H$	0 to 15 (15)	
SG1	Threshold interrupt (INTSG1TI)		88	1058			$+160_H$	0 to 15 (15)	
SG2	Threshold interrupt (INTSG2TI)		89	1059			$+164_H$	0 to 15 15)	
SG3	Threshold interrupt (INTSG3TI)		90	105A			$+168_H$	0 to 15 (15)	
SG4	Threshold interrupt (INTSG4TI)		91	105B			$+16C_H$	0 to 15 (15)	
OCD/DCU	Debugger hot-attach interrupt (INTDCUTDI)	√	92	105C			$+170_H$	0 to 15 (15)	
DMAC	DMA Transfer Error (INTDMAERR)		93	105D			$+174_H$	0 to 15 (15)	
	DMA channel 0 transfer completion/transfer count compare match interrupt (INTDMA0)		94	105E			$+178_H$	0 to 15 (15)	
	DMA channel 1 transfer completion/transfer count compare match interrupt (INTDMA1)		95	105F			$+17C_H$	0 to 15 (15)	
	DMA channel 2 transfer completion/transfer count compare match interrupt (INTDMA2)		96	1060			$+180_H$	0 to 15 (15)	
	DMA channel 3 transfer completion/transfer count compare match interrupt (INTDMA3)		97	1061			$+184_H$	0 to 15 (15)	
	DMA channel 4 transfer completion/transfer count compare match interrupt (INTDMA4)		98	1062			$+188_H$	0 to 15 (15)	
	DMA channel 5 transfer completion/transfer count compare match interrupt (INTDMA5)		99	1063			$+18C_H$	0 to 15 (15)	
	DMA channel 6 transfer completion/transfer count compare match interrupt (INTDMA6)		100	1064			$+190_H$	0 to 15 (15)	
	DMA channel 7 transfer completion/transfer count compare match interrupt (INTDMA7)		101	1065			$+194_H$	0 to 15 (15)	
	DMA channel 8 transfer completion/transfer count compare match interrupt (INTDMA8)		102	1066			$+198_H$	0 to 15 (15)	
	DMA channel 9 transfer completion/transfer count compare match interrupt (INTDMA9)		103	1067			$+19C_H$	0 to 15 (15)	
	DMA channel 10 transfer completion/transfer count compare match interrupt (INTDMA10)		104	1068			$+1A0_H$	0 to 15 (15)	
	DMA channel 11 transfer completion/transfer count compare match interrupt (INTDMA11)		105	1069			$+1A4_H$	0 to 15 (15)	
	DMA channel 12 transfer completion/transfer count compare match interrupt (INTDMA12)		106	106A			$+1A8_H$	0 to 15 (15)	
	DMA channel 13 transfer completion/transfer count compare match interrupt (INTDMA13)		107	106B			$+1AC_H$	0 to 15 (15)	
	DMA channel 14 transfer completion/transfer count compare match interrupt (INTDMA14)		108	106C			$+1B0_H$	0 to 15 (15)	
	DMA channel 15 transfer completion/transfer count compare match interrupt (INTDMA15)		109	106D			$+1B4_H$	0 to 15 (15)	
RIIC0	Receive error/event interrupt (INTRIIC0LEE)	√	110	106E			$+1B8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC0RI)		111	106F			$+1BC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC0TI)		112	1070			$+1C0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC0TEI)	√	113	1071			$+1C4_H$	0 to 15 (15)	
RIIC1	Receive error/event interrupt (INTRIIC1LEE)	√	114	1072			$+1C8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC1RI)		115	1073			$+1CC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC1TI)		116	1074			$+1D0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC1TEI)	√	117	1075			$+1D4_H$	0 to 15 (15)	

Table 7.7 D1L2(H) Interrupt Exception Handler and Priority (4/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RSCAN1	Channel 1 error interrupt (INTRCAN1ERR)	√	118	1076	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+1D8_H$	0 to 15 (15)	High
	Channel 1 COM RX FIFO interrupt (INTRCAN1REC)	√	119	1077			$+1DC_H$	0 to 15 (15)	
	Channel 1 TX interrupt (INTRCAN1TRX)	√	120	1078			$+1E0_H$	0 to 15 (15)	
TAUJ0	Channel 0 interrupt (INTTAUJ0I0)		121	1079			$+1E4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUJ0I1)		122	107A			$+1E8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUJ0I2)		123	107B			$+1EC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUJ0I3)		124	107C			$+1F0_H$	0 to 15 (15)	
OSTM0	Timer interrupt (INTOSTM0)		125	107D			$+1F4_H$	0 to 15 (15)	
OSTM1	Timer interrupt (INTOSTM1)		126	107E			$+1F8_H$	0 to 15 (15)	
CSIG1	Reception error interrupt (INTCSIGT1IRE)		127	107F			$+1FC_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG1IR)		128	1080			$+200_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG1IC)		129	1081			$+204_H$	0 to 15 15)	
CSIG2	Reception error interrupt (INTCSIGT2IRE)		130	1082			$+208_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG2IR)		131	1083			$+20C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG2IC)		132	1084			$+210_H$	0 to 15 (15)	
CSIG3	Reception error interrupt (INTCSIGT3IRE)		133	1085			$+214_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG3IR)		134	1086			$+218_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG3IC)		135	1087			$+21C_H$	0 to 15 (15)	
CSIH1	Reception error interrupt (INTCSIH1IRE)		136	1088			$+220_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH1IR)		137	1089			$+224_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH1IC)		138	108A			$+228_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH1IJC)		139	108B			$+22C_H$	0 to 15 (15)	

Table 7.7 D1L2(H) Interrupt Exception Handler and Priority (5/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
PWM Generators and Diagnostic	A/D Converter trigger queue full interrupt (INTQFULL)		140	108C	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+230_H$	0 to 15 (15)	High
	PWM Generator channel 0 interrupt (INTPWGA0)		141	108D			$+234_H$	0 to 15 (15)	
	PWM Generator channel 1 interrupt (INTPWGA1)		142	108E			$+238_H$	0 to 15 (15)	
	PWM Generator channel 2 interrupt (INTPWGA2)		143	108F			$+23C_H$	0 to 15 (15)	
	PWM Generator channel 3 interrupt (INTPWGA3)		144	1090			$+240_H$	0 to 15 (15)	
	PWM Generator channel 4 interrupt (INTPWGA4)		145	1091			$+244_H$	0 to 15 (15)	
	PWM Generator channel 5 interrupt (INTPWGA5)		146	1092			$+248_H$	0 to 15 (15)	
	PWM Generator channel 6 interrupt (INTPWGA6)		147	1093			$+24C_H$	0 to 15 (15)	
	PWM Generator channel 7 interrupt (INTPWGA7)		148	1094			$+250_H$	0 to 15 (15)	
	PWM Generator channel 8 interrupt (INTPWGA8)		149	1095			$+254_H$	0 to 15 (15)	
	PWM Generator channel 9 interrupt (INTPWGA9)		150	1096			$+258_H$	0 to 15 (15)	
	PWM Generator channel 10 interrupt (INTPWGA10)		151	1097			$+25C_H$	0 to 15 (15)	
	PWM Generator channel 11 interrupt (INTPWGA11)		152	1098			$+260_H$	0 to 15 (15)	
	PWM Generator channel 12 interrupt (INTPWGA12)		153	1099			$+264_H$	0 to 15 (15)	
	PWM Generator channel 13 interrupt (INTPWGA13)		154	109A			$+268_H$	0 to 15 (15)	
	PWM Generator channel 14 interrupt (INTPWGA14)		155	109B			$+26C_H$	0 to 15 (15)	
	PWM Generator channel 15 interrupt (INTPWGA15)		156	109C			$+270_H$	0 to 15 (15)	
	PWM Generator channel 16 interrupt (INTPWGA16)		157	109D			$+274_H$	0 to 15 (15)	
	PWM Generator channel 17 interrupt (INTPWGA17)		158	109E			$+278_H$	0 to 15 (15)	
	PWM Generator channel 18 interrupt (INTPWGA18)		159	109F			$+27C_H$	0 to 15 (15)	
	PWM Generator channel 19 interrupt (INTPWGA19)		160	10A0			$+280_H$	0 to 15 (15)	
	PWM Generator channel 20 interrupt (INTPWGA20)		161	10A1			$+284_H$	0 to 15 (15)	
	PWM Generator channel 21 interrupt (INTPWGA21)		162	10A2			$+288_H$	0 to 15 (15)	
	PWM Generator channel 22 interrupt (INTPWGA22)		163	10A3			$+28C_H$	0 to 15 (15)	
	PWM Generator channel 23 interrupt (INTPWGA23)		164	10A4			$+290_H$	0 to 15 (15)	
ICU-S2	CMD registers ready to write interrupt (INTICUSTWRDY)		165	10A5			$+294_H$	0 to 15 (15)	
	CMD registers ready to read interrupt (INTICUSTRRDY)		166	10A6			$+298_H$	0 to 15 (15)	
–	Reserved	√	167	10A7			$+29C_H$	0 to 15 (15)	
–	Reserved		168	10A8			$+2A0_H$	0 to 15 (15)	
–	Reserved		169	10A9			$+2A4_H$	0 to 15 (15)	
–	Reserved	√	170	10AA			$+2A8_H$	0 to 15 (15)	
Flash control	Flash access error (INTFLERR)	√	171	10AB			$+2AC_H$	0 to 15 (15)	
Flash control	Flash sequencer ready (INTFLENDNM)		172	10AC			$+2B0_H$	0 to 15 (15)	
–	Reserved	√	173	10AD			$+2B4_H$	0 to 15 (15)	
LCBI0	Read data ready interrupt (INTLCBI0RDY)		174	10AE			$+2B8_H$	0 to 15 (15)	
	Write buffer empty interrupt (INTLCBI0EMPTY)		175	10AF			$+2BC_H$	0 to 15 (15)	
	Write buffer half full interrupt (INTLCBI0HALF)		176	10B0			$+2C0_H$	0 to 15 (15)	
	Write buffer full interrupt (INTLCBI0FULL)		177	10B1			$+2C4_H$	0 to 15 (15)	
	Write buffer quarter full interrupt (INTLCBI0QTR)		178	10B2			$+2C8_H$	0 to 15 (15)	
	Write buffer three quarters full interrupt (INTLCBI03QTR)		179	10B3			$+2CC_H$	0 to 15 (15)	
RSCAN2	Channel 2 error interrupt (INTRCAN2ERR)	√	180	10B4			$+2D0_H$	0 to 15 (15)	
	Channel 2 COM RX FIFO interrupt (INTRCAN2REC)	√	181	10B5			$+2D4_H$	0 to 15 (15)	
	Channel 2 TX interrupt (INTRCAN2TRX)	√	182	10B6			$+2D8_H$	0 to 15 (15)	
PCMP0	FIFO buffer fill interrupt (INTPCMP0FFIL)		183	10B7			$+2DC_H$	0 to 15 (15)	
	Error interrupt (INTPCMP0FERR)		184	10B8			$+2E0_H$	0 to 15 (15)	

Table 7.7 D1L2(H) Interrupt Exception Handler and Priority (6/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
ISM0	Target position reached interrupt (INTISM0REACHED)		185	10B9	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+2E4_H$	0 to 15 (15)	High
	Idle state interrupt (INTISM0DONE)		186	10BA			$+2E8_H$	0 to 15 (15)	
	ZPD measurement started interrupt (INTISM0ZPDAD)		187	10BB			$+2EC_H$	0 to 15 (15)	
–	Reserved	√	188	10BC			$+2F0_H$	0 to 15 (15)	
SSIF0	Multi-purpose interrupt (INTSSIF0)	√	189	10BD			$+2F4_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF0RX)	√	190	10BE			$+2F8_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF0TX)	√	191	10BF			$+2FC_H$	0 to 15 (15)	
SSIF1	Multi-purpose interrupt (INTSSIF1)	√	192	10C0			$+300_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF1RX)	√	193	10C1			$+304_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF1TX)	√	194	10C2			$+308_H$	0 to 15 (15)	
–	Reserved	√	195	10C3			$+30C_H$	0 to 15 (15)	
–	Reserved	√	196	10C4			$+310_H$	0 to 15 (15)	
–	Reserved	√	197	10C5			$+314_H$	0 to 15 (15)	
–	Reserved	√	198	10C6			$+318_H$	0 to 15 (15)	
VDCE0	Error interrupt (INTVDCE0ERR)	√	199	10C7			$+31C_H$	0 to 15 (15)	
	Reserved		200	10C8			$+320_H$	0 to 15 (15)	
	Reserved	√	201	10C9			$+324_H$	0 to 15 (15)	
	VSYNC output at Scaler 0 interrupt (INTVDCE0S0LOVSYNC)	√	202	10CA			$+328_H$	0 to 15 (15)	
	Line detection of designated line at Graphics3 interrupt (INTVDCE0GR3VLINE)	√	203	10CB			$+32C_H$	0 to 15 (15)	
	Reserved	√	204	10CC			$+330_H$	0 to 15 (15)	
	Reserved	√	205	10CD			$+334_H$	0 to 15 (15)	
	VSYNC output at Scaler 1 interrupt (INTVDCE0S1LOVSYNC)	√	206	10CE			$+338_H$	0 to 15 (15)	
	Reserved	√	207	10CF			$+33C_H$	0 to 15 (15)	
	Reserved	√	208	10D0			$+340_H$	0 to 15 (15)	
	Reserved	√	209	10D1			$+344_H$	0 to 15 (15)	
	Reserved	√	210	10D2			$+348_H$	0 to 15 (15)	
–	Reserved	√	211	10D3			$+34C_H$	0 to 15 (15)	
–	Reserved	√	212	10D4			$+350_H$	0 to 15 (15)	
–	Reserved	√	213	10D5			$+354_H$	0 to 15 (15)	
–	Reserved	√	214	10D6			$+358_H$	0 to 15 (15)	
RLIN32	Status interrupt interrupt (INTRLIN32UR2)		215	10D7			$+35C_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN32UR1)		216	10D8			$+360_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN32UR0)		217	10D9			$+364_H$	0 to 15 (15)	
RLIN33	Status interrupt (INTRLIN33UR2)		218	10DA			$+368_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN33UR1)		219	10DB			$+36C_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN33UR0)		220	10DC			$+370_H$	0 to 15 (15)	

Table 7.7 D1L2(H) Interrupt Exception Handler and Priority (7/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
–	Reserved	√	221	10DD	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+374_H$	0 to 15 (15)	High
–	Reserved		222	10DE			$+378_H$	0 to 15 (15)	
–	Reserved	√	223	10DF			$+37C_H$	0 to 15 (15)	
–	Reserved	√	224	10E0			$+380_H$	0 to 15 (15)	
–	Reserved	√	225	10E1			$+384_H$	0 to 15 (15)	
–	Reserved	√	226	10E2			$+388_H$	0 to 15 (15)	
–	Reserved	√	227	10E3			$+38C_H$	0 to 15 (15)	
–	Reserved	√	228	10E4			$+390_H$	0 to 15 (15)	
–	Reserved	√	229	10E5			$+394_H$	0 to 15 (15)	
–	Reserved	√	230	10E6			$+398_H$	0 to 15 (15)	
–	Reserved	√	231	10E7			$+39C_H$	0 to 15 (15)	
–	Reserved	√	232	10E8			$+3A0_H$	0 to 15 (15)	
–	Reserved	√	233	10E9			$+3A4_H$	0 to 15 (15)	
–	Reserved	√	234	10EA			$+3A8_H$	0 to 15 (15)	
–	Reserved	√	235	10EB			$+3AC_H$	0 to 15 (15)	
–	Reserved		236	10EC			$+3B0_H$	0 to 15 (15)	
–	Reserved		237	10ED			$+3B4_H$	0 to 15 (15)	
–	Reserved		238	10EE			$+3B8_H$	0 to 15 (15)	
–	Reserved	√	239	10EF			$+3BC_H$	0 to 15 (15)	
–	Reserved	√	240	10F0			$+3C0_H$	0 to 15 (15)	
–	Reserved	√	241	10F1			$+3C4_H$	0 to 15 (15)	
–	Reserved	√	242	10F2			$+3C8_H$	0 to 15 (15)	
–	Reserved		243	10F3			$+3CC_H$	0 to 15 (15)	
–	Reserved	√	244	10F4			$+3D0_H$	0 to 15 (15)	
–	Reserved	√	245	10F5			$+3D4_H$	0 to 15 (15)	
–	Reserved	√	246	10F6			$+3D8_H$	0 to 15 (15)	
–	Reserved	√	247	10F7			$+3DC_H$	0 to 15 (15)	
–	Reserved		248	10F8			$+3E0_H$	0 to 15 (15)	
–	Reserved		249	10F9			$+3E4_H$	0 to 15 (15)	
–	Reserved		250	10FA			$+3E8_H$	0 to 15 (15)	
–	Reserved		251	10FB			$+3EC_H$	0 to 15 (15)	
–	Reserved	√	252	10FC			$+3F0_H$	0 to 15 (15)	
–	Reserved		253	10FD			$+3F4_H$	0 to 15 (15)	
–	Reserved	√	254	10FE			$+3F8_H$	0 to 15 (15)	
Internal bus errors	This interrupt is a logical OR combination of various internal bus error indications. The exact cause of this error interrupt can be evaluated by use of the bus error status registers. Refer to Section 7.4.9, Bus error status registers for details.		255	10FF			$+3FC_H$	0 to 15 (15)	

Note 1. For a level interrupt, the status register in each module should be cleared during interrupt processing by software. The EICTn bit in EICn is set to 1. The EIRFn bit in EICn cannot be cleared by software.

7.4.3 D1M1 Interrupt Exception Handler and Priority

Table 7.8 D1M1 Interrupt Exception Handler and Priority (1/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
Non-maskable interrupt			(FENMI)	$E0_H$	$+0E0_H$	$+0E0_H$	—		High
FE level interrupt			(FEINT)	$F0_H$	$+0F0_H$	$+0F0_H$	—		
WDTA0	75 % interrupt (INTWDTA0)		0	1000	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+000_H$	0 to 15 (15)	
WDTA1	75 % interrupt (INTWDTA1)		1	1001			$+004_H$	0 to 15 (15)	
RTCA0	1 second interrupt (INTRTCA01S)		2	1002			$+008_H$	0 to 15 (15)	
	Alarm interrupt (INTRTCA0AL)		3	1003			$+00C_H$	0 to 15 (15)	
	Fixed interval interrupt (INTRTCA0R)		4	1004			$+010_H$	0 to 15 (15)	
AWOT0	Timer interrupt (INTAWOT0)		5	1005			$+014_H$	0 to 15 (15)	
Port	External interrupt 0 (INTP0)		6	1006			$+018_H$	0 to 15 (15)	
	External interrupt 1 (INTP1)		7	1007			$+01C_H$	0 to 15 (15)	
	External interrupt 2 (INTP2)		8	1008			$+020_H$	0 to 15 (15)	
	External interrupt 3 (INTP3)		9	1009			$+024_H$	0 to 15 (15)	
	External interrupt 4 (INTP4)		10	100A			$+028_H$	0 to 15 (15)	
	External interrupt 5 (INTP5)		11	100B			$+02C_H$	0 to 15 (15)	
	External interrupt 6 (INTP6)		12	100C			$+030_H$	0 to 15 (15)	
	External interrupt 7 (INTP7)		13	100D			$+034_H$	0 to 15 (15)	
	External interrupt 8 (INTP8)		14	100E			$+038_H$	0 to 15 (15)	
	External interrupt 9 (INTP9)		15	100F			$+03C_H$	0 to 15 (15)	
	External interrupt 10 (INTP10)		16	1010			$+040_H$	0 to 15 (15)	
TAUB0	Channel 0 interrupt (INTTAUB0I0)		17	1011			$+044_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB0I1)		18	1012			$+048_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB0I2)		19	1013			$+04C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB0I3)		20	1014			$+050_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB0I4)		21	1015			$+054_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB0I5)		22	1016			$+058_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB0I6)		23	1017			$+05C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB0I7)		24	1018			$+060_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB0I8)		25	1019			$+064_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB0I9)		26	101A			$+068_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB0I10)		27	101B			$+06C_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB0I11)		28	101C			$+070_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB0I12)		29	101D			$+074_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB0I13)		30	101E			$+078_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB0I14)		31	101F			$+07C_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB0I15)		32	1020			$+080_H$	0 to 15 (15)	

Table 7.8 D1M1 Interrupt Exception Handler and Priority (2/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
TAUB1	Channel 0 interrupt (INTTAUB1I0)		33	1021	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+084_H$	0 to 15 (15)	High
	Channel 1 interrupt (INTTAUB1I1)		34	1022			$+088_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB1I2)		35	1023			$+08C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB1I3)		36	1024			$+090_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB1I4)		37	1025			$+094_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB1I5)		38	1026			$+098_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB1I6)		39	1027			$+09C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB1I7)		40	1028			$+0A0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB1I8)		41	1029			$+0A4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB1I9)		42	102A			$+0A8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB1I10)		43	102B			$+0AC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB1I11)		44	102C			$+0B0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB1I12)		45	102D			$+0B4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB1I13)		46	102E			$+0B8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB1I14)		47	102F			$+0BC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB1I15)		48	1030			$+0C0_H$	0 to 15 (15)	
TAUB2	Channel 0 interrupt (INTTAUB2I0)		49	1031			$+0C4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB2I1)		50	1032			$+0C8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB2I2)		51	1033			$+0CC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB2I3)		52	1034			$+0D0_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB2I4)		53	1035			$+0D4_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB2I5)		54	1036			$+0D8_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB2I6)		55	1037			$+0DC_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB2I7)		56	1038			$+0E0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB2I8)		57	1039			$+0E4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB2I9)		58	103A			$+0E8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB2I10)		59	103B			$+0EC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB2I11)		60	103C			$+0F0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB2I12)		61	103D			$+0F4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB2I13)		62	103E			$+0F8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB2I14)		63	103F			$+0FC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB2I15)		64	1040			$+100_H$	0 to 15 (15)	
ADCE0	Temperature Sensor interrupt (INTADCE0TSN)		65	1041			$+104_H$	0 to 15 (15)	
	Scan group 1 interrupt (INTADCE0I1)		66	1042			$+108_H$	0 to 15 (15)	
	Scan group 2 interrupt (INTADCE0I2)		67	1043			$+10C_H$	0 to 15 (15)	
	Scan group 3 interrupt (INTADCE0I3)		68	1044			$+110_H$	0 to 15 (15)	
RSCAN	Channel 0 to 2 global error interrupt (INTRCANGERR)	√	69	1045			$+114_H$	0 to 15 (15)	
	Channel 0 to 2 RX FIFO interrupt (INTRCANGRECC)	√	70	1046			$+118_H$	0 to 15 (15)	
	Channel 0 error interrupt (INTRCAN0ERR)	√	71	1047			$+11C_H$	0 to 15 (15)	
	Channel 0 COM RX FIFO interrupt (INTRCAN0REC)	√	72	1048			$+120_H$	0 to 15 (15)	
	Channel 0 TX interrupt (INTRCAN0TRX)	√	73	1049			$+124_H$	0 to 15 (15)	
CSIG0	Reception error interrupt (INTCSIGT0IRE)		74	104A			$+128_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG0IR)		75	104B			$+12C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG0IC)		76	104C			$+130_H$	0 to 15 (15)	
CSIH0	Reception error interrupt (INTCSIH0IRE)		77	104D			$+134_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH0IR)		78	104E			$+138_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH0IC)		79	104F			$+13C_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH0IJC)		80	1050			$+140_H$	0 to 15 (15)	

Table 7.8 D1M1 Interrupt Exception Handler and Priority (3/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RLIN30	Status interrupt (INTRLIN30UR2)		81	1051	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+144_H$	0 to 15 (15)	High
	Reception complete interrupt (INTRLIN30UR1)		82	1052			$+148_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN30UR0)		83	1053			$+14C_H$	0 to 15 (15)	
RLIN31	Status interrupt (INTRLIN31UR2)		84	1054			$+150_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN31UR1)		85	1055			$+154_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN31UR0)		86	1056			$+158_H$	0 to 15 (15)	
SG0	Threshold interrupt (INTSG0TI)		87	1057			$+15C_H$	0 to 15 (15)	
SG1	Threshold interrupt (INTSG1TI)		88	1058			$+160_H$	0 to 15 (15)	
SG2	Threshold interrupt (INTSG2TI)		89	1059			$+164_H$	0 to 15 15)	
SG3	Threshold interrupt (INTSG3TI)		90	105A			$+168_H$	0 to 15 (15)	
SG4	Threshold interrupt (INTSG4TI)		91	105B			$+16C_H$	0 to 15 (15)	
OCD/DCU	Debugger hot-attach interrupt (INTDCUTDI)	√	92	105C			$+170_H$	0 to 15 (15)	
DMAC	DMA Transfer Error (INTDMAERR)		93	105D			$+174_H$	0 to 15 (15)	
	DMA channel 0 transfer completion/transfer count compare match interrupt (INTDMA0)		94	105E			$+178_H$	0 to 15 (15)	
	DMA channel 1 transfer completion/transfer count compare match interrupt (INTDMA1)		95	105F			$+17C_H$	0 to 15 (15)	
	DMA channel 2 transfer completion/transfer count compare match interrupt (INTDMA2)		96	1060			$+180_H$	0 to 15 (15)	
	DMA channel 3 transfer completion/transfer count compare match interrupt (INTDMA3)		97	1061			$+184_H$	0 to 15 (15)	
	DMA channel 4 transfer completion/transfer count compare match interrupt (INTDMA4)		98	1062			$+188_H$	0 to 15 (15)	
	DMA channel 5 transfer completion/transfer count compare match interrupt (INTDMA5)		99	1063			$+18C_H$	0 to 15 (15)	
	DMA channel 6 transfer completion/transfer count compare match interrupt (INTDMA6)		100	1064			$+190_H$	0 to 15 (15)	
	DMA channel 7 transfer completion/transfer count compare match interrupt (INTDMA7)		101	1065			$+194_H$	0 to 15 (15)	
	DMA channel 8 transfer completion/transfer count compare match interrupt (INTDMA8)		102	1066			$+198_H$	0 to 15 (15)	
	DMA channel 9 transfer completion/transfer count compare match interrupt (INTDMA9)		103	1067			$+19C_H$	0 to 15 (15)	
	DMA channel 10 transfer completion/transfer count compare match interrupt (INTDMA10)		104	1068			$+1A0_H$	0 to 15 (15)	
	DMA channel 11 transfer completion/transfer count compare match interrupt (INTDMA11)		105	1069			$+1A4_H$	0 to 15 (15)	
	DMA channel 12 transfer completion/transfer count compare match interrupt (INTDMA12)		106	106A			$+1A8_H$	0 to 15 (15)	
	DMA channel 13 transfer completion/transfer count compare match interrupt (INTDMA13)		107	106B			$+1AC_H$	0 to 15 (15)	
	DMA channel 14 transfer completion/transfer count compare match interrupt (INTDMA14)		108	106C			$+1B0_H$	0 to 15 (15)	
	DMA channel 15 transfer completion/transfer count compare match interrupt (INTDMA15)		109	106D			$+1B4_H$	0 to 15 (15)	
RIIC0	Receive error/event interrupt (INTRIIC0LEE)	√	110	106E			$+1B8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC0RI)		111	106F			$+1BC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC0TI)		112	1070			$+1C0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC0TEI)	√	113	1071			$+1C4_H$	0 to 15 (15)	
RIIC1	Receive error/event interrupt (INTRIIC1LEE)	√	114	1072			$+1C8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC1RI)		115	1073			$+1CC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC1TI)		116	1074			$+1D0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC1TEI)	√	117	1075			$+1D4_H$	0 to 15 (15)	

Table 7.8 D1M1 Interrupt Exception Handler and Priority (4/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RSCAN1	Channel 1 error interrupt (INTRCAN1ERR)	√	118	1076	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+1D8_H$	0 to 15 (15)	High
	Channel 1 COM RX FIFO interrupt (INTRCAN1REC)	√	119	1077			$+1DC_H$	0 to 15 (15)	
	Channel 1 TX interrupt (INTRCAN1TRX)	√	120	1078			$+1E0_H$	0 to 15 (15)	
TAUJ0	Channel 0 interrupt (INTTAUJ0I0)		121	1079			$+1E4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUJ0I1)		122	107A			$+1E8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUJ0I2)		123	107B			$+1EC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUJ0I3)		124	107C			$+1F0_H$	0 to 15 (15)	
OSTM0	Timer interrupt (INTOSTM0)		125	107D			$+1F4_H$	0 to 15 (15)	
OSTM1	Timer interrupt (INTOSTM1)		126	107E			$+1F8_H$	0 to 15 (15)	
CSIG1	Reception error interrupt (INTCSIGT1IRE)		127	107F			$+1FC_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG1IR)		128	1080			$+200_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG1IC)		129	1081			$+204_H$	0 to 15 15)	
CSIG2	Reception error interrupt (INTCSIGT2IRE)		130	1082			$+208_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG2IR)		131	1083			$+20C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG2IC)		132	1084			$+210_H$	0 to 15 (15)	
CSIG3	Reception error interrupt (INTCSIGT3IRE)		133	1085			$+214_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG3IR)		134	1086			$+218_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG3IC)		135	1087			$+21C_H$	0 to 15 (15)	
CSIH1	Reception error interrupt (INTCSIH1IRE)		136	1088			$+220_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH1IR)		137	1089			$+224_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH1IC)		138	108A			$+228_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH1IJC)		139	108B			$+22C_H$	0 to 15 (15)	

Table 7.8 D1M1 Interrupt Exception Handler and Priority (5/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
PWM Generators and Diagnostic	A/D Converter trigger queue full interrupt (INTQFULL)		140	108C	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+230_H$	0 to 15 (15)	High
	PWM Generator channel 0 interrupt (INTPWGA0)		141	108D			$+234_H$	0 to 15 (15)	
	PWM Generator channel 1 interrupt (INTPWGA1)		142	108E			$+238_H$	0 to 15 (15)	
	PWM Generator channel 2 interrupt (INTPWGA2)		143	108F			$+23C_H$	0 to 15 (15)	
	PWM Generator channel 3 interrupt (INTPWGA3)		144	1090			$+240_H$	0 to 15 (15)	
	PWM Generator channel 4 interrupt (INTPWGA4)		145	1091			$+244_H$	0 to 15 (15)	
	PWM Generator channel 5 interrupt (INTPWGA5)		146	1092			$+248_H$	0 to 15 (15)	
	PWM Generator channel 6 interrupt (INTPWGA6)		147	1093			$+24C_H$	0 to 15 (15)	
	PWM Generator channel 7 interrupt (INTPWGA7)		148	1094			$+250_H$	0 to 15 (15)	
	PWM Generator channel 8 interrupt (INTPWGA8)		149	1095			$+254_H$	0 to 15 (15)	
	PWM Generator channel 9 interrupt (INTPWGA9)		150	1096			$+258_H$	0 to 15 (15)	
	PWM Generator channel 10 interrupt (INTPWGA10)		151	1097			$+25C_H$	0 to 15 (15)	
	PWM Generator channel 11 interrupt (INTPWGA11)		152	1098			$+260_H$	0 to 15 (15)	
	PWM Generator channel 12 interrupt (INTPWGA12)		153	1099			$+264_H$	0 to 15 (15)	
	PWM Generator channel 13 interrupt (INTPWGA13)		154	109A			$+268_H$	0 to 15 (15)	
	PWM Generator channel 14 interrupt (INTPWGA14)		155	109B			$+26C_H$	0 to 15 (15)	
	PWM Generator channel 15 interrupt (INTPWGA15)		156	109C			$+270_H$	0 to 15 (15)	
	PWM Generator channel 16 interrupt (INTPWGA16)		157	109D			$+274_H$	0 to 15 (15)	
	PWM Generator channel 17 interrupt (INTPWGA17)		158	109E			$+278_H$	0 to 15 (15)	
	PWM Generator channel 18 interrupt (INTPWGA18)		159	109F			$+27C_H$	0 to 15 (15)	
	PWM Generator channel 19 interrupt (INTPWGA19)		160	10A0			$+280_H$	0 to 15 (15)	
	PWM Generator channel 20 interrupt (INTPWGA20)		161	10A1			$+284_H$	0 to 15 (15)	
	PWM Generator channel 21 interrupt (INTPWGA21)		162	10A2			$+288_H$	0 to 15 (15)	
	PWM Generator channel 22 interrupt (INTPWGA22)		163	10A3			$+28C_H$	0 to 15 (15)	
	PWM Generator channel 23 interrupt (INTPWGA23)		164	10A4			$+290_H$	0 to 15 (15)	
ICU-S2	CMD registers ready to write interrupt (INTICUSTWRDY)		165	10A5			$+294_H$	0 to 15 (15)	
	CMD registers ready to read interrupt (INTICUSTRRDY)		166	10A6			$+298_H$	0 to 15 (15)	
–	Reserved	√	167	10A7			$+29C_H$	0 to 15 (15)	
–	Reserved		168	10A8			$+2A0_H$	0 to 15 (15)	
–	Reserved		169	10A9			$+2A4_H$	0 to 15 (15)	
–	Reserved	√	170	10AA			$+2A8_H$	0 to 15 (15)	
Flash control	Flash access error (INTFLERR)	√	171	10AB			$+2AC_H$	0 to 15 (15)	
Flash control	Flash sequencer ready (INTFLENDNM)		172	10AC			$+2B0_H$	0 to 15 (15)	
–	Reserved	√	173	10AD			$+2B4_H$	0 to 15 (15)	
LCBI0	Read data ready interrupt (INTLCBI0RDY)		174	10AE			$+2B8_H$	0 to 15 (15)	
	Write buffer empty interrupt (INTLCBI0EMPTY)		175	10AF			$+2BC_H$	0 to 15 (15)	
	Write buffer half full interrupt (INTLCBI0HALF)		176	10B0			$+2C0_H$	0 to 15 (15)	
	Write buffer full interrupt (INTLCBI0FULL)		177	10B1			$+2C4_H$	0 to 15 (15)	
	Write buffer quarter full interrupt (INTLCBI0QTR)		178	10B2			$+2C8_H$	0 to 15 (15)	
	Write buffer three quarters full interrupt (INTLCBI03QTR)		179	10B3			$+2CC_H$	0 to 15 (15)	
RSCAN2	Channel 2 error interrupt (INTRCAN2ERR)	√	180	10B4			$+2D0_H$	0 to 15 (15)	
	Channel 2 COM RX FIFO interrupt (INTRCAN2REC)	√	181	10B5			$+2D4_H$	0 to 15 (15)	
	Channel 2 TX interrupt (INTRCAN2TRX)	√	182	10B6			$+2D8_H$	0 to 15 (15)	
PCMP0	FIFO buffer fill interrupt (INTPCMP0FFIL)		183	10B7			$+2DC_H$	0 to 15 (15)	
	Error interrupt (INTPCMP0FERR)		184	10B8			$+2E0_H$	0 to 15 (15)	

Table 7.8 D1M1 Interrupt Exception Handler and Priority (6/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
ISM0	Target position reached interrupt (INTISM0REACHED)		185	10B9	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+2E4_H$	0 to 15 (15)	High
	Idle state interrupt (INTISM0DONE)		186	10BA			$+2E8_H$	0 to 15 (15)	
	ZPD measurement started interrupt (INTISM0ZPDAD)		187	10BB			$+2EC_H$	0 to 15 (15)	
–	Reserved	√	188	10BC			$+2F0_H$	0 to 15 (15)	
SSIF0	Multi-purpose interrupt (INTSSIF0)	√	189	10BD			$+2F4_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF0RX)	√	190	10BE			$+2F8_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF0TX)	√	191	10BF			$+2FC_H$	0 to 15 (15)	
SSIF1	Multi-purpose interrupt (INTSSIF1)	√	192	10C0			$+300_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF1RX)	√	193	10C1			$+304_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF1TX)	√	194	10C2			$+308_H$	0 to 15 (15)	
–	Reserved	√	195	10C3			$+30C_H$	0 to 15 (15)	
GPU2D	Pause interrupt (INTGPU2D0PAUSE)	√	196	10C4			$+310_H$	0 to 15 (15)	
	SYNC interrupt (INTGPU2D0SYNC)	√	197	10C5			$+314_H$	0 to 15 (15)	
	Stop / Stall / MBI Error interrupt (INTGPU2D0SP)	√	198	10C6			$+318_H$	0 to 15 (15)	
VDCE0	Error interrupt (INTVDCE0ERR)	√	199	10C7			$+31C_H$	0 to 15 (15)	
	VBLANK detection at Graphics 3 (INTVDCE0GR3VBLANK)		200	10C8			$+320_H$	0 to 15 (15)	
	VSYNC input at Scaler 0 interrupt (INTVDCE0S0VIVSYNC)	√	201	10C9			$+324_H$	0 to 15 (15)	
	VSYNC output at Scaler 0 interrupt (INTVDCE0S0LOVSYNC)	√	202	10CA			$+328_H$	0 to 15 (15)	
	Line detection of designated line at Graphics3 interrupt (INTVDCE0GR3VLINE)	√	203	10CB			$+32C_H$	0 to 15 (15)	
	End of field for record function at Scaler 0 interrupt (INTVDCE0S0VFIELD)	√	204	10CC			$+330_H$	0 to 15 (15)	
	Reserved	√	205	10CD			$+334_H$	0 to 15 (15)	
	VSYNC output at Scaler 1 interrupt (INTVDCE0S1LOVSYNC)	√	206	10CE			$+338_H$	0 to 15 (15)	
	Reserved	√	207	10CF			$+33C_H$	0 to 15 (15)	
	VSYNC input at Output Image Render interrupt (INTVDCE0OIRVIVSYNC)	√	208	10D0			$+340_H$	0 to 15 (15)	
	VSYNC output at Output Image Render interrupt (INTVDCE0OIRLOVSYNC)	√	209	10D1			$+344_H$	0 to 15 (15)	
	Line detection of VSYNC at Output Image Render interrupt (INTVDCE0OIRVLINE)	√	210	10D2			$+348_H$	0 to 15 (15)	
–	Reserved	√	211	10D3			$+34C_H$	0 to 15 (15)	
–	Reserved	√	212	10D4			$+350_H$	0 to 15 (15)	
–	Reserved	√	213	10D5			$+354_H$	0 to 15 (15)	
–	Reserved	√	214	10D6			$+358_H$	0 to 15 (15)	
RLIN32	Status interrupt interrupt (INTRLIN32UR2)		215	10D7			$+35C_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN32UR1)		216	10D8			$+360_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN32UR0)		217	10D9			$+364_H$	0 to 15 (15)	
RLIN33	Status interrupt (INTRLIN33UR2)		218	10DA			$+368_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN33UR1)		219	10DB			$+36C_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN33UR0)		220	10DC			$+370_H$	0 to 15 (15)	

Table 7.8 D1M1 Interrupt Exception Handler and Priority (7/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
–	Reserved	√	221	10DD	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+374_H$	0 to 15 (15)	High
–	Reserved		222	10DE			$+378_H$	0 to 15 (15)	
–	Reserved	√	223	10DF			$+37C_H$	0 to 15 (15)	
–	Reserved	√	224	10E0			$+380_H$	0 to 15 (15)	
–	Reserved	√	225	10E1			$+384_H$	0 to 15 (15)	
–	Reserved	√	226	10E2			$+388_H$	0 to 15 (15)	
–	Reserved	√	227	10E3			$+38C_H$	0 to 15 (15)	
–	Reserved	√	228	10E4			$+390_H$	0 to 15 (15)	
–	Reserved	√	229	10E5			$+394_H$	0 to 15 (15)	
–	Reserved	√	230	10E6			$+398_H$	0 to 15 (15)	
–	Reserved	√	231	10E7			$+39C_H$	0 to 15 (15)	
–	Reserved	√	232	10E8			$+3A0_H$	0 to 15 (15)	
–	Reserved	√	233	10E9			$+3A4_H$	0 to 15 (15)	
–	Reserved	√	234	10EA			$+3A8_H$	0 to 15 (15)	
–	Reserved	√	235	10EB			$+3AC_H$	0 to 15 (15)	
–	Reserved		236	10EC			$+3B0_H$	0 to 15 (15)	
–	Reserved		237	10ED			$+3B4_H$	0 to 15 (15)	
–	Reserved		238	10EE			$+3B8_H$	0 to 15 (15)	
–	Reserved	√	239	10EF			$+3BC_H$	0 to 15 (15)	
VOWE	Common interrupt (INTVOWE0)	√	240	10F0			$+3C0_H$	0 to 15 (15)	
–	Reserved	√	241	10F1			$+3C4_H$	0 to 15 (15)	
JCUA	Compression/decompression process interrupt (INTJCUA0EDI)	√	242	10F2			$+3C8_H$	0 to 15 (15)	
	Data transfer interrupt (INTJCUA0DTI)		243	10F3			$+3CC_H$	0 to 15 (15)	
ETNB0	Interrupt request line 0 (INTETNB0LINE0)	√	244	10F4			$+3D0_H$	0 to 15 (15)	
	Interrupt request line 1 (INTETNB0LINE1)	√	245	10F5			$+3D4_H$	0 to 15 (15)	
	Interrupt request line 2 (INTETNB0LINE2)	√	246	10F6			$+3D8_H$	0 to 15 (15)	
	Interrupt request line 3 (INTETNB0LINE3)	√	247	10F7			$+3DC_H$	0 to 15 (15)	
–	Reserved		248	10F8			$+3E0_H$	0 to 15 (15)	
–	Reserved		249	10F9			$+3E4_H$	0 to 15 (15)	
–	Reserved		250	10FA			$+3E8_H$	0 to 15 (15)	
–	Reserved		251	10FB			$+3EC_H$	0 to 15 (15)	
–	Reserved	√	252	10FC			$+3F0_H$	0 to 15 (15)	
–	Reserved		253	10FD			$+3F4_H$	0 to 15 (15)	
–	Reserved	√	254	10FE			$+3F8_H$	0 to 15 (15)	
Internal bus errors	This interrupt is a logical OR combination of various internal bus error indications. The exact cause of this error interrupt can be evaluated by use of the bus error status registers. Refer to Section 7.4.9, Bus error status registers for details.		255	10FF			$+3FC_H$	0 to 15 (15)	

Note 1. For a level interrupt, the status register in each module should be cleared during interrupt processing by software. The EICTn bit in EICn is set to 1. The EIRFn bit in EICn cannot be cleared by software.

7.4.4 D1M1H Interrupt Exception Handler and Priority

Table 7.9 D1M1H Interrupt Exception Handler and Priority (1/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
Non-maskable interrupt			(FENMI)	$E0_H$	$+0E0_H$	$+0E0_H$	—		High
FE level interrupt			(FEINT)	$F0_H$	$+0F0_H$	$+0F0_H$	—		
WDTA0	75 % interrupt (INTWDTA0)		0	1000	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+000_H$	0 to 15 (15)	
WDTA1	75 % interrupt (INTWDTA1)		1	1001			$+004_H$	0 to 15 (15)	
RTCA0	1 second interrupt (INTRTCA01S)		2	1002			$+008_H$	0 to 15 (15)	
	Alarm interrupt (INTRTCA0AL)		3	1003			$+00C_H$	0 to 15 (15)	
	Fixed interval interrupt (INTRTCA0R)		4	1004			$+010_H$	0 to 15 (15)	
AWOT0	Timer interrupt (INTAWOT0)		5	1005			$+014_H$	0 to 15 (15)	
Port	External interrupt 0 (INTP0)		6	1006			$+018_H$	0 to 15 (15)	
	External interrupt 1 (INTP1)		7	1007			$+01C_H$	0 to 15 (15)	
	External interrupt 2 (INTP2)		8	1008			$+020_H$	0 to 15 (15)	
	External interrupt 3 (INTP3)		9	1009			$+024_H$	0 to 15 (15)	
	External interrupt 4 (INTP4)		10	100A			$+028_H$	0 to 15 (15)	
	External interrupt 5 (INTP5)		11	100B			$+02C_H$	0 to 15 (15)	
	External interrupt 6 (INTP6)		12	100C			$+030_H$	0 to 15 (15)	
	External interrupt 7 (INTP7)		13	100D			$+034_H$	0 to 15 (15)	
	External interrupt 8 (INTP8)		14	100E			$+038_H$	0 to 15 (15)	
	External interrupt 9 (INTP9)		15	100F			$+03C_H$	0 to 15 (15)	
	External interrupt 10 (INTP10)		16	1010			$+040_H$	0 to 15 (15)	
TAUB0	Channel 0 interrupt (INTTAUB0I0)		17	1011			$+044_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB0I1)		18	1012			$+048_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB0I2)		19	1013			$+04C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB0I3)		20	1014			$+050_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB0I4)		21	1015			$+054_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB0I5)		22	1016			$+058_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB0I6)		23	1017			$+05C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB0I7)		24	1018			$+060_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB0I8)		25	1019			$+064_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB0I9)		26	101A			$+068_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB0I10)		27	101B			$+06C_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB0I11)		28	101C			$+070_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB0I12)		29	101D			$+074_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB0I13)		30	101E			$+078_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB0I14)		31	101F			$+07C_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB0I15)		32	1020			$+080_H$	0 to 15 (15)	

Table 7.9 D1M1H Interrupt Exception Handler and Priority (2/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
TAUB1	Channel 0 interrupt (INTTAUB1I0)		33	1021	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+084_H$	0 to 15 (15)	High
	Channel 1 interrupt (INTTAUB1I1)		34	1022			$+088_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB1I2)		35	1023			$+08C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB1I3)		36	1024			$+090_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB1I4)		37	1025			$+094_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB1I5)		38	1026			$+098_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB1I6)		39	1027			$+09C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB1I7)		40	1028			$+0A0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB1I8)		41	1029			$+0A4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB1I9)		42	102A			$+0A8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB1I10)		43	102B			$+0AC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB1I11)		44	102C			$+0B0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB1I12)		45	102D			$+0B4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB1I13)		46	102E			$+0B8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB1I14)		47	102F			$+0BC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB1I15)		48	1030			$+0C0_H$	0 to 15 (15)	
TAUB2	Channel 0 interrupt (INTTAUB2I0)		49	1031			$+0C4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB2I1)		50	1032			$+0C8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB2I2)		51	1033			$+0CC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB2I3)		52	1034			$+0D0_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB2I4)		53	1035			$+0D4_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB2I5)		54	1036			$+0D8_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB2I6)		55	1037			$+0DC_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB2I7)		56	1038			$+0E0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB2I8)		57	1039			$+0E4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB2I9)		58	103A			$+0E8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB2I10)		59	103B			$+0EC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB2I11)		60	103C			$+0F0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB2I12)		61	103D			$+0F4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB2I13)		62	103E			$+0F8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB2I14)		63	103F			$+0FC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB2I15)		64	1040			$+100_H$	0 to 15 (15)	
ADCE0	Temperature Sensor interrupt (INTADCE0TSN)		65	1041			$+104_H$	0 to 15 (15)	
	Scan group 1 interrupt (INTADCE0I1)		66	1042			$+108_H$	0 to 15 (15)	
	Scan group 2 interrupt (INTADCE0I2)		67	1043			$+10C_H$	0 to 15 (15)	
	Scan group 3 interrupt (INTADCE0I3)		68	1044			$+110_H$	0 to 15 (15)	
RSCAN	Channel 0 to 2 global error interrupt (INTRCANGERR)	√	69	1045			$+114_H$	0 to 15 (15)	
	Channel 0 to 2 RX FIFO interrupt (INTRCANGRECC)	√	70	1046			$+118_H$	0 to 15 (15)	
	Channel 0 error interrupt (INTRCAN0ERR)	√	71	1047			$+11C_H$	0 to 15 (15)	
	Channel 0 COM RX FIFO interrupt (INTRCAN0REC)	√	72	1048			$+120_H$	0 to 15 (15)	
	Channel 0 TX interrupt (INTRCAN0TRX)	√	73	1049			$+124_H$	0 to 15 (15)	
CSIG0	Reception error interrupt (INTCSIGT0IRE)		74	104A			$+128_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG0IR)		75	104B			$+12C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG0IC)		76	104C			$+130_H$	0 to 15 (15)	
CSIH0	Reception error interrupt (INTCSIH0IRE)		77	104D			$+134_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH0IR)		78	104E			$+138_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH0IC)		79	104F			$+13C_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH0IJC)		80	1050			$+140_H$	0 to 15 (15)	

Table 7.9 D1M1H Interrupt Exception Handler and Priority (3/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RLIN30	Status interrupt (INTRLIN30UR2)		81	1051	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+144_H$	0 to 15 (15)	High
	Reception complete interrupt (INTRLIN30UR1)		82	1052			$+148_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN30UR0)		83	1053			$+14C_H$	0 to 15 (15)	
RLIN31	Status interrupt (INTRLIN31UR2)		84	1054			$+150_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN31UR1)		85	1055			$+154_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN31UR0)		86	1056			$+158_H$	0 to 15 (15)	
SG0	Threshold interrupt (INTSG0TI)		87	1057			$+15C_H$	0 to 15 (15)	
SG1	Threshold interrupt (INTSG1TI)		88	1058			$+160_H$	0 to 15 (15)	
SG2	Threshold interrupt (INTSG2TI)		89	1059			$+164_H$	0 to 15 15)	
SG3	Threshold interrupt (INTSG3TI)		90	105A			$+168_H$	0 to 15 (15)	
SG4	Threshold interrupt (INTSG4TI)		91	105B			$+16C_H$	0 to 15 (15)	
OCD/DCU	Debugger hot-attach interrupt (INTDCUTDI)	√	92	105C			$+170_H$	0 to 15 (15)	
DMAC	DMA Transfer Error (INTDMAERR)		93	105D			$+174_H$	0 to 15 (15)	
	DMA channel 0 transfer completion/transfer count compare match interrupt (INTDMA0)		94	105E			$+178_H$	0 to 15 (15)	
	DMA channel 1 transfer completion/transfer count compare match interrupt (INTDMA1)		95	105F			$+17C_H$	0 to 15 (15)	
	DMA channel 2 transfer completion/transfer count compare match interrupt (INTDMA2)		96	1060			$+180_H$	0 to 15 (15)	
	DMA channel 3 transfer completion/transfer count compare match interrupt (INTDMA3)		97	1061			$+184_H$	0 to 15 (15)	
	DMA channel 4 transfer completion/transfer count compare match interrupt (INTDMA4)		98	1062			$+188_H$	0 to 15 (15)	
	DMA channel 5 transfer completion/transfer count compare match interrupt (INTDMA5)		99	1063			$+18C_H$	0 to 15 (15)	
	DMA channel 6 transfer completion/transfer count compare match interrupt (INTDMA6)		100	1064			$+190_H$	0 to 15 (15)	
	DMA channel 7 transfer completion/transfer count compare match interrupt (INTDMA7)		101	1065			$+194_H$	0 to 15 (15)	
	DMA channel 8 transfer completion/transfer count compare match interrupt (INTDMA8)		102	1066			$+198_H$	0 to 15 (15)	
	DMA channel 9 transfer completion/transfer count compare match interrupt (INTDMA9)		103	1067			$+19C_H$	0 to 15 (15)	
	DMA channel 10 transfer completion/transfer count compare match interrupt (INTDMA10)		104	1068			$+1A0_H$	0 to 15 (15)	
	DMA channel 11 transfer completion/transfer count compare match interrupt (INTDMA11)		105	1069			$+1A4_H$	0 to 15 (15)	
	DMA channel 12 transfer completion/transfer count compare match interrupt (INTDMA12)		106	106A			$+1A8_H$	0 to 15 (15)	
	DMA channel 13 transfer completion/transfer count compare match interrupt (INTDMA13)		107	106B			$+1AC_H$	0 to 15 (15)	
	DMA channel 14 transfer completion/transfer count compare match interrupt (INTDMA14)		108	106C			$+1B0_H$	0 to 15 (15)	
	DMA channel 15 transfer completion/transfer count compare match interrupt (INTDMA15)		109	106D			$+1B4_H$	0 to 15 (15)	
RIIC0	Receive error/event interrupt (INTRIIC0LEE)	√	110	106E			$+1B8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC0RI)		111	106F			$+1BC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC0TI)		112	1070			$+1C0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC0TEI)	√	113	1071			$+1C4_H$	0 to 15 (15)	
RIIC1	Receive error/event interrupt (INTRIIC1LEE)	√	114	1072			$+1C8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC1RI)		115	1073			$+1CC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC1TI)		116	1074			$+1D0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC1TEI)	√	117	1075			$+1D4_H$	0 to 15 (15)	

Table 7.9 D1M1H Interrupt Exception Handler and Priority (4/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RSCAN1	Channel 1 error interrupt (INTRCAN1ERR)	√	118	1076	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+1D8_H$	0 to 15 (15)	High
	Channel 1 COM RX FIFO interrupt (INTRCAN1REC)	√	119	1077			$+1DC_H$	0 to 15 (15)	
	Channel 1 TX interrupt (INTRCAN1TRX)	√	120	1078			$+1E0_H$	0 to 15 (15)	
TAUJ0	Channel 0 interrupt (INTTAUJ0I0)		121	1079			$+1E4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUJ0I1)		122	107A			$+1E8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUJ0I2)		123	107B			$+1EC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUJ0I3)		124	107C			$+1F0_H$	0 to 15 (15)	
OSTM0	Timer interrupt (INTOSTM0)		125	107D			$+1F4_H$	0 to 15 (15)	
OSTM1	Timer interrupt (INTOSTM1)		126	107E			$+1F8_H$	0 to 15 (15)	
CSIG1	Reception error interrupt (INTCSIGT1IRE)		127	107F			$+1FC_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG1IR)		128	1080			$+200_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG1IC)		129	1081			$+204_H$	0 to 15 15)	
CSIG2	Reception error interrupt (INTCSIGT2IRE)		130	1082			$+208_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG2IR)		131	1083			$+20C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG2IC)		132	1084			$+210_H$	0 to 15 (15)	
CSIG3	Reception error interrupt (INTCSIGT3IRE)		133	1085			$+214_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG3IR)		134	1086			$+218_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG3IC)		135	1087			$+21C_H$	0 to 15 (15)	
CSIH1	Reception error interrupt (INTCSIH1IRE)		136	1088			$+220_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH1IR)		137	1089			$+224_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH1IC)		138	108A			$+228_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH1IJC)		139	108B			$+22C_H$	0 to 15 (15)	

Table 7.9 D1M1H Interrupt Exception Handler and Priority (5/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
PWM Generators and Diagnostic	A/D Converter trigger queue full interrupt (INTQFULL)		140	108C	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+230_H$	0 to 15 (15)	High
	PWM Generator channel 0 interrupt (INTPWGA0)		141	108D			$+234_H$	0 to 15 (15)	
	PWM Generator channel 1 interrupt (INTPWGA1)		142	108E			$+238_H$	0 to 15 (15)	
	PWM Generator channel 2 interrupt (INTPWGA2)		143	108F			$+23C_H$	0 to 15 (15)	
	PWM Generator channel 3 interrupt (INTPWGA3)		144	1090			$+240_H$	0 to 15 (15)	
	PWM Generator channel 4 interrupt (INTPWGA4)		145	1091			$+244_H$	0 to 15 (15)	
	PWM Generator channel 5 interrupt (INTPWGA5)		146	1092			$+248_H$	0 to 15 (15)	
	PWM Generator channel 6 interrupt (INTPWGA6)		147	1093			$+24C_H$	0 to 15 (15)	
	PWM Generator channel 7 interrupt (INTPWGA7)		148	1094			$+250_H$	0 to 15 (15)	
	PWM Generator channel 8 interrupt (INTPWGA8)		149	1095			$+254_H$	0 to 15 (15)	
	PWM Generator channel 9 interrupt (INTPWGA9)		150	1096			$+258_H$	0 to 15 (15)	
	PWM Generator channel 10 interrupt (INTPWGA10)		151	1097			$+25C_H$	0 to 15 (15)	
	PWM Generator channel 11 interrupt (INTPWGA11)		152	1098			$+260_H$	0 to 15 (15)	
	PWM Generator channel 12 interrupt (INTPWGA12)		153	1099			$+264_H$	0 to 15 (15)	
	PWM Generator channel 13 interrupt (INTPWGA13)		154	109A			$+268_H$	0 to 15 (15)	
	PWM Generator channel 14 interrupt (INTPWGA14)		155	109B			$+26C_H$	0 to 15 (15)	
	PWM Generator channel 15 interrupt (INTPWGA15)		156	109C			$+270_H$	0 to 15 (15)	
	PWM Generator channel 16 interrupt (INTPWGA16)		157	109D			$+274_H$	0 to 15 (15)	
	PWM Generator channel 17 interrupt (INTPWGA17)		158	109E			$+278_H$	0 to 15 (15)	
	PWM Generator channel 18 interrupt (INTPWGA18)		159	109F			$+27C_H$	0 to 15 (15)	
	PWM Generator channel 19 interrupt (INTPWGA19)		160	10A0			$+280_H$	0 to 15 (15)	
	PWM Generator channel 20 interrupt (INTPWGA20)		161	10A1			$+284_H$	0 to 15 (15)	
	PWM Generator channel 21 interrupt (INTPWGA21)		162	10A2			$+288_H$	0 to 15 (15)	
	PWM Generator channel 22 interrupt (INTPWGA22)		163	10A3			$+28C_H$	0 to 15 (15)	
	PWM Generator channel 23 interrupt (INTPWGA23)		164	10A4			$+290_H$	0 to 15 (15)	
ICU-S2	CMD registers ready to write interrupt (INTICUSTWRDY)		165	10A5			$+294_H$	0 to 15 (15)	
	CMD registers ready to read interrupt (INTICUSTRRDY)		166	10A6			$+298_H$	0 to 15 (15)	
–	Reserved	√	167	10A7			$+29C_H$	0 to 15 (15)	
–	Reserved		168	10A8			$+2A0_H$	0 to 15 (15)	
–	Reserved		169	10A9			$+2A4_H$	0 to 15 (15)	
–	Reserved	√	170	10AA			$+2A8_H$	0 to 15 (15)	
Flash control	Flash access error (INTFLERR)	√	171	10AB			$+2AC_H$	0 to 15 (15)	
Flash control	Flash sequencer ready (INTFLENDNM)		172	10AC			$+2B0_H$	0 to 15 (15)	
–	Reserved	√	173	10AD			$+2B4_H$	0 to 15 (15)	
–	Reserved		174	10AE			$+2B8_H$	0 to 15 (15)	
–	Reserved		175	10AF			$+2BC_H$	0 to 15 (15)	
–	Reserved		176	10B0			$+2C0_H$	0 to 15 (15)	
–	Reserved		177	10B1			$+2C4_H$	0 to 15 (15)	
–	Reserved		178	10B2			$+2C8_H$	0 to 15 (15)	
–	Reserved		179	10B3			$+2CC_H$	0 to 15 (15)	
RSCAN2	Channel 2 error interrupt (INTRCAN2ERR)	√	180	10B4			$+2D0_H$	0 to 15 (15)	
	Channel 2 COM RX FIFO interrupt (INTRCAN2REC)	√	181	10B5			$+2D4_H$	0 to 15 (15)	
	Channel 2 TX interrupt (INTRCAN2TRX)	√	182	10B6			$+2D8_H$	0 to 15 (15)	
PCMP0	FIFO buffer fill interrupt (INTPCMP0FFIL)		183	10B7			$+2DC_H$	0 to 15 (15)	
	Error interrupt (INTPCMP0FERR)		184	10B8			$+2E0_H$	0 to 15 (15)	

Table 7.9 D1M1H Interrupt Exception Handler and Priority (6/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
ISM0	Target position reached interrupt (INTISM0REACHED)		185	10B9	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+2E4_H$	0 to 15 (15)	High
	Idle state interrupt (INTISM0DONE)		186	10BA			$+2E8_H$	0 to 15 (15)	
	ZPD measurement started interrupt (INTISM0ZPDAD)		187	10BB			$+2EC_H$	0 to 15 (15)	
–	Reserved	√	188	10BC			$+2F0_H$	0 to 15 (15)	
SSIF0	Multi-purpose interrupt (INTSSIF0)	√	189	10BD			$+2F4_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF0RX)	√	190	10BE			$+2F8_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF0TX)	√	191	10BF			$+2FC_H$	0 to 15 (15)	
SSIF1	Multi-purpose interrupt (INTSSIF1)	√	192	10C0			$+300_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF1RX)	√	193	10C1			$+304_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF1TX)	√	194	10C2			$+308_H$	0 to 15 (15)	
SDRA	Underflow of auto refresh - level interrupt (INTSDRA)	√	195	10C3			$+30C_H$	0 to 15 (15)	
GPU2D	Pause interrupt (INTGPU2D0PAUSE)	√	196	10C4			$+310_H$	0 to 15 (15)	
	SYNC interrupt (INTGPU2D0SYNC)	√	197	10C5			$+314_H$	0 to 15 (15)	
	Stop / Stall / MBI Error interrupt (INTGPU2D0SP)	√	198	10C6			$+318_H$	0 to 15 (15)	
VDCE0	Error interrupt (INTVDCE0ERR)	√	199	10C7			$+31C_H$	0 to 15 (15)	
	VBLANK detection at Graphics 3 (INTVDCE0GR3VBLANK)		200	10C8			$+320_H$	0 to 15 (15)	
	VSYNC input at Scaler 0 interrupt (INTVDCE0S0VIVSYNC)	√	201	10C9			$+324_H$	0 to 15 (15)	
	VSYNC output at Scaler 0 interrupt (INTVDCE0S0LOVSYNC)	√	202	10CA			$+328_H$	0 to 15 (15)	
	Line detection of designated line at Graphics3 interrupt (INTVDCE0GR3VLINE)	√	203	10CB			$+32C_H$	0 to 15 (15)	
	End of field for record function at Scaler 0 interrupt (INTVDCE0S0VFIELD)	√	204	10CC			$+330_H$	0 to 15 (15)	
	Reserved	√	205	10CD			$+334_H$	0 to 15 (15)	
	VSYNC output at Scaler 1 interrupt (INTVDCE0S1LOVSYNC)	√	206	10CE			$+338_H$	0 to 15 (15)	
	Reserved	√	207	10CF			$+33C_H$	0 to 15 (15)	
	VSYNC input at Output Image Render interrupt (INTVDCE0OIRVIVSYNC)	√	208	10D0			$+340_H$	0 to 15 (15)	
	VSYNC output at Output Image Render interrupt (INTVDCE0OIRLOVSYNC)	√	209	10D1			$+344_H$	0 to 15 (15)	
	Line detection of VSYNC at Output Image Render interrupt (INTVDCE0OIRVLINE)	√	210	10D2			$+348_H$	0 to 15 (15)	
–	Reserved	√	211	10D3			$+34C_H$	0 to 15 (15)	
–	Reserved	√	212	10D4			$+350_H$	0 to 15 (15)	
–	Reserved	√	213	10D5			$+354_H$	0 to 15 (15)	
–	Reserved	√	214	10D6			$+358_H$	0 to 15 (15)	
RLIN32	Status interrupt interrupt (INTRLIN32UR2)		215	10D7			$+35C_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN32UR1)		216	10D8			$+360_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN32UR0)		217	10D9			$+364_H$	0 to 15 (15)	
RLIN33	Status interrupt (INTRLIN33UR2)		218	10DA			$+368_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN33UR1)		219	10DB			$+36C_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN33UR0)		220	10DC			$+370_H$	0 to 15 (15)	

Table 7.9 D1M1H Interrupt Exception Handler and Priority (7/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
–	Reserved	√	221	10DD	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+374_H$	0 to 15 (15)	High
–	Reserved		222	10DE			$+378_H$	0 to 15 (15)	
–	Reserved	√	223	10DF			$+37C_H$	0 to 15 (15)	
–	Reserved	√	224	10E0			$+380_H$	0 to 15 (15)	
–	Reserved	√	225	10E1			$+384_H$	0 to 15 (15)	
–	Reserved	√	226	10E2			$+388_H$	0 to 15 (15)	
–	Reserved	√	227	10E3			$+38C_H$	0 to 15 (15)	
–	Reserved	√	228	10E4			$+390_H$	0 to 15 (15)	
–	Reserved	√	229	10E5			$+394_H$	0 to 15 (15)	
–	Reserved	√	230	10E6			$+398_H$	0 to 15 (15)	
–	Reserved	√	231	10E7			$+39C_H$	0 to 15 (15)	
–	Reserved	√	232	10E8			$+3A0_H$	0 to 15 (15)	
–	Reserved	√	233	10E9			$+3A4_H$	0 to 15 (15)	
–	Reserved	√	234	10EA			$+3A8_H$	0 to 15 (15)	
–	Reserved	√	235	10EB			$+3AC_H$	0 to 15 (15)	
–	Reserved		236	10EC			$+3B0_H$	0 to 15 (15)	
–	Reserved		237	10ED			$+3B4_H$	0 to 15 (15)	
–	Reserved		238	10EE			$+3B8_H$	0 to 15 (15)	
–	Reserved	√	239	10EF			$+3BC_H$	0 to 15 (15)	
VOWE	Common interrupt (INTVOWE0)	√	240	10F0			$+3C0_H$	0 to 15 (15)	
–	Reserved	√	241	10F1			$+3C4_H$	0 to 15 (15)	
JCUA	Compression/decompression process interrupt (INTJCUA0EDI)	√	242	10F2			$+3C8_H$	0 to 15 (15)	
	Data transfer interrupt (INTJCUA0DTI)		243	10F3			$+3CC_H$	0 to 15 (15)	
ETNB0	Interrupt request line 0 (INTETNB0LINE0)	√	244	10F4			$+3D0_H$	0 to 15 (15)	
	Interrupt request line 1 (INTETNB0LINE1)	√	245	10F5			$+3D4_H$	0 to 15 (15)	
	Interrupt request line 2 (INTETNB0LINE2)	√	246	10F6			$+3D8_H$	0 to 15 (15)	
	Interrupt request line 3 (INTETNB0LINE3)	√	247	10F7			$+3DC_H$	0 to 15 (15)	
–	Reserved		248	10F8			$+3E0_H$	0 to 15 (15)	
–	Reserved		249	10F9			$+3E4_H$	0 to 15 (15)	
–	Reserved		250	10FA			$+3E8_H$	0 to 15 (15)	
–	Reserved		251	10FB			$+3EC_H$	0 to 15 (15)	
–	Reserved	√	252	10FC			$+3F0_H$	0 to 15 (15)	
–	Reserved		253	10FD			$+3F4_H$	0 to 15 (15)	
–	Reserved	√	254	10FE			$+3F8_H$	0 to 15 (15)	
Internal bus errors	This interrupt is a logical OR combination of various internal bus error indications. The exact cause of this error interrupt can be evaluated by use of the bus error status registers. Refer to Section 7.4.9, Bus error status registers for details.		255	10FF			$+3FC_H$	0 to 15 (15)	

Note 1. For a level interrupt, the status register in each module should be cleared during interrupt processing by software. The EICTn bit in EICn is set to 1. The EIRFn bit in EICn cannot be cleared by software.

7.4.5 D1M2 Interrupt Exception Handler and Priority

Table 7.10 D1M2 Interrupt Exception Handler and Priority (1/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
Non-maskable interrupt			(FENMI)	$E0_H$	$+0E0_H$	$+0E0_H$	—		High
FE level interrupt			(FEINT)	$F0_H$	$+0F0_H$	$+0F0_H$	—		
WDTA0	75 % interrupt (INTWDTA0)		0	1000	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+000_H$	0 to 15 (15)	
WDTA1	75 % interrupt (INTWDTA1)		1	1001			$+004_H$	0 to 15 (15)	
RTCA0	1 second interrupt (INTRTCA01S)		2	1002			$+008_H$	0 to 15 (15)	
	Alarm interrupt (INTRTCA0AL)		3	1003			$+00C_H$	0 to 15 (15)	
	Fixed interval interrupt (INTRTCA0R)		4	1004			$+010_H$	0 to 15 (15)	
AWOT0	Timer interrupt (INTAWOT0)		5	1005			$+014_H$	0 to 15 (15)	
Port	External interrupt 0 (INTP0)		6	1006			$+018_H$	0 to 15 (15)	
	External interrupt 1 (INTP1)		7	1007			$+01C_H$	0 to 15 (15)	
	External interrupt 2 (INTP2)		8	1008			$+020_H$	0 to 15 (15)	
	External interrupt 3 (INTP3)		9	1009			$+024_H$	0 to 15 (15)	
	External interrupt 4 (INTP4)		10	100A			$+028_H$	0 to 15 (15)	
	External interrupt 5 (INTP5)		11	100B			$+02C_H$	0 to 15 (15)	
	External interrupt 6 (INTP6)		12	100C			$+030_H$	0 to 15 (15)	
	External interrupt 7 (INTP7)		13	100D			$+034_H$	0 to 15 (15)	
	External interrupt 8 (INTP8)		14	100E			$+038_H$	0 to 15 (15)	
	External interrupt 9 (INTP9)		15	100F			$+03C_H$	0 to 15 (15)	
	External interrupt 10 (INTP10)		16	1010			$+040_H$	0 to 15 (15)	
TAUB0	Channel 0 interrupt (INTTAUB0I0)		17	1011			$+044_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB0I1)		18	1012			$+048_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB0I2)		19	1013			$+04C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB0I3)		20	1014			$+050_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB0I4)		21	1015			$+054_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB0I5)		22	1016			$+058_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB0I6)		23	1017			$+05C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB0I7)		24	1018			$+060_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB0I8)		25	1019			$+064_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB0I9)		26	101A			$+068_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB0I10)		27	101B			$+06C_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB0I11)		28	101C			$+070_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB0I12)		29	101D			$+074_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB0I13)		30	101E			$+078_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB0I14)		31	101F			$+07C_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB0I15)		32	1020			$+080_H$	0 to 15 (15)	

Table 7.10 D1M2 Interrupt Exception Handler and Priority (2/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
TAUB1	Channel 0 interrupt (INTTAUB1I0)		33	1021	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+084_H$	0 to 15 (15)	High
	Channel 1 interrupt (INTTAUB1I1)		34	1022			$+088_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB1I2)		35	1023			$+08C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB1I3)		36	1024			$+090_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB1I4)		37	1025			$+094_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB1I5)		38	1026			$+098_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB1I6)		39	1027			$+09C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB1I7)		40	1028			$+0A0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB1I8)		41	1029			$+0A4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB1I9)		42	102A			$+0A8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB1I10)		43	102B			$+0AC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB1I11)		44	102C			$+0B0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB1I12)		45	102D			$+0B4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB1I13)		46	102E			$+0B8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB1I14)		47	102F			$+0BC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB1I15)		48	1030			$+0C0_H$	0 to 15 (15)	
TAUB2	Channel 0 interrupt (INTTAUB2I0)		49	1031			$+0C4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB2I1)		50	1032			$+0C8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB2I2)		51	1033			$+0CC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB2I3)		52	1034			$+0D0_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB2I4)		53	1035			$+0D4_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB2I5)		54	1036			$+0D8_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB2I6)		55	1037			$+0DC_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB2I7)		56	1038			$+0E0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB2I8)		57	1039			$+0E4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB2I9)		58	103A			$+0E8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB2I10)		59	103B			$+0EC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB2I11)		60	103C			$+0F0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB2I12)		61	103D			$+0F4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB2I13)		62	103E			$+0F8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB2I14)		63	103F			$+0FC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB2I15)		64	1040			$+100_H$	0 to 15 (15)	
ADCE0	Temperature Sensor interrupt (INTADCE0TSN)		65	1041			$+104_H$	0 to 15 (15)	
	Scan group 1 interrupt (INTADCE0I1)		66	1042			$+108_H$	0 to 15 (15)	
	Scan group 2 interrupt (INTADCE0I2)		67	1043			$+10C_H$	0 to 15 (15)	
	Scan group 3 interrupt (INTADCE0I3)		68	1044			$+110_H$	0 to 15 (15)	
RSCAN	Channel 0 to 2 global error interrupt (INTRCANGERR)	√	69	1045			$+114_H$	0 to 15 (15)	
	Channel 0 to 2 RX FIFO interrupt (INTRCANGRECC)	√	70	1046			$+118_H$	0 to 15 (15)	
	Channel 0 error interrupt (INTRCAN0ERR)	√	71	1047			$+11C_H$	0 to 15 (15)	
	Channel 0 COM RX FIFO interrupt (INTRCAN0REC)	√	72	1048			$+120_H$	0 to 15 (15)	
	Channel 0 TX interrupt (INTRCAN0TRX)	√	73	1049			$+124_H$	0 to 15 (15)	
CSIG0	Reception error interrupt (INTCSIGT0IRE)		74	104A			$+128_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG0IR)		75	104B			$+12C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG0IC)		76	104C			$+130_H$	0 to 15 (15)	
CSIH0	Reception error interrupt (INTCSIH0IRE)		77	104D			$+134_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH0IR)		78	104E			$+138_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH0IC)		79	104F			$+13C_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH0IJC)		80	1050			$+140_H$	0 to 15 (15)	

Table 7.10 D1M2 Interrupt Exception Handler and Priority (3/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RLIN30	Status interrupt (INTRLIN30UR2)		81	1051	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+144_H$	0 to 15 (15)	High
	Reception complete interrupt (INTRLIN30UR1)		82	1052			$+148_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN30UR0)		83	1053			$+14C_H$	0 to 15 (15)	
RLIN31	Status interrupt (INTRLIN31UR2)		84	1054			$+150_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN31UR1)		85	1055			$+154_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN31UR0)		86	1056			$+158_H$	0 to 15 (15)	
SG0	Threshold interrupt (INTSG0TI)		87	1057			$+15C_H$	0 to 15 (15)	
SG1	Threshold interrupt (INTSG1TI)		88	1058			$+160_H$	0 to 15 (15)	
SG2	Threshold interrupt (INTSG2TI)		89	1059			$+164_H$	0 to 15 15)	
SG3	Threshold interrupt (INTSG3TI)		90	105A			$+168_H$	0 to 15 (15)	
SG4	Threshold interrupt (INTSG4TI)		91	105B			$+16C_H$	0 to 15 (15)	
OCD/DCU	Debugger hot-attach interrupt (INTDCUTDI)	√	92	105C			$+170_H$	0 to 15 (15)	
DMAC	DMA Transfer Error (INTDMAERR)		93	105D			$+174_H$	0 to 15 (15)	
	DMA channel 0 transfer completion/transfer count compare match interrupt (INTDMA0)		94	105E			$+178_H$	0 to 15 (15)	
	DMA channel 1 transfer completion/transfer count compare match interrupt (INTDMA1)		95	105F			$+17C_H$	0 to 15 (15)	
	DMA channel 2 transfer completion/transfer count compare match interrupt (INTDMA2)		96	1060			$+180_H$	0 to 15 (15)	
	DMA channel 3 transfer completion/transfer count compare match interrupt (INTDMA3)		97	1061			$+184_H$	0 to 15 (15)	
	DMA channel 4 transfer completion/transfer count compare match interrupt (INTDMA4)		98	1062			$+188_H$	0 to 15 (15)	
	DMA channel 5 transfer completion/transfer count compare match interrupt (INTDMA5)		99	1063			$+18C_H$	0 to 15 (15)	
	DMA channel 6 transfer completion/transfer count compare match interrupt (INTDMA6)		100	1064			$+190_H$	0 to 15 (15)	
	DMA channel 7 transfer completion/transfer count compare match interrupt (INTDMA7)		101	1065			$+194_H$	0 to 15 (15)	
	DMA channel 8 transfer completion/transfer count compare match interrupt (INTDMA8)		102	1066			$+198_H$	0 to 15 (15)	
	DMA channel 9 transfer completion/transfer count compare match interrupt (INTDMA9)		103	1067			$+19C_H$	0 to 15 (15)	
	DMA channel 10 transfer completion/transfer count compare match interrupt (INTDMA10)		104	1068			$+1A0_H$	0 to 15 (15)	
	DMA channel 11 transfer completion/transfer count compare match interrupt (INTDMA11)		105	1069			$+1A4_H$	0 to 15 (15)	
	DMA channel 12 transfer completion/transfer count compare match interrupt (INTDMA12)		106	106A			$+1A8_H$	0 to 15 (15)	
	DMA channel 13 transfer completion/transfer count compare match interrupt (INTDMA13)		107	106B			$+1AC_H$	0 to 15 (15)	
	DMA channel 14 transfer completion/transfer count compare match interrupt (INTDMA14)		108	106C			$+1B0_H$	0 to 15 (15)	
	DMA channel 15 transfer completion/transfer count compare match interrupt (INTDMA15)		109	106D			$+1B4_H$	0 to 15 (15)	
RIIC0	Receive error/event interrupt (INTRIIC0LEE)	√	110	106E			$+1B8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC0RI)		111	106F			$+1BC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC0TI)		112	1070			$+1C0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC0TEI)	√	113	1071			$+1C4_H$	0 to 15 (15)	
RIIC1	Receive error/event interrupt (INTRIIC1LEE)	√	114	1072			$+1C8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC1RI)		115	1073			$+1CC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC1TI)		116	1074			$+1D0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC1TEI)	√	117	1075			$+1D4_H$	0 to 15 (15)	

Table 7.10 D1M2 Interrupt Exception Handler and Priority (4/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RSCAN1	Channel 1 error interrupt (INTRCAN1ERR)	√	118	1076	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+1D8_H$	0 to 15 (15)	High
	Channel 1 COM RX FIFO interrupt (INTRCAN1REC)	√	119	1077			$+1DC_H$	0 to 15 (15)	
	Channel 1 TX interrupt (INTRCAN1TRX)	√	120	1078			$+1E0_H$	0 to 15 (15)	
TAUJ0	Channel 0 interrupt (INTTAUJ0I0)		121	1079			$+1E4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUJ0I1)		122	107A			$+1E8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUJ0I2)		123	107B			$+1EC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUJ0I3)		124	107C			$+1F0_H$	0 to 15 (15)	
OSTM0	Timer interrupt (INTOSTM0)		125	107D			$+1F4_H$	0 to 15 (15)	
OSTM1	Timer interrupt (INTOSTM1)		126	107E			$+1F8_H$	0 to 15 (15)	
CSIG1	Reception error interrupt (INTCSIGT1IRE)		127	107F			$+1FC_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG1IR)		128	1080			$+200_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG1IC)		129	1081			$+204_H$	0 to 15 15)	
CSIG2	Reception error interrupt (INTCSIGT2IRE)		130	1082			$+208_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG2IR)		131	1083			$+20C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG2IC)		132	1084			$+210_H$	0 to 15 (15)	
CSIG3	Reception error interrupt (INTCSIGT3IRE)		133	1085			$+214_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG3IR)		134	1086			$+218_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG3IC)		135	1087			$+21C_H$	0 to 15 (15)	
CSIH1	Reception error interrupt (INTCSIH1IRE)		136	1088			$+220_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH1IR)		137	1089			$+224_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH1IC)		138	108A			$+228_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH1IJC)		139	108B			$+22C_H$	0 to 15 (15)	

Table 7.10 D1M2 Interrupt Exception Handler and Priority (5/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
PWM Generators and Diagnostic	A/D Converter trigger queue full interrupt (INTQFULL)		140	108C	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+230_H$	0 to 15 (15)	High
	PWM Generator channel 0 interrupt (INTPWGA0)		141	108D			$+234_H$	0 to 15 (15)	
	PWM Generator channel 1 interrupt (INTPWGA1)		142	108E			$+238_H$	0 to 15 (15)	
	PWM Generator channel 2 interrupt (INTPWGA2)		143	108F			$+23C_H$	0 to 15 (15)	
	PWM Generator channel 3 interrupt (INTPWGA3)		144	1090			$+240_H$	0 to 15 (15)	
	PWM Generator channel 4 interrupt (INTPWGA4)		145	1091			$+244_H$	0 to 15 (15)	
	PWM Generator channel 5 interrupt (INTPWGA5)		146	1092			$+248_H$	0 to 15 (15)	
	PWM Generator channel 6 interrupt (INTPWGA6)		147	1093			$+24C_H$	0 to 15 (15)	
	PWM Generator channel 7 interrupt (INTPWGA7)		148	1094			$+250_H$	0 to 15 (15)	
	PWM Generator channel 8 interrupt (INTPWGA8)		149	1095			$+254_H$	0 to 15 (15)	
	PWM Generator channel 9 interrupt (INTPWGA9)		150	1096			$+258_H$	0 to 15 (15)	
	PWM Generator channel 10 interrupt (INTPWGA10)		151	1097			$+25C_H$	0 to 15 (15)	
	PWM Generator channel 11 interrupt (INTPWGA11)		152	1098			$+260_H$	0 to 15 (15)	
	PWM Generator channel 12 interrupt (INTPWGA12)		153	1099			$+264_H$	0 to 15 (15)	
	PWM Generator channel 13 interrupt (INTPWGA13)		154	109A			$+268_H$	0 to 15 (15)	
	PWM Generator channel 14 interrupt (INTPWGA14)		155	109B			$+26C_H$	0 to 15 (15)	
	PWM Generator channel 15 interrupt (INTPWGA15)		156	109C			$+270_H$	0 to 15 (15)	
	PWM Generator channel 16 interrupt (INTPWGA16)		157	109D			$+274_H$	0 to 15 (15)	
	PWM Generator channel 17 interrupt (INTPWGA17)		158	109E			$+278_H$	0 to 15 (15)	
	PWM Generator channel 18 interrupt (INTPWGA18)		159	109F			$+27C_H$	0 to 15 (15)	
	PWM Generator channel 19 interrupt (INTPWGA19)		160	10A0			$+280_H$	0 to 15 (15)	
	PWM Generator channel 20 interrupt (INTPWGA20)		161	10A1			$+284_H$	0 to 15 (15)	
	PWM Generator channel 21 interrupt (INTPWGA21)		162	10A2			$+288_H$	0 to 15 (15)	
	PWM Generator channel 22 interrupt (INTPWGA22)		163	10A3			$+28C_H$	0 to 15 (15)	
	PWM Generator channel 23 interrupt (INTPWGA23)		164	10A4			$+290_H$	0 to 15 (15)	
ICU-S2	CMD registers ready to write interrupt (INTICUSTWRDY)		165	10A5			$+294_H$	0 to 15 (15)	
	CMD registers ready to read interrupt (INTICUSTRRDY)		166	10A6			$+298_H$	0 to 15 (15)	
–	Reserved	√	167	10A7			$+29C_H$	0 to 15 (15)	
–	Reserved		168	10A8			$+2A0_H$	0 to 15 (15)	
–	Reserved		169	10A9			$+2A4_H$	0 to 15 (15)	
–	Reserved	√	170	10AA			$+2A8_H$	0 to 15 (15)	
Flash control	Flash access error (INTFLERR)	√	171	10AB			$+2AC_H$	0 to 15 (15)	
Flash control	Flash sequencer ready (INTFLENDNM)		172	10AC			$+2B0_H$	0 to 15 (15)	
–	Reserved	√	173	10AD			$+2B4_H$	0 to 15 (15)	
–	Reserved		174	10AE			$+2B8_H$	0 to 15 (15)	
–	Reserved		175	10AF			$+2BC_H$	0 to 15 (15)	
–	Reserved		176	10B0			$+2C0_H$	0 to 15 (15)	
–	Reserved		177	10B1			$+2C4_H$	0 to 15 (15)	
–	Reserved		178	10B2			$+2C8_H$	0 to 15 (15)	
–	Reserved		179	10B3			$+2CC_H$	0 to 15 (15)	
RSCAN2	Channel 2 error interrupt (INTRCAN2ERR)	√	180	10B4			$+2D0_H$	0 to 15 (15)	
	Channel 2 COM RX FIFO interrupt (INTRCAN2REC)	√	181	10B5			$+2D4_H$	0 to 15 (15)	
	Channel 2 TX interrupt (INTRCAN2TRX)	√	182	10B6			$+2D8_H$	0 to 15 (15)	
PCMP0	FIFO buffer fill interrupt (INTPCMP0FFIL)		183	10B7			$+2DC_H$	0 to 15 (15)	
	Error interrupt (INTPCMP0FERR)		184	10B8			$+2E0_H$	0 to 15 (15)	

Table 7.10 D1M2 Interrupt Exception Handler and Priority (6/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
ISM0	Target position reached interrupt (INTISM0REACHED)		185	10B9	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+2E4_H$	0 to 15 (15)	High
	Idle state interrupt (INTISM0DONE)		186	10BA			$+2E8_H$	0 to 15 (15)	
	ZPD measurement started interrupt (INTISM0ZPDAD)		187	10BB			$+2EC_H$	0 to 15 (15)	
–	Reserved	√	188	10BC			$+2F0_H$	0 to 15 (15)	
SSIF0	Multi-purpose interrupt (INTSSIF0)	√	189	10BD			$+2F4_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF0RX)	√	190	10BE			$+2F8_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF0TX)	√	191	10BF			$+2FC_H$	0 to 15 (15)	
SSIF1	Multi-purpose interrupt (INTSSIF1)	√	192	10C0			$+300_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF1RX)	√	193	10C1			$+304_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF1TX)	√	194	10C2			$+308_H$	0 to 15 (15)	
–	Reserved	√	195	10C3			$+30C_H$	0 to 15 (15)	
GPU2D	Pause interrupt (INTGPU2D0PAUSE)	√	196	10C4			$+310_H$	0 to 15 (15)	
	SYNC interrupt (INTGPU2D0SYNC)	√	197	10C5			$+314_H$	0 to 15 (15)	
	Stop / Stall / MBI Error interrupt (INTGPU2D0SP)	√	198	10C6			$+318_H$	0 to 15 (15)	
VDCE0	Error interrupt (INTVDCE0ERR)	√	199	10C7			$+31C_H$	0 to 15 (15)	
	VBLANK detection at Graphics 3 (INTVDCE0GR3VBLANK)		200	10C8			$+320_H$	0 to 15 (15)	
	VSYNC input at Scaler 0 interrupt (INTVDCE0S0VIVSYNC)	√	201	10C9			$+324_H$	0 to 15 (15)	
	VSYNC output at Scaler 0 interrupt (INTVDCE0S0LOVSYNC)	√	202	10CA			$+328_H$	0 to 15 (15)	
	Line detection of designated line at Graphics3 interrupt (INTVDCE0GR3VLINE)	√	203	10CB			$+32C_H$	0 to 15 (15)	
	End of field for record function at Scaler 0 interrupt (INTVDCE0S0VFIELD)	√	204	10CC			$+330_H$	0 to 15 (15)	
	Reserved	√	205	10CD			$+334_H$	0 to 15 (15)	
	VSYNC output at Scaler 1 interrupt (INTVDCE0S1LOVSYNC)	√	206	10CE			$+338_H$	0 to 15 (15)	
	Reserved	√	207	10CF			$+33C_H$	0 to 15 (15)	
	VSYNC input at Output Image Render interrupt (INTVDCE0OIRVIVSYNC)	√	208	10D0			$+340_H$	0 to 15 (15)	
	VSYNC output at Output Image Render interrupt (INTVDCE0OIRLOVSYNC)	√	209	10D1			$+344_H$	0 to 15 (15)	
	Line detection of VSYNC at Output Image Render interrupt (INTVDCE0OIRVLINE)	√	210	10D2			$+348_H$	0 to 15 (15)	
–	Reserved	√	211	10D3			$+34C_H$	0 to 15 (15)	
–	Reserved	√	212	10D4			$+350_H$	0 to 15 (15)	
–	Reserved	√	213	10D5			$+354_H$	0 to 15 (15)	
–	Reserved	√	214	10D6			$+358_H$	0 to 15 (15)	
RLIN32	Status interrupt interrupt (INTRLIN32UR2)		215	10D7			$+35C_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN32UR1)		216	10D8			$+360_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN32UR0)		217	10D9			$+364_H$	0 to 15 (15)	
RLIN33	Status interrupt (INTRLIN33UR2)		218	10DA			$+368_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN33UR1)		219	10DB			$+36C_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN33UR0)		220	10DC			$+370_H$	0 to 15 (15)	

Table 7.10 D1M2 Interrupt Exception Handler and Priority (7/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
VDCE1	Error interrupt (INTVDCE1ERR)	√	221	10DD	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+374_H$	0 to 15 (15)	High
	VBLANK detection at Graphics 3 (INTVDCE1GR3VBLANK)		222	10DE			$+378_H$	0 to 15 (15)	
	VSYNC input at Scaler 0 interrupt (INTVDCE1S0VIVSYNC)	√	223	10DF			$+37C_H$	0 to 15 (15)	
	VSYNC output at Scaler 0 interrupt (INTVDCE1S0LOVSYNC)	√	224	10E0			$+380_H$	0 to 15 (15)	
	Line detection of designated line at Graphics3 interrupt (INTVDCE1GR3VLINE)	√	225	10E1			$+384_H$	0 to 15 (15)	
	End of field for record function at Scaler 0 interrupt (INTVDCE1S0VFIELD)	√	226	10E2			$+388_H$	0 to 15 (15)	
	Line detection of designated line at Scalar 0 interrupt (INTVDCE1S0WLINE)	√	227	10E3			$+38C_H$	0 to 15 (15)	
	VSYNC output at Scaler 1 interrupt (INTVDCE1S1LOVSYNC)	√	228	10E4			$+390_H$	0 to 15 (15)	
–	Reserved	√	229	10E5			$+394_H$	0 to 15 (15)	
–	Reserved	√	230	10E6			$+398_H$	0 to 15 (15)	
–	Reserved	√	231	10E7			$+39C_H$	0 to 15 (15)	
–	Reserved	√	232	10E8			$+3A0_H$	0 to 15 (15)	
–	Reserved	√	233	10E9			$+3A4_H$	0 to 15 (15)	
–	Reserved	√	234	10EA			$+3A8_H$	0 to 15 (15)	
–	Reserved	√	235	10EB			$+3AC_H$	0 to 15 (15)	
–	Reserved		236	10EC			$+3B0_H$	0 to 15 (15)	
–	Reserved		237	10ED			$+3B4_H$	0 to 15 (15)	
–	Reserved		238	10EE			$+3B8_H$	0 to 15 (15)	
–	Reserved	√	239	10EF			$+3BC_H$	0 to 15 (15)	
VOWE	Common interrupt (INTVOWE0)	√	240	10F0			$+3C0_H$	0 to 15 (15)	
–	Reserved	√	241	10F1			$+3C4_H$	0 to 15 (15)	
JCUA	Compression/decompression process interrupt (INTJCUA0EDI)	√	242	10F2			$+3C8_H$	0 to 15 (15)	
	Data transfer interrupt (INTJCUA0DTI)		243	10F3			$+3CC_H$	0 to 15 (15)	
ETNB0	Interrupt request line 0 (INTETNB0LINE0)	√	244	10F4			$+3D0_H$	0 to 15 (15)	
	Interrupt request line 1 (INTETNB0LINE1)	√	245	10F5			$+3D4_H$	0 to 15 (15)	
	Interrupt request line 2 (INTETNB0LINE2)	√	246	10F6			$+3D8_H$	0 to 15 (15)	
	Interrupt request line 3 (INTETNB0LINE3)	√	247	10F7			$+3DC_H$	0 to 15 (15)	
–	Reserved		248	10F8			$+3E0_H$	0 to 15 (15)	
–	Reserved		249	10F9			$+3E4_H$	0 to 15 (15)	
–	Reserved		250	10FA			$+3E8_H$	0 to 15 (15)	
–	Reserved		251	10FB			$+3EC_H$	0 to 15 (15)	
–	Reserved	√	252	10FC			$+3F0_H$	0 to 15 (15)	
–	Reserved		253	10FD			$+3F4_H$	0 to 15 (15)	
–	Reserved	√	254	10FE			$+3F8_H$	0 to 15 (15)	
Internal bus errors	This interrupt is a logical OR combination of various internal bus error indications. The exact cause of this error interrupt can be evaluated by use of the bus error status registers. Refer to Section 7.4.9, Bus error status registers for details.		255	10FF			$+3FC_H$	0 to 15 (15)	

Note 1. For a level interrupt, the status register in each module should be cleared during interrupt processing by software. The EICTn bit in EICn is set to 1. The EIRFn bit in EICn cannot be cleared by software.

7.4.6 D1M2H Interrupt Exception Handler and Priority

Table 7.11 D1M2H Interrupt Exception Handler and Priority (1/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
Non-maskable interrupt			(FENMI)	$E0_H$	$+0E0_H$	$+0E0_H$	—		High
FE level interrupt			(FEINT)	$F0_H$	$+0F0_H$	$+0F0_H$	—		
WDTA0	75 % interrupt (INTWDTA0)		0	1000	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+000_H$	0 to 15 (15)	
WDTA1	75 % interrupt (INTWDTA1)		1	1001			$+004_H$	0 to 15 (15)	
RTCA0	1 second interrupt (INTRTCA01S)		2	1002			$+008_H$	0 to 15 (15)	
	Alarm interrupt (INTRTCA0AL)		3	1003			$+00C_H$	0 to 15 (15)	
	Fixed interval interrupt (INTRTCA0R)		4	1004			$+010_H$	0 to 15 (15)	
AWOT0	Timer interrupt (INTAWOT0)		5	1005			$+014_H$	0 to 15 (15)	
Port	External interrupt 0 (INTP0)		6	1006			$+018_H$	0 to 15 (15)	
	External interrupt 1 (INTP1)		7	1007			$+01C_H$	0 to 15 (15)	
	External interrupt 2 (INTP2)		8	1008			$+020_H$	0 to 15 (15)	
	External interrupt 3 (INTP3)		9	1009			$+024_H$	0 to 15 (15)	
	External interrupt 4 (INTP4)		10	100A			$+028_H$	0 to 15 (15)	
	External interrupt 5 (INTP5)		11	100B			$+02C_H$	0 to 15 (15)	
	External interrupt 6 (INTP6)		12	100C			$+030_H$	0 to 15 (15)	
	External interrupt 7 (INTP7)		13	100D			$+034_H$	0 to 15 (15)	
	External interrupt 8 (INTP8)		14	100E			$+038_H$	0 to 15 (15)	
	External interrupt 9 (INTP9)		15	100F			$+03C_H$	0 to 15 (15)	
	External interrupt 10 (INTP10)		16	1010			$+040_H$	0 to 15 (15)	
TAUB0	Channel 0 interrupt (INTTAUB0I0)		17	1011			$+044_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB0I1)		18	1012			$+048_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB0I2)		19	1013			$+04C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB0I3)		20	1014			$+050_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB0I4)		21	1015			$+054_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB0I5)		22	1016			$+058_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB0I6)		23	1017			$+05C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB0I7)		24	1018			$+060_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB0I8)		25	1019			$+064_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB0I9)		26	101A			$+068_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB0I10)		27	101B			$+06C_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB0I11)		28	101C			$+070_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB0I12)		29	101D			$+074_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB0I13)		30	101E			$+078_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB0I14)		31	101F			$+07C_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB0I15)		32	1020			$+080_H$	0 to 15 (15)	

Table 7.11 D1M2H Interrupt Exception Handler and Priority (2/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
TAUB1	Channel 0 interrupt (INTTAUB1I0)		33	1021	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+084_H$	0 to 15 (15)	High
	Channel 1 interrupt (INTTAUB1I1)		34	1022			$+088_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB1I2)		35	1023			$+08C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB1I3)		36	1024			$+090_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB1I4)		37	1025			$+094_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB1I5)		38	1026			$+098_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB1I6)		39	1027			$+09C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB1I7)		40	1028			$+0A0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB1I8)		41	1029			$+0A4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB1I9)		42	102A			$+0A8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB1I10)		43	102B			$+0AC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB1I11)		44	102C			$+0B0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB1I12)		45	102D			$+0B4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB1I13)		46	102E			$+0B8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB1I14)		47	102F			$+0BC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB1I15)		48	1030			$+0C0_H$	0 to 15 (15)	
TAUB2	Channel 0 interrupt (INTTAUB2I0)		49	1031			$+0C4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB2I1)		50	1032			$+0C8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB2I2)		51	1033			$+0CC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB2I3)		52	1034			$+0D0_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB2I4)		53	1035			$+0D4_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB2I5)		54	1036			$+0D8_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB2I6)		55	1037			$+0DC_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB2I7)		56	1038			$+0E0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB2I8)		57	1039			$+0E4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB2I9)		58	103A			$+0E8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB2I10)		59	103B			$+0EC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB2I11)		60	103C			$+0F0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB2I12)		61	103D			$+0F4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB2I13)		62	103E			$+0F8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB2I14)		63	103F			$+0FC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB2I15)		64	1040			$+100_H$	0 to 15 (15)	
ADCE0	Temperature Sensor interrupt (INTADCE0TSN)		65	1041			$+104_H$	0 to 15 (15)	
	Scan group 1 interrupt (INTADCE0I1)		66	1042			$+108_H$	0 to 15 (15)	
	Scan group 2 interrupt (INTADCE0I2)		67	1043			$+10C_H$	0 to 15 (15)	
	Scan group 3 interrupt (INTADCE0I3)		68	1044			$+110_H$	0 to 15 (15)	
RSCAN	Channel 0 to 2 global error interrupt (INTRCANGERR)	√	69	1045			$+114_H$	0 to 15 (15)	
	Channel 0 to 2 RX FIFO interrupt (INTRCANGRECC)	√	70	1046			$+118_H$	0 to 15 (15)	
	Channel 0 error interrupt (INTRCAN0ERR)	√	71	1047			$+11C_H$	0 to 15 (15)	
	Channel 0 COM RX FIFO interrupt (INTRCAN0REC)	√	72	1048			$+120_H$	0 to 15 (15)	
	Channel 0 TX interrupt (INTRCAN0TRX)	√	73	1049			$+124_H$	0 to 15 (15)	
CSIG0	Reception error interrupt (INTCSIGT0IRE)		74	104A			$+128_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG0IR)		75	104B			$+12C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG0IC)		76	104C			$+130_H$	0 to 15 (15)	
CSIH0	Reception error interrupt (INTCSIH0IRE)		77	104D			$+134_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH0IR)		78	104E			$+138_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH0IC)		79	104F			$+13C_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH0IJC)		80	1050			$+140_H$	0 to 15 (15)	

Table 7.11 D1M2H Interrupt Exception Handler and Priority (3/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RLIN30	Status interrupt (INTRLIN30UR2)		81	1051	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+144_H$	0 to 15 (15)	High
	Reception complete interrupt (INTRLIN30UR1)		82	1052			$+148_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN30UR0)		83	1053			$+14C_H$	0 to 15 (15)	
RLIN31	Status interrupt (INTRLIN31UR2)		84	1054			$+150_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN31UR1)		85	1055			$+154_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN31UR0)		86	1056			$+158_H$	0 to 15 (15)	
SG0	Threshold interrupt (INTSG0TI)		87	1057			$+15C_H$	0 to 15 (15)	
SG1	Threshold interrupt (INTSG1TI)		88	1058			$+160_H$	0 to 15 (15)	
SG2	Threshold interrupt (INTSG2TI)		89	1059			$+164_H$	0 to 15 15)	
SG3	Threshold interrupt (INTSG3TI)		90	105A			$+168_H$	0 to 15 (15)	
SG4	Threshold interrupt (INTSG4TI)		91	105B			$+16C_H$	0 to 15 (15)	
OCD/DCU	Debugger hot-attach interrupt (INTDCUTDI)	√	92	105C			$+170_H$	0 to 15 (15)	
DMAC	DMA Transfer Error (INTDMAERR)		93	105D			$+174_H$	0 to 15 (15)	
	DMA channel 0 transfer completion/transfer count compare match interrupt (INTDMA0)		94	105E			$+178_H$	0 to 15 (15)	
	DMA channel 1 transfer completion/transfer count compare match interrupt (INTDMA1)		95	105F			$+17C_H$	0 to 15 (15)	
	DMA channel 2 transfer completion/transfer count compare match interrupt (INTDMA2)		96	1060			$+180_H$	0 to 15 (15)	
	DMA channel 3 transfer completion/transfer count compare match interrupt (INTDMA3)		97	1061			$+184_H$	0 to 15 (15)	
	DMA channel 4 transfer completion/transfer count compare match interrupt (INTDMA4)		98	1062			$+188_H$	0 to 15 (15)	
	DMA channel 5 transfer completion/transfer count compare match interrupt (INTDMA5)		99	1063			$+18C_H$	0 to 15 (15)	
	DMA channel 6 transfer completion/transfer count compare match interrupt (INTDMA6)		100	1064			$+190_H$	0 to 15 (15)	
	DMA channel 7 transfer completion/transfer count compare match interrupt (INTDMA7)		101	1065			$+194_H$	0 to 15 (15)	
	DMA channel 8 transfer completion/transfer count compare match interrupt (INTDMA8)		102	1066			$+198_H$	0 to 15 (15)	
	DMA channel 9 transfer completion/transfer count compare match interrupt (INTDMA9)		103	1067			$+19C_H$	0 to 15 (15)	
	DMA channel 10 transfer completion/transfer count compare match interrupt (INTDMA10)		104	1068			$+1A0_H$	0 to 15 (15)	
	DMA channel 11 transfer completion/transfer count compare match interrupt (INTDMA11)		105	1069			$+1A4_H$	0 to 15 (15)	
	DMA channel 12 transfer completion/transfer count compare match interrupt (INTDMA12)		106	106A			$+1A8_H$	0 to 15 (15)	
	DMA channel 13 transfer completion/transfer count compare match interrupt (INTDMA13)		107	106B			$+1AC_H$	0 to 15 (15)	
	DMA channel 14 transfer completion/transfer count compare match interrupt (INTDMA14)		108	106C			$+1B0_H$	0 to 15 (15)	
	DMA channel 15 transfer completion/transfer count compare match interrupt (INTDMA15)		109	106D			$+1B4_H$	0 to 15 (15)	
RIIC0	Receive error/event interrupt (INTRIIC0LEE)	√	110	106E			$+1B8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC0RI)		111	106F			$+1BC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC0TI)		112	1070			$+1C0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC0TEI)	√	113	1071			$+1C4_H$	0 to 15 (15)	
RIIC1	Receive error/event interrupt (INTRIIC1LEE)	√	114	1072			$+1C8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC1RI)		115	1073			$+1CC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC1TI)		116	1074			$+1D0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC1TEI)	√	117	1075			$+1D4_H$	0 to 15 (15)	

Table 7.11 D1M2H Interrupt Exception Handler and Priority (4/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RSCAN1	Channel 1 error interrupt (INTRCAN1ERR)	√	118	1076	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+1D8_H$	0 to 15 (15)	High
	Channel 1 COM RX FIFO interrupt (INTRCAN1REC)	√	119	1077			$+1DC_H$	0 to 15 (15)	
	Channel 1 TX interrupt (INTRCAN1TRX)	√	120	1078			$+1E0_H$	0 to 15 (15)	
TAUJ0	Channel 0 interrupt (INTTAUJ0I0)		121	1079			$+1E4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUJ0I1)		122	107A			$+1E8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUJ0I2)		123	107B			$+1EC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUJ0I3)		124	107C			$+1F0_H$	0 to 15 (15)	
OSTM0	Timer interrupt (INTOSTM0)		125	107D			$+1F4_H$	0 to 15 (15)	
OSTM1	Timer interrupt (INTOSTM1)		126	107E			$+1F8_H$	0 to 15 (15)	
CSIG1	Reception error interrupt (INTCSIGT1IRE)		127	107F			$+1FC_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG1IR)		128	1080			$+200_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG1IC)		129	1081			$+204_H$	0 to 15 15)	
CSIG2	Reception error interrupt (INTCSIGT2IRE)		130	1082			$+208_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG2IR)		131	1083			$+20C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG2IC)		132	1084			$+210_H$	0 to 15 (15)	
CSIG3	Reception error interrupt (INTCSIGT3IRE)		133	1085			$+214_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG3IR)		134	1086			$+218_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG3IC)		135	1087			$+21C_H$	0 to 15 (15)	
CSIH1	Reception error interrupt (INTCSIH1IRE)		136	1088			$+220_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH1IR)		137	1089			$+224_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH1IC)		138	108A			$+228_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH1IJC)		139	108B			$+22C_H$	0 to 15 (15)	

Table 7.11 D1M2H Interrupt Exception Handler and Priority (5/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
PWM Generators and Diagnostic	A/D Converter trigger queue full interrupt (INTQFULL)		140	108C	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+230_H$	0 to 15 (15)	High
	PWM Generator channel 0 interrupt (INTPWGA0)		141	108D			$+234_H$	0 to 15 (15)	
	PWM Generator channel 1 interrupt (INTPWGA1)		142	108E			$+238_H$	0 to 15 (15)	
	PWM Generator channel 2 interrupt (INTPWGA2)		143	108F			$+23C_H$	0 to 15 (15)	
	PWM Generator channel 3 interrupt (INTPWGA3)		144	1090			$+240_H$	0 to 15 (15)	
	PWM Generator channel 4 interrupt (INTPWGA4)		145	1091			$+244_H$	0 to 15 (15)	
	PWM Generator channel 5 interrupt (INTPWGA5)		146	1092			$+248_H$	0 to 15 (15)	
	PWM Generator channel 6 interrupt (INTPWGA6)		147	1093			$+24C_H$	0 to 15 (15)	
	PWM Generator channel 7 interrupt (INTPWGA7)		148	1094			$+250_H$	0 to 15 (15)	
	PWM Generator channel 8 interrupt (INTPWGA8)		149	1095			$+254_H$	0 to 15 (15)	
	PWM Generator channel 9 interrupt (INTPWGA9)		150	1096			$+258_H$	0 to 15 (15)	
	PWM Generator channel 10 interrupt (INTPWGA10)		151	1097			$+25C_H$	0 to 15 (15)	
	PWM Generator channel 11 interrupt (INTPWGA11)		152	1098			$+260_H$	0 to 15 (15)	
	PWM Generator channel 12 interrupt (INTPWGA12)		153	1099			$+264_H$	0 to 15 (15)	
	PWM Generator channel 13 interrupt (INTPWGA13)		154	109A			$+268_H$	0 to 15 (15)	
	PWM Generator channel 14 interrupt (INTPWGA14)		155	109B			$+26C_H$	0 to 15 (15)	
	PWM Generator channel 15 interrupt (INTPWGA15)		156	109C			$+270_H$	0 to 15 (15)	
	PWM Generator channel 16 interrupt (INTPWGA16)		157	109D			$+274_H$	0 to 15 (15)	
	PWM Generator channel 17 interrupt (INTPWGA17)		158	109E			$+278_H$	0 to 15 (15)	
	PWM Generator channel 18 interrupt (INTPWGA18)		159	109F			$+27C_H$	0 to 15 (15)	
	PWM Generator channel 19 interrupt (INTPWGA19)		160	10A0			$+280_H$	0 to 15 (15)	
	PWM Generator channel 20 interrupt (INTPWGA20)		161	10A1			$+284_H$	0 to 15 (15)	
	PWM Generator channel 21 interrupt (INTPWGA21)		162	10A2			$+288_H$	0 to 15 (15)	
	PWM Generator channel 22 interrupt (INTPWGA22)		163	10A3			$+28C_H$	0 to 15 (15)	
	PWM Generator channel 23 interrupt (INTPWGA23)		164	10A4			$+290_H$	0 to 15 (15)	
ICU-S2	CMD registers ready to write interrupt (INTICUSTWRDY)		165	10A5			$+294_H$	0 to 15 (15)	
	CMD registers ready to read interrupt (INTICUSTRRDY)		166	10A6			$+298_H$	0 to 15 (15)	
–	Reserved	√	167	10A7			$+29C_H$	0 to 15 (15)	
–	Reserved		168	10A8			$+2A0_H$	0 to 15 (15)	
–	Reserved		169	10A9			$+2A4_H$	0 to 15 (15)	
–	Reserved	√	170	10AA			$+2A8_H$	0 to 15 (15)	
Flash control	Flash access error (INTFLERR)	√	171	10AB			$+2AC_H$	0 to 15 (15)	
Flash control	Flash sequencer ready (INTFLENDNM)		172	10AC			$+2B0_H$	0 to 15 (15)	
–	Reserved	√	173	10AD			$+2B4_H$	0 to 15 (15)	
–	Reserved		174	10AE			$+2B8_H$	0 to 15 (15)	
–	Reserved		175	10AF			$+2BC_H$	0 to 15 (15)	
–	Reserved		176	10B0			$+2C0_H$	0 to 15 (15)	
–	Reserved		177	10B1			$+2C4_H$	0 to 15 (15)	
–	Reserved		178	10B2			$+2C8_H$	0 to 15 (15)	
–	Reserved		179	10B3			$+2CC_H$	0 to 15 (15)	
RSCAN2	Channel 2 error interrupt (INTRCAN2ERR)	√	180	10B4			$+2D0_H$	0 to 15 (15)	
	Channel 2 COM RX FIFO interrupt (INTRCAN2REC)	√	181	10B5			$+2D4_H$	0 to 15 (15)	
	Channel 2 TX interrupt (INTRCAN2TRX)	√	182	10B6			$+2D8_H$	0 to 15 (15)	
PCMP0	FIFO buffer fill interrupt (INTPCMP0FFIL)		183	10B7			$+2DC_H$	0 to 15 (15)	
	Error interrupt (INTPCMP0FERR)		184	10B8			$+2E0_H$	0 to 15 (15)	

Table 7.11 D1M2H Interrupt Exception Handler and Priority (6/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
ISM0	Target position reached interrupt (INTISM0REACHED)		185	10B9	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+2E4_H$	0 to 15 (15)	High
	Idle state interrupt (INTISM0DONE)		186	10BA			$+2E8_H$	0 to 15 (15)	
	ZPD measurement started interrupt (INTISM0ZPDAD)		187	10BB			$+2EC_H$	0 to 15 (15)	
–	Reserved	√	188	10BC			$+2F0_H$	0 to 15 (15)	
SSIF0	Multi-purpose interrupt (INTSSIF0)	√	189	10BD			$+2F4_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF0RX)	√	190	10BE			$+2F8_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF0TX)	√	191	10BF			$+2FC_H$	0 to 15 (15)	
SSIF1	Multi-purpose interrupt (INTSSIF1)	√	192	10C0			$+300_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF1RX)	√	193	10C1			$+304_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF1TX)	√	194	10C2			$+308_H$	0 to 15 (15)	
–	Reserved	√	195	10C3			$+30C_H$	0 to 15 (15)	
GPU2D	Pause interrupt (INTGPU2D0PAUSE)	√	196	10C4			$+310_H$	0 to 15 (15)	
	SYNC interrupt (INTGPU2D0SYNC)	√	197	10C5			$+314_H$	0 to 15 (15)	
	Stop / Stall / MBI Error interrupt (INTGPU2D0SP)	√	198	10C6			$+318_H$	0 to 15 (15)	
VDCE0	Error interrupt (INTVDCE0ERR)	√	199	10C7			$+31C_H$	0 to 15 (15)	
	VBLANK detection at Graphics 3 (INTVDCE0GR3VBLANK)		200	10C8			$+320_H$	0 to 15 (15)	
	VSYNC input at Scaler 0 interrupt (INTVDCE0S0VIVSYNC)	√	201	10C9			$+324_H$	0 to 15 (15)	
	VSYNC output at Scaler 0 interrupt (INTVDCE0S0LOVSYNC)	√	202	10CA			$+328_H$	0 to 15 (15)	
	Line detection of designated line at Graphics3 interrupt (INTVDCE0GR3VLINE)	√	203	10CB			$+32C_H$	0 to 15 (15)	
	End of field for record function at Scaler 0 interrupt (INTVDCE0S0VFIELD)	√	204	10CC			$+330_H$	0 to 15 (15)	
	Reserved	√	205	10CD			$+334_H$	0 to 15 (15)	
	VSYNC output at Scaler 1 interrupt (INTVDCE0S1LOVSYNC)	√	206	10CE			$+338_H$	0 to 15 (15)	
	Reserved	√	207	10CF			$+33C_H$	0 to 15 (15)	
	VSYNC input at Output Image Render interrupt (INTVDCE0OIRVIVSYNC)	√	208	10D0			$+340_H$	0 to 15 (15)	
	VSYNC output at Output Image Render interrupt (INTVDCE0OIRLOVSYNC)	√	209	10D1			$+344_H$	0 to 15 (15)	
	Line detection of VSYNC at Output Image Render interrupt (INTVDCE0OIRVLINE)	√	210	10D2			$+348_H$	0 to 15 (15)	
–	Reserved	√	211	10D3			$+34C_H$	0 to 15 (15)	
–	Reserved	√	212	10D4			$+350_H$	0 to 15 (15)	
–	Reserved	√	213	10D5			$+354_H$	0 to 15 (15)	
–	Reserved	√	214	10D6			$+358_H$	0 to 15 (15)	
RLIN32	Status interrupt interrupt (INTRLIN32UR2)		215	10D7			$+35C_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN32UR1)		216	10D8			$+360_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN32UR0)		217	10D9			$+364_H$	0 to 15 (15)	
RLIN33	Status interrupt (INTRLIN33UR2)		218	10DA			$+368_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN33UR1)		219	10DB			$+36C_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN33UR0)		220	10DC			$+370_H$	0 to 15 (15)	

Table 7.11 D1M2H Interrupt Exception Handler and Priority (7/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
VDCE1	Error interrupt (INTVDCE1ERR)	√	221	10DD	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+374_H$	0 to 15 (15)	High
	VBLANK detection at Graphics 3 (INTVDCE1GR3VBLANK)		222	10DE			$+378_H$	0 to 15 (15)	
	VSYNC input at Scaler 0 interrupt (INTVDCE1S0VIVSYNC)	√	223	10DF			$+37C_H$	0 to 15 (15)	
	VSYNC output at Scaler 0 interrupt (INTVDCE1S0LOVSYNC)	√	224	10E0			$+380_H$	0 to 15 (15)	
	Line detection of designated line at Graphics3 interrupt (INTVDCE1GR3VLINE)	√	225	10E1			$+384_H$	0 to 15 (15)	
	End of field for record function at Scaler 0 interrupt (INTVDCE1S0VFIELD)	√	226	10E2			$+388_H$	0 to 15 (15)	
	Line detection of designated line at Scalar 0 interrupt (INTVDCE1S0WLINE)	√	227	10E3			$+38C_H$	0 to 15 (15)	
	VSYNC output at Scaler 1 interrupt (INTVDCE1S1LOVSYNC)	√	228	10E4			$+390_H$	0 to 15 (15)	
–	Reserved	√	229	10E5			$+394_H$	0 to 15 (15)	
–	Reserved	√	230	10E6			$+398_H$	0 to 15 (15)	
–	Reserved	√	231	10E7			$+39C_H$	0 to 15 (15)	
–	Reserved	√	232	10E8			$+3A0_H$	0 to 15 (15)	
–	Reserved	√	233	10E9			$+3A4_H$	0 to 15 (15)	
MLBB0	Status interrupt (INTMLBB0STA0)	√	234	10EA			$+3A8_H$	0 to 15 (15)	
	Status interrupt (INTMLBB0STA1)	√	235	10EB			$+3AC_H$	0 to 15 (15)	
	MediaLB system interrupt (INTMLBB0SYS)		236	10EC			$+3B0_H$	0 to 15 (15)	
–	Reserved		237	10ED			$+3B4_H$	0 to 15 (15)	
MIPI0	MIPI Buffer Overflow interrupt (INTMIPI0OVF)		238	10EE			$+3B8_H$	0 to 15 (15)	
	MIPI Controller interrupt (INTMIPI0CTL)	√	239	10EF			$+3BC_H$	0 to 15 (15)	
VOWE	Common interrupt (INTVOWE0)	√	240	10F0			$+3C0_H$	0 to 15 (15)	
–	Reserved	√	241	10F1			$+3C4_H$	0 to 15 (15)	
JCUA	Compression/decompression process interrupt (INTJCUA0EDI)	√	242	10F2			$+3C8_H$	0 to 15 (15)	
	Data transfer interrupt (INTJCUA0DTI)		243	10F3			$+3CC_H$	0 to 15 (15)	
ETNB0	Interrupt request line 0 (INTETNB0LINE0)	√	244	10F4			$+3D0_H$	0 to 15 (15)	
	Interrupt request line 1 (INTETNB0LINE1)	√	245	10F5			$+3D4_H$	0 to 15 (15)	
	Interrupt request line 2 (INTETNB0LINE2)	√	246	10F6			$+3D8_H$	0 to 15 (15)	
	Interrupt request line 3 (INTETNB0LINE3)	√	247	10F7			$+3DC_H$	0 to 15 (15)	
–	Reserved		248	10F8			$+3E0_H$	0 to 15 (15)	
–	Reserved		249	10F9			$+3E4_H$	0 to 15 (15)	
–	Reserved		250	10FA			$+3E8_H$	0 to 15 (15)	
–	Reserved		251	10FB			$+3EC_H$	0 to 15 (15)	
–	Reserved	√	252	10FC			$+3F0_H$	0 to 15 (15)	
–	Reserved		253	10FD			$+3F4_H$	0 to 15 (15)	
–	Reserved	√	254	10FE			$+3F8_H$	0 to 15 (15)	
Internal bus errors	This interrupt is a logical OR combination of various internal bus error indications. The exact cause of this error interrupt can be evaluated by use of the bus error status registers. Refer to Section 7.4.9, Bus error status registers for details.		255	10FF			$+3FC_H$	0 to 15 (15)	

Note 1. For a level interrupt, the status register in each module should be cleared during interrupt processing by software. The EICTn bit in EICn is set to 1. The EIRFn bit in EICn cannot be cleared by software.

7.4.7 D1M1A Interrupt Exception Handler and Priority

Table 7.12 D1M1A Interrupt Exception Handler and Priority (1/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
Non-maskable interrupt			(FENMI)	$E0_H$	$+0E0_H$	$+0E0_H$	—		High
FE level interrupt			(FEINT)	$F0_H$	$+0F0_H$	$+0F0_H$	—		
WDTA0	75 % interrupt (INTWDTA0)		0	1000	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+000_H$	0 to 15 (15)	
WDTA1	75 % interrupt (INTWDTA1)		1	1001			$+004_H$	0 to 15 (15)	
RTCA0	1 second interrupt (INTRTCA01S)		2	1002			$+008_H$	0 to 15 (15)	
	Alarm interrupt (INTRTCA0AL)		3	1003			$+00C_H$	0 to 15 (15)	
	Fixed interval interrupt (INTRTCA0R)		4	1004			$+010_H$	0 to 15 (15)	
AWOT0	Timer interrupt (INTAWOT0)		5	1005			$+014_H$	0 to 15 (15)	
Port	External interrupt 0 (INTP0)		6	1006			$+018_H$	0 to 15 (15)	
	External interrupt 1 (INTP1)		7	1007			$+01C_H$	0 to 15 (15)	
	External interrupt 2 (INTP2)		8	1008			$+020_H$	0 to 15 (15)	
	External interrupt 3 (INTP3)		9	1009			$+024_H$	0 to 15 (15)	
	External interrupt 4 (INTP4)		10	100A			$+028_H$	0 to 15 (15)	
	External interrupt 5 (INTP5)		11	100B			$+02C_H$	0 to 15 (15)	
	External interrupt 6 (INTP6)		12	100C			$+030_H$	0 to 15 (15)	
	External interrupt 7 (INTP7)		13	100D			$+034_H$	0 to 15 (15)	
	External interrupt 8 (INTP8)		14	100E			$+038_H$	0 to 15 (15)	
	External interrupt 9 (INTP9)		15	100F			$+03C_H$	0 to 15 (15)	
	External interrupt 10 (INTP10)		16	1010			$+040_H$	0 to 15 (15)	
TAUB0	Channel 0 interrupt (INTTAUB0I0)		17	1011			$+044_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB0I1)		18	1012			$+048_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB0I2)		19	1013			$+04C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB0I3)		20	1014			$+050_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB0I4)		21	1015			$+054_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB0I5)		22	1016			$+058_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB0I6)		23	1017			$+05C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB0I7)		24	1018			$+060_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB0I8)		25	1019			$+064_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB0I9)		26	101A			$+068_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB0I10)		27	101B			$+06C_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB0I11)		28	101C			$+070_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB0I12)		29	101D			$+074_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB0I13)		30	101E			$+078_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB0I14)		31	101F			$+07C_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB0I15)		32	1020			$+080_H$	0 to 15 (15)	

Table 7.12 D1M1A Interrupt Exception Handler and Priority (2/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
TAUB1	Channel 0 interrupt (INTTAUB1I0)		33	1021	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+084_H$	0 to 15 (15)	High
	Channel 1 interrupt (INTTAUB1I1)		34	1022			$+088_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB1I2)		35	1023			$+08C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB1I3)		36	1024			$+090_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB1I4)		37	1025			$+094_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB1I5)		38	1026			$+098_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB1I6)		39	1027			$+09C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB1I7)		40	1028			$+0A0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB1I8)		41	1029			$+0A4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB1I9)		42	102A			$+0A8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB1I10)		43	102B			$+0AC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB1I11)		44	102C			$+0B0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB1I12)		45	102D			$+0B4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB1I13)		46	102E			$+0B8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB1I14)		47	102F			$+0BC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB1I15)		48	1030			$+0C0_H$	0 to 15 (15)	
TAUB2	Channel 0 interrupt (INTTAUB2I0)		49	1031			$+0C4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB2I1)		50	1032			$+0C8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB2I2)		51	1033			$+0CC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB2I3)		52	1034			$+0D0_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB2I4)		53	1035			$+0D4_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB2I5)		54	1036			$+0D8_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB2I6)		55	1037			$+0DC_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB2I7)		56	1038			$+0E0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB2I8)		57	1039			$+0E4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB2I9)		58	103A			$+0E8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB2I10)		59	103B			$+0EC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB2I11)		60	103C			$+0F0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB2I12)		61	103D			$+0F4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB2I13)		62	103E			$+0F8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB2I14)		63	103F			$+0FC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB2I15)		64	1040			$+100_H$	0 to 15 (15)	
ADCE0	Temperature Sensor interrupt (INTADCE0TSN)		65	1041			$+104_H$	0 to 15 (15)	
	Scan group 1 interrupt (INTADCE0I1)		66	1042			$+108_H$	0 to 15 (15)	
	Scan group 2 interrupt (INTADCE0I2)		67	1043			$+10C_H$	0 to 15 (15)	
	Scan group 3 interrupt (INTADCE0I3)		68	1044			$+110_H$	0 to 15 (15)	
RSCAN	Channel 0 to 2 global error interrupt (INTRCANGERR)	√	69	1045			$+114_H$	0 to 15 (15)	
	Channel 0 to 2 RX FIFO interrupt (INTRCANGRECC)	√	70	1046			$+118_H$	0 to 15 (15)	
	Channel 0 error interrupt (INTRCAN0ERR)	√	71	1047			$+11C_H$	0 to 15 (15)	
	Channel 0 COM RX FIFO interrupt (INTRCAN0REC)	√	72	1048			$+120_H$	0 to 15 (15)	
	Channel 0 TX interrupt (INTRCAN0TRX)	√	73	1049			$+124_H$	0 to 15 (15)	
CSIG0	Reception error interrupt (INTCSIGT0IRE)		74	104A			$+128_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG0IR)		75	104B			$+12C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG0IC)		76	104C			$+130_H$	0 to 15 (15)	
CSIH0	Reception error interrupt (INTCSIH0IRE)		77	104D			$+134_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH0IR)		78	104E			$+138_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH0IC)		79	104F			$+13C_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH0IJC)		80	1050			$+140_H$	0 to 15 (15)	

Table 7.12 D1M1A Interrupt Exception Handler and Priority (3/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RLIN30	Status interrupt (INTRLIN30UR2)		81	1051	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+144_H$	0 to 15 (15)	High
	Reception complete interrupt (INTRLIN30UR1)		82	1052			$+148_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN30UR0)		83	1053			$+14C_H$	0 to 15 (15)	
RLIN31	Status interrupt (INTRLIN31UR2)		84	1054			$+150_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN31UR1)		85	1055			$+154_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN31UR0)		86	1056			$+158_H$	0 to 15 (15)	
SG0	Threshold interrupt (INTSG0TI)		87	1057			$+15C_H$	0 to 15 (15)	
SG1	Threshold interrupt (INTSG1TI)		88	1058			$+160_H$	0 to 15 (15)	
SG2	Threshold interrupt (INTSG2TI)		89	1059			$+164_H$	0 to 15 15)	
SG3	Threshold interrupt (INTSG3TI)		90	105A			$+168_H$	0 to 15 (15)	
SG4	Threshold interrupt (INTSG4TI)		91	105B			$+16C_H$	0 to 15 (15)	
OCD/DCU	Debugger hot-attach interrupt (INTDCUTDI)	√	92	105C			$+170_H$	0 to 15 (15)	
DMAC	DMA Transfer Error (INTDMAERR)		93	105D			$+174_H$	0 to 15 (15)	
	DMA channel 0 transfer completion/transfer count compare match interrupt (INTDMA0)		94	105E			$+178_H$	0 to 15 (15)	
	DMA channel 1 transfer completion/transfer count compare match interrupt (INTDMA1)		95	105F			$+17C_H$	0 to 15 (15)	
	DMA channel 2 transfer completion/transfer count compare match interrupt (INTDMA2)		96	1060			$+180_H$	0 to 15 (15)	
	DMA channel 3 transfer completion/transfer count compare match interrupt (INTDMA3)		97	1061			$+184_H$	0 to 15 (15)	
	DMA channel 4 transfer completion/transfer count compare match interrupt (INTDMA4)		98	1062			$+188_H$	0 to 15 (15)	
	DMA channel 5 transfer completion/transfer count compare match interrupt (INTDMA5)		99	1063			$+18C_H$	0 to 15 (15)	
	DMA channel 6 transfer completion/transfer count compare match interrupt (INTDMA6)		100	1064			$+190_H$	0 to 15 (15)	
	DMA channel 7 transfer completion/transfer count compare match interrupt (INTDMA7)		101	1065			$+194_H$	0 to 15 (15)	
	DMA channel 8 transfer completion/transfer count compare match interrupt (INTDMA8)		102	1066			$+198_H$	0 to 15 (15)	
	DMA channel 9 transfer completion/transfer count compare match interrupt (INTDMA9)		103	1067			$+19C_H$	0 to 15 (15)	
	DMA channel 10 transfer completion/transfer count compare match interrupt (INTDMA10)		104	1068			$+1A0_H$	0 to 15 (15)	
	DMA channel 11 transfer completion/transfer count compare match interrupt (INTDMA11)		105	1069			$+1A4_H$	0 to 15 (15)	
	DMA channel 12 transfer completion/transfer count compare match interrupt (INTDMA12)		106	106A			$+1A8_H$	0 to 15 (15)	
	DMA channel 13 transfer completion/transfer count compare match interrupt (INTDMA13)		107	106B			$+1AC_H$	0 to 15 (15)	
	DMA channel 14 transfer completion/transfer count compare match interrupt (INTDMA14)		108	106C			$+1B0_H$	0 to 15 (15)	
	DMA channel 15 transfer completion/transfer count compare match interrupt (INTDMA15)		109	106D			$+1B4_H$	0 to 15 (15)	
RIIC0	Receive error/event interrupt (INTRIIC0LEE)	√	110	106E			$+1B8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC0RI)		111	106F			$+1BC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC0TI)		112	1070			$+1C0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC0TEI)	√	113	1071			$+1C4_H$	0 to 15 (15)	
RIIC1	Receive error/event interrupt (INTRIIC1LEE)	√	114	1072			$+1C8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC1RI)		115	1073			$+1CC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC1TI)		116	1074			$+1D0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC1TEI)	√	117	1075			$+1D4_H$	0 to 15 (15)	

Table 7.12 D1M1A Interrupt Exception Handler and Priority (4/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RSCAN1	Channel 1 error interrupt (INTRCAN1ERR)	√	118	1076	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+1D8_H$	0 to 15 (15)	High
	Channel 1 COM RX FIFO interrupt (INTRCAN1REC)	√	119	1077			$+1DC_H$	0 to 15 (15)	
	Channel 1 TX interrupt (INTRCAN1TRX)	√	120	1078			$+1E0_H$	0 to 15 (15)	
TAUJ0	Channel 0 interrupt (INTTAUJ0I0)		121	1079			$+1E4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUJ0I1)		122	107A			$+1E8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUJ0I2)		123	107B			$+1EC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUJ0I3)		124	107C			$+1F0_H$	0 to 15 (15)	
OSTM0	Timer interrupt (INTOSTM0)		125	107D			$+1F4_H$	0 to 15 (15)	
OSTM1	Timer interrupt (INTOSTM1)		126	107E			$+1F8_H$	0 to 15 (15)	
CSIG1	Reception error interrupt (INTCSIGT1IRE)		127	107F			$+1FC_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG1IR)		128	1080			$+200_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG1IC)		129	1081			$+204_H$	0 to 15 15)	
CSIG2	Reception error interrupt (INTCSIGT2IRE)		130	1082			$+208_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG2IR)		131	1083			$+20C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG2IC)		132	1084			$+210_H$	0 to 15 (15)	
CSIG3	Reception error interrupt (INTCSIGT3IRE)		133	1085			$+214_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG3IR)		134	1086			$+218_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG3IC)		135	1087			$+21C_H$	0 to 15 (15)	
CSIH1	Reception error interrupt (INTCSIH1IRE)		136	1088			$+220_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH1IR)		137	1089			$+224_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH1IC)		138	108A			$+228_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH1IJC)		139	108B			$+22C_H$	0 to 15 (15)	

Table 7.12 D1M1A Interrupt Exception Handler and Priority (5/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
PWM Generators and Diagnostic	A/D Converter trigger queue full interrupt (INTQFULL)		140	108C	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+230_H$	0 to 15 (15)	High
	PWM Generator channel 0 interrupt (INTPWGA0)		141	108D			$+234_H$	0 to 15 (15)	
	PWM Generator channel 1 interrupt (INTPWGA1)		142	108E			$+238_H$	0 to 15 (15)	
	PWM Generator channel 2 interrupt (INTPWGA2)		143	108F			$+23C_H$	0 to 15 (15)	
	PWM Generator channel 3 interrupt (INTPWGA3)		144	1090			$+240_H$	0 to 15 (15)	
	PWM Generator channel 4 interrupt (INTPWGA4)		145	1091			$+244_H$	0 to 15 (15)	
	PWM Generator channel 5 interrupt (INTPWGA5)		146	1092			$+248_H$	0 to 15 (15)	
	PWM Generator channel 6 interrupt (INTPWGA6)		147	1093			$+24C_H$	0 to 15 (15)	
	PWM Generator channel 7 interrupt (INTPWGA7)		148	1094			$+250_H$	0 to 15 (15)	
	PWM Generator channel 8 interrupt (INTPWGA8)		149	1095			$+254_H$	0 to 15 (15)	
	PWM Generator channel 9 interrupt (INTPWGA9)		150	1096			$+258_H$	0 to 15 (15)	
	PWM Generator channel 10 interrupt (INTPWGA10)		151	1097			$+25C_H$	0 to 15 (15)	
	PWM Generator channel 11 interrupt (INTPWGA11)		152	1098			$+260_H$	0 to 15 (15)	
	PWM Generator channel 12 interrupt (INTPWGA12)		153	1099			$+264_H$	0 to 15 (15)	
	PWM Generator channel 13 interrupt (INTPWGA13)		154	109A			$+268_H$	0 to 15 (15)	
	PWM Generator channel 14 interrupt (INTPWGA14)		155	109B			$+26C_H$	0 to 15 (15)	
	PWM Generator channel 15 interrupt (INTPWGA15)		156	109C			$+270_H$	0 to 15 (15)	
	PWM Generator channel 16 interrupt (INTPWGA16)		157	109D			$+274_H$	0 to 15 (15)	
	PWM Generator channel 17 interrupt (INTPWGA17)		158	109E			$+278_H$	0 to 15 (15)	
	PWM Generator channel 18 interrupt (INTPWGA18)		159	109F			$+27C_H$	0 to 15 (15)	
	PWM Generator channel 19 interrupt (INTPWGA19)		160	10A0			$+280_H$	0 to 15 (15)	
	PWM Generator channel 20 interrupt (INTPWGA20)		161	10A1			$+284_H$	0 to 15 (15)	
	PWM Generator channel 21 interrupt (INTPWGA21)		162	10A2			$+288_H$	0 to 15 (15)	
	PWM Generator channel 22 interrupt (INTPWGA22)		163	10A3			$+28C_H$	0 to 15 (15)	
	PWM Generator channel 23 interrupt (INTPWGA23)		164	10A4			$+290_H$	0 to 15 (15)	
ICU-S2	CMD registers ready to write interrupt (INTICUSTWRDY)		165	10A5			$+294_H$	0 to 15 (15)	
	CMD registers ready to read interrupt (INTICUSTRRDY)		166	10A6			$+298_H$	0 to 15 (15)	
–	Reserved	√	167	10A7			$+29C_H$	0 to 15 (15)	
–	Reserved		168	10A8			$+2A0_H$	0 to 15 (15)	
–	Reserved		169	10A9			$+2A4_H$	0 to 15 (15)	
–	Reserved	√	170	10AA			$+2A8_H$	0 to 15 (15)	
Flash control	Flash access error (INTFLERR)	√	171	10AB			$+2AC_H$	0 to 15 (15)	
Flash control	Flash sequencer ready (INTFLENDNM)		172	10AC			$+2B0_H$	0 to 15 (15)	
NFMA0	INTNFMA0	√	173	10AD			$+2B4_H$	0 to 15 (15)	
–	Reserved		174	10AE			$+2B8_H$	0 to 15 (15)	
–	Reserved		175	10AF			$+2BC_H$	0 to 15 (15)	
–	Reserved		176	10B0			$+2C0_H$	0 to 15 (15)	
–	Reserved		177	10B1			$+2C4_H$	0 to 15 (15)	
–	Reserved		178	10B2			$+2C8_H$	0 to 15 (15)	
–	Reserved		179	10B3			$+2CC_H$	0 to 15 (15)	
RSCAN2	Channel 2 error interrupt (INTRCAN2ERR)	√	180	10B4			$+2D0_H$	0 to 15 (15)	
	Channel 2 COM RX FIFO interrupt (INTRCAN2REC)	√	181	10B5			$+2D4_H$	0 to 15 (15)	
	Channel 2 TX interrupt (INTRCAN2TRX)	√	182	10B6			$+2D8_H$	0 to 15 (15)	
PCMP0	FIFO buffer fill interrupt (INTPCMP0FFIL)		183	10B7			$+2DC_H$	0 to 15 (15)	
	Error interrupt (INTPCMP0FERR)		184	10B8			$+2E0_H$	0 to 15 (15)	

Table 7.12 D1M1A Interrupt Exception Handler and Priority (6/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
ISM0	Target position reached interrupt (INTISM0REACHED)		185	10B9	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+2E4_H$	0 to 15 (15)	High
	Idle state interrupt (INTISM0DONE)		186	10BA			$+2E8_H$	0 to 15 (15)	
	ZPD measurement started interrupt (INTISM0ZPDAD)		187	10BB			$+2EC_H$	0 to 15 (15)	
–	Reserved	√	188	10BC			$+2F0_H$	0 to 15 (15)	
SSIF0	Multi-purpose interrupt (INTSSIF0)	√	189	10BD			$+2F4_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF0RX)	√	190	10BE			$+2F8_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF0TX)	√	191	10BF			$+2FC_H$	0 to 15 (15)	
SSIF1	Multi-purpose interrupt (INTSSIF1)	√	192	10C0			$+300_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF1RX)	√	193	10C1			$+304_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF1TX)	√	194	10C2			$+308_H$	0 to 15 (15)	
SDRA	Underflow of auto refresh - level interrupt (INTSDRA)	√	195	10C3			$+30C_H$	0 to 15 (15)	
GPU2D	Pause interrupt (INTGPU2D0PAUSE)	√	196	10C4			$+310_H$	0 to 15 (15)	
	SYNC interrupt (INTGPU2D0SYNC)	√	197	10C5			$+314_H$	0 to 15 (15)	
	Stop / Stall / MBI Error interrupt (INTGPU2D0SP)	√	198	10C6			$+318_H$	0 to 15 (15)	
VDCE0	Error interrupt (INTVDCE0ERR)	√	199	10C7			$+31C_H$	0 to 15 (15)	
	VBLANK detection at Graphics 3 (INTVDCE0GR3VBLANK)		200	10C8			$+320_H$	0 to 15 (15)	
	VSYNC input at Scaler 0 interrupt (INTVDCE0S0VIVSYNC)	√	201	10C9			$+324_H$	0 to 15 (15)	
	VSYNC output at Scaler 0 interrupt (INTVDCE0S0LOVSYNC)	√	202	10CA			$+328_H$	0 to 15 (15)	
	Line detection of designated line at Graphics3 interrupt (INTVDCE0GR3VLINE)	√	203	10CB			$+32C_H$	0 to 15 (15)	
	End of field for record function at Scaler 0 interrupt (INTVDCE0S0VFIELD)	√	204	10CC			$+330_H$	0 to 15 (15)	
	Reserved	√	205	10CD			$+334_H$	0 to 15 (15)	
	VSYNC output at Scaler 1 interrupt (INTVDCE0S1LOVSYNC)	√	206	10CE			$+338_H$	0 to 15 (15)	
	Reserved	√	207	10CF			$+33C_H$	0 to 15 (15)	
	VSYNC input at Output Image Render interrupt (INTVDCE0OIRVIVSYNC)	√	208	10D0			$+340_H$	0 to 15 (15)	
	VSYNC output at Output Image Render interrupt (INTVDCE0OIRLOVSYNC)	√	209	10D1			$+344_H$	0 to 15 (15)	
	Line detection of VSYNC at Output Image Render interrupt (INTVDCE0OIRVLINE)	√	210	10D2			$+348_H$	0 to 15 (15)	
–	Reserved	√	211	10D3			$+34C_H$	0 to 15 (15)	
–	Reserved		212	10D4			$+350_H$	0 to 15 (15)	
–	Reserved		213	10D5			$+354_H$	0 to 15 (15)	
–	Reserved		214	10D6			$+358_H$	0 to 15 (15)	
RLIN32	Status interrupt interrupt (INTRLIN32UR2)		215	10D7			$+35C_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN32UR1)		216	10D8			$+360_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN32UR0)		217	10D9			$+364_H$	0 to 15 (15)	
RLIN33	Status interrupt (INTRLIN33UR2)		218	10DA			$+368_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN33UR1)		219	10DB			$+36C_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN33UR0)		220	10DC			$+370_H$	0 to 15 (15)	

Table 7.12 D1M1A Interrupt Exception Handler and Priority (7/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
VDCE1	Error interrupt (INTVDCE1ERR)	√	221	10DD	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+374_H$	0 to 15 (15)	High
	VBLANK detection at Graphics 3 (INTVDCE1GR3VBLANK)		222	10DE			$+378_H$	0 to 15 (15)	
	VSYNC input at Scaler 0 interrupt (INTVDCE1S0VIVSYNC)	√	223	10DF			$+37C_H$	0 to 15 (15)	
	VSYNC output at Scaler 0 interrupt (INTVDCE1S0LOVSYNC)	√	224	10E0			$+380_H$	0 to 15 (15)	
	Line detection of designated line at Graphics3 interrupt (INTVDCE1GR3VLINE)	√	225	10E1			$+384_H$	0 to 15 (15)	
	End of field for record function at Scaler 0 interrupt (INTVDCE1S0VFIELD)	√	226	10E2			$+388_H$	0 to 15 (15)	
	Line detection of designated line at Scalar 0 interrupt (INTVDCE1S0WLINE)	√	227	10E3			$+38C_H$	0 to 15 (15)	
	VSYNC output at Scaler 1 interrupt (INTVDCE1S1LOVSYNC)	√	228	10E4			$+390_H$	0 to 15 (15)	
–	Reserved	√	229	10E5			$+394_H$	0 to 15 (15)	
–	Reserved	√	230	10E6			$+398_H$	0 to 15 (15)	
–	Reserved	√	231	10E7			$+39C_H$	0 to 15 (15)	
–	Reserved	√	232	10E8			$+3A0_H$	0 to 15 (15)	
–	Reserved	√	233	10E9			$+3A4_H$	0 to 15 (15)	
–	Reserved	√	234	10EA			$+3A8_H$	0 to 15 (15)	
–	Reserved		235	10EB			$+3AC_H$	0 to 15 (15)	
–	Reserved		236	10EC			$+3B0_H$	0 to 15 (15)	
–	Reserved		237	10ED			$+3B4_H$	0 to 15 (15)	
–	Reserved		238	10EE			$+3B8_H$	0 to 15 (15)	
–	Reserved		239	10EF			$+3BC_H$	0 to 15 (15)	
VOWE	Common interrupt (INTVOWE0)	√	240	10F0			$+3C0_H$	0 to 15 (15)	
–	Reserved		241	10F1			$+3C4_H$	0 to 15 (15)	
JCUA	Compression/decompression process interrupt (INTJCUA0EDI)	√	242	10F2			$+3C8_H$	0 to 15 (15)	
	Data transfer interrupt (INTJCUA0DTI)		243	10F3			$+3CC_H$	0 to 15 (15)	
ETNB0	Interrupt request line 0 (INTETNB0LINE0)	√	244	10F4			$+3D0_H$	0 to 15 (15)	
	Interrupt request line 1 (INTETNB0LINE1)	√	245	10F5			$+3D4_H$	0 to 15 (15)	
	Interrupt request line 2 (INTETNB0LINE2)	√	246	10F6			$+3D8_H$	0 to 15 (15)	
	Interrupt request line 3 (INTETNB0LINE3)	√	247	10F7			$+3DC_H$	0 to 15 (15)	
–	Reserved		248	10F8			$+3E0_H$	0 to 15 (15)	
–	Reserved		249	10F9			$+3E4_H$	0 to 15 (15)	
–	Reserved		250	10FA			$+3E8_H$	0 to 15 (15)	
–	Reserved		251	10FB			$+3EC_H$	0 to 15 (15)	
–	Reserved	√	252	10FC			$+3F0_H$	0 to 15 (15)	
–	Reserved		253	10FD			$+3F4_H$	0 to 15 (15)	
–	Reserved	√	254	10FE			$+3F8_H$	0 to 15 (15)	
Internal bus errors	This interrupt is a logical OR combination of various internal bus error indications. The exact cause of this error interrupt can be evaluated by use of the bus error status registers. Refer to Section 7.4.9, Bus error status registers for details.		255	10FF			$+3FC_H$	0 to 15 (15)	

Note 1. For a level interrupt, the status register in each module should be cleared during interrupt processing by software. The EICTn bit in EICn is set to 1. The EIRFn bit in EICn cannot be cleared by software.

7.4.8 D1M1-V2 Interrupt Exception Handler and Priority

Table 7.13 D1M1-V2 Interrupt Exception Handler and Priority (1/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
Non-maskable interrupt			(FENMI)	$E0_H$	$+0E0_H$	$+0E0_H$	—		High
FE level interrupt			(FEINT)	$F0_H$	$+0F0_H$	$+0F0_H$	—		
WDTA0	75 % interrupt (INTWDTA0)		0	1000	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+000_H$	0 to 15 (15)	
WDTA1	75 % interrupt (INTWDTA1)		1	1001			$+004_H$	0 to 15 (15)	
RTCA0	1 second interrupt (INTRTCA01S)		2	1002			$+008_H$	0 to 15 (15)	
	Alarm interrupt (INTRTCA0AL)		3	1003			$+00C_H$	0 to 15 (15)	
	Fixed interval interrupt (INTRTCA0R)		4	1004			$+010_H$	0 to 15 (15)	
AWOT0	Timer interrupt (INTAWOT0)		5	1005			$+014_H$	0 to 15 (15)	
Port	External interrupt 0 (INTP0)		6	1006			$+018_H$	0 to 15 (15)	
	External interrupt 1 (INTP1)		7	1007			$+01C_H$	0 to 15 (15)	
	External interrupt 2 (INTP2)		8	1008			$+020_H$	0 to 15 (15)	
	External interrupt 3 (INTP3)		9	1009			$+024_H$	0 to 15 (15)	
	External interrupt 4 (INTP4)		10	100A			$+028_H$	0 to 15 (15)	
	External interrupt 5 (INTP5)		11	100B			$+02C_H$	0 to 15 (15)	
	External interrupt 6 (INTP6)		12	100C			$+030_H$	0 to 15 (15)	
	External interrupt 7 (INTP7)		13	100D			$+034_H$	0 to 15 (15)	
	External interrupt 8 (INTP8)		14	100E			$+038_H$	0 to 15 (15)	
	External interrupt 9 (INTP9)		15	100F			$+03C_H$	0 to 15 (15)	
	External interrupt 10 (INTP10)		16	1010			$+040_H$	0 to 15 (15)	
TAUB0	Channel 0 interrupt (INTTAUB0I0)		17	1011			$+044_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB0I1)		18	1012			$+048_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB0I2)		19	1013			$+04C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB0I3)		20	1014			$+050_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB0I4)		21	1015			$+054_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB0I5)		22	1016			$+058_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB0I6)		23	1017			$+05C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB0I7)		24	1018			$+060_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB0I8)		25	1019			$+064_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB0I9)		26	101A			$+068_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB0I10)		27	101B			$+06C_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB0I11)		28	101C			$+070_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB0I12)		29	101D			$+074_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB0I13)		30	101E			$+078_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB0I14)		31	101F			$+07C_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB0I15)		32	1020			$+080_H$	0 to 15 (15)	

Table 7.13 D1M1-V2 Interrupt Exception Handler and Priority (2/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
TAUB1	Channel 0 interrupt (INTTAUB1I0)		33	1021	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+084_H$	0 to 15 (15)	High
	Channel 1 interrupt (INTTAUB1I1)		34	1022			$+088_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB1I2)		35	1023			$+08C_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB1I3)		36	1024			$+090_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB1I4)		37	1025			$+094_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB1I5)		38	1026			$+098_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB1I6)		39	1027			$+09C_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB1I7)		40	1028			$+0A0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB1I8)		41	1029			$+0A4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB1I9)		42	102A			$+0A8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB1I10)		43	102B			$+0AC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB1I11)		44	102C			$+0B0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB1I12)		45	102D			$+0B4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB1I13)		46	102E			$+0B8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB1I14)		47	102F			$+0BC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB1I15)		48	1030			$+0C0_H$	0 to 15 (15)	
TAUB2	Channel 0 interrupt (INTTAUB2I0)		49	1031			$+0C4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUB2I1)		50	1032			$+0C8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUB2I2)		51	1033			$+0CC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUB2I3)		52	1034			$+0D0_H$	0 to 15 (15)	
	Channel 4 interrupt (INTTAUB2I4)		53	1035			$+0D4_H$	0 to 15 (15)	
	Channel 5 interrupt (INTTAUB2I5)		54	1036			$+0D8_H$	0 to 15 (15)	
	Channel 6 interrupt (INTTAUB2I6)		55	1037			$+0DC_H$	0 to 15 (15)	
	Channel 7 interrupt (INTTAUB2I7)		56	1038			$+0E0_H$	0 to 15 (15)	
	Channel 8 interrupt (INTTAUB2I8)		57	1039			$+0E4_H$	0 to 15 (15)	
	Channel 9 interrupt (INTTAUB2I9)		58	103A			$+0E8_H$	0 to 15 (15)	
	Channel 10 interrupt (INTTAUB2I10)		59	103B			$+0EC_H$	0 to 15 (15)	
	Channel 11 interrupt (INTTAUB2I11)		60	103C			$+0F0_H$	0 to 15 (15)	
	Channel 12 interrupt (INTTAUB2I12)		61	103D			$+0F4_H$	0 to 15 (15)	
	Channel 13 interrupt (INTTAUB2I13)		62	103E			$+0F8_H$	0 to 15 (15)	
	Channel 14 interrupt (INTTAUB2I14)		63	103F			$+0FC_H$	0 to 15 (15)	
	Channel 15 interrupt (INTTAUB2I15)		64	1040			$+100_H$	0 to 15 (15)	
ADCE0	Temperature Sensor interrupt (INTADCE0TSN)		65	1041			$+104_H$	0 to 15 (15)	
	Scan group 1 interrupt (INTADCE0I1)		66	1042			$+108_H$	0 to 15 (15)	
	Scan group 2 interrupt (INTADCE0I2)		67	1043			$+10C_H$	0 to 15 (15)	
	Scan group 3 interrupt (INTADCE0I3)		68	1044			$+110_H$	0 to 15 (15)	
RSCAN	Channel 0 to 2 global error interrupt (INTRCANGERR)	√	69	1045			$+114_H$	0 to 15 (15)	
	Channel 0 to 2 RX FIFO interrupt (INTRCANGRECC)	√	70	1046			$+118_H$	0 to 15 (15)	
	Channel 0 error interrupt (INTRCAN0ERR)	√	71	1047			$+11C_H$	0 to 15 (15)	
	Channel 0 COM RX FIFO interrupt (INTRCAN0REC)	√	72	1048			$+120_H$	0 to 15 (15)	
	Channel 0 TX interrupt (INTRCAN0TRX)	√	73	1049			$+124_H$	0 to 15 (15)	
CSIG0	Reception error interrupt (INTCSIGT0IRE)		74	104A			$+128_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG0IR)		75	104B			$+12C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG0IC)		76	104C			$+130_H$	0 to 15 (15)	
CSIH0	Reception error interrupt (INTCSIH0IRE)		77	104D			$+134_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH0IR)		78	104E			$+138_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH0IC)		79	104F			$+13C_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH0IJC)		80	1050			$+140_H$	0 to 15 (15)	

Table 7.13 D1M1-V2 Interrupt Exception Handler and Priority (3/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RLIN30	Status interrupt (INTRLIN30UR2)		81	1051	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+144_H$	0 to 15 (15)	High
	Reception complete interrupt (INTRLIN30UR1)		82	1052			$+148_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN30UR0)		83	1053			$+14C_H$	0 to 15 (15)	
RLIN31	Status interrupt (INTRLIN31UR2)		84	1054			$+150_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN31UR1)		85	1055			$+154_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN31UR0)		86	1056			$+158_H$	0 to 15 (15)	
SG0	Threshold interrupt (INTSG0TI)		87	1057			$+15C_H$	0 to 15 (15)	
SG1	Threshold interrupt (INTSG1TI)		88	1058			$+160_H$	0 to 15 (15)	
SG2	Threshold interrupt (INTSG2TI)		89	1059			$+164_H$	0 to 15 15)	
SG3	Threshold interrupt (INTSG3TI)		90	105A			$+168_H$	0 to 15 (15)	
SG4	Threshold interrupt (INTSG4TI)		91	105B			$+16C_H$	0 to 15 (15)	
OCD/DCU	Debugger hot-attach interrupt (INTDCUTDI)	√	92	105C			$+170_H$	0 to 15 (15)	
DMAC	DMA Transfer Error (INTDMAERR)		93	105D			$+174_H$	0 to 15 (15)	
	DMA channel 0 transfer completion/transfer count compare match interrupt (INTDMA0)		94	105E			$+178_H$	0 to 15 (15)	
	DMA channel 1 transfer completion/transfer count compare match interrupt (INTDMA1)		95	105F			$+17C_H$	0 to 15 (15)	
	DMA channel 2 transfer completion/transfer count compare match interrupt (INTDMA2)		96	1060			$+180_H$	0 to 15 (15)	
	DMA channel 3 transfer completion/transfer count compare match interrupt (INTDMA3)		97	1061			$+184_H$	0 to 15 (15)	
	DMA channel 4 transfer completion/transfer count compare match interrupt (INTDMA4)		98	1062			$+188_H$	0 to 15 (15)	
	DMA channel 5 transfer completion/transfer count compare match interrupt (INTDMA5)		99	1063			$+18C_H$	0 to 15 (15)	
	DMA channel 6 transfer completion/transfer count compare match interrupt (INTDMA6)		100	1064			$+190_H$	0 to 15 (15)	
	DMA channel 7 transfer completion/transfer count compare match interrupt (INTDMA7)		101	1065			$+194_H$	0 to 15 (15)	
	DMA channel 8 transfer completion/transfer count compare match interrupt (INTDMA8)		102	1066			$+198_H$	0 to 15 (15)	
	DMA channel 9 transfer completion/transfer count compare match interrupt (INTDMA9)		103	1067			$+19C_H$	0 to 15 (15)	
	DMA channel 10 transfer completion/transfer count compare match interrupt (INTDMA10)		104	1068			$+1A0_H$	0 to 15 (15)	
	DMA channel 11 transfer completion/transfer count compare match interrupt (INTDMA11)		105	1069			$+1A4_H$	0 to 15 (15)	
	DMA channel 12 transfer completion/transfer count compare match interrupt (INTDMA12)		106	106A			$+1A8_H$	0 to 15 (15)	
	DMA channel 13 transfer completion/transfer count compare match interrupt (INTDMA13)		107	106B			$+1AC_H$	0 to 15 (15)	
	DMA channel 14 transfer completion/transfer count compare match interrupt (INTDMA14)		108	106C			$+1B0_H$	0 to 15 (15)	
	DMA channel 15 transfer completion/transfer count compare match interrupt (INTDMA15)		109	106D			$+1B4_H$	0 to 15 (15)	
RIIC0	Receive error/event interrupt (INTRIIC0LEE)	√	110	106E			$+1B8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC0RI)		111	106F			$+1BC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC0TI)		112	1070			$+1C0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC0TEI)	√	113	1071			$+1C4_H$	0 to 15 (15)	
RIIC1	Receive error/event interrupt (INTRIIC1LEE)	√	114	1072			$+1C8_H$	0 to 15 (15)	
	Data received interrupt (INTRIIC1RI)		115	1073			$+1CC_H$	0 to 15 (15)	
	Data buffer empty interrupt (INTRIIC1TI)		116	1074			$+1D0_H$	0 to 15 (15)	
	Data transmitted interrupt (INTRIIC1TEI)	√	117	1075			$+1D4_H$	0 to 15 (15)	

Table 7.13 D1M1-V2 Interrupt Exception Handler and Priority (4/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
RSCAN1	Channel 1 error interrupt (INTRCAN1ERR)	√	118	1076	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+1D8_H$	0 to 15 (15)	High
	Channel 1 COM RX FIFO interrupt (INTRCAN1REC)	√	119	1077			$+1DC_H$	0 to 15 (15)	
	Channel 1 TX interrupt (INTRCAN1TRX)	√	120	1078			$+1E0_H$	0 to 15 (15)	
TAUJ0	Channel 0 interrupt (INTTAUJ0I0)		121	1079			$+1E4_H$	0 to 15 (15)	
	Channel 1 interrupt (INTTAUJ0I1)		122	107A			$+1E8_H$	0 to 15 (15)	
	Channel 2 interrupt (INTTAUJ0I2)		123	107B			$+1EC_H$	0 to 15 (15)	
	Channel 3 interrupt (INTTAUJ0I3)		124	107C			$+1F0_H$	0 to 15 (15)	
OSTM0	Timer interrupt (INTOSTM0)		125	107D			$+1F4_H$	0 to 15 (15)	
OSTM1	Timer interrupt (INTOSTM1)		126	107E			$+1F8_H$	0 to 15 (15)	
CSIG1	Reception error interrupt (INTCSIGT1IRE)		127	107F			$+1FC_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG1IR)		128	1080			$+200_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG1IC)		129	1081			$+204_H$	0 to 15 15)	
CSIG2	Reception error interrupt (INTCSIGT2IRE)		130	1082			$+208_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG2IR)		131	1083			$+20C_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG2IC)		132	1084			$+210_H$	0 to 15 (15)	
CSIG3	Reception error interrupt (INTCSIGT3IRE)		133	1085			$+214_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIG3IR)		134	1086			$+218_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIG3IC)		135	1087			$+21C_H$	0 to 15 (15)	
CSIH1	Reception error interrupt (INTCSIH1IRE)		136	1088			$+220_H$	0 to 15 (15)	
	Reception status interrupt (INTCSIH1IR)		137	1089			$+224_H$	0 to 15 (15)	
	Communication status interrupt (INTCSIH1IC)		138	108A			$+228_H$	0 to 15 (15)	
	Job completion interrupt (INTCSIH1IJC)		139	108B			$+22C_H$	0 to 15 (15)	

Table 7.13 D1M1-V2 Interrupt Exception Handler and Priority (5/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
PWM Generators and Diagnostic	A/D Converter trigger queue full interrupt (INTQFULL)		140	108C	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+230_H$	0 to 15 (15)	High
	PWM Generator channel 0 interrupt (INTPWGA0)		141	108D			$+234_H$	0 to 15 (15)	
	PWM Generator channel 1 interrupt (INTPWGA1)		142	108E			$+238_H$	0 to 15 (15)	
	PWM Generator channel 2 interrupt (INTPWGA2)		143	108F			$+23C_H$	0 to 15 (15)	
	PWM Generator channel 3 interrupt (INTPWGA3)		144	1090			$+240_H$	0 to 15 (15)	
	PWM Generator channel 4 interrupt (INTPWGA4)		145	1091			$+244_H$	0 to 15 (15)	
	PWM Generator channel 5 interrupt (INTPWGA5)		146	1092			$+248_H$	0 to 15 (15)	
	PWM Generator channel 6 interrupt (INTPWGA6)		147	1093			$+24C_H$	0 to 15 (15)	
	PWM Generator channel 7 interrupt (INTPWGA7)		148	1094			$+250_H$	0 to 15 (15)	
	PWM Generator channel 8 interrupt (INTPWGA8)		149	1095			$+254_H$	0 to 15 (15)	
	PWM Generator channel 9 interrupt (INTPWGA9)		150	1096			$+258_H$	0 to 15 (15)	
	PWM Generator channel 10 interrupt (INTPWGA10)		151	1097			$+25C_H$	0 to 15 (15)	
	PWM Generator channel 11 interrupt (INTPWGA11)		152	1098			$+260_H$	0 to 15 (15)	
	PWM Generator channel 12 interrupt (INTPWGA12)		153	1099			$+264_H$	0 to 15 (15)	
	PWM Generator channel 13 interrupt (INTPWGA13)		154	109A			$+268_H$	0 to 15 (15)	
	PWM Generator channel 14 interrupt (INTPWGA14)		155	109B			$+26C_H$	0 to 15 (15)	
	PWM Generator channel 15 interrupt (INTPWGA15)		156	109C			$+270_H$	0 to 15 (15)	
	PWM Generator channel 16 interrupt (INTPWGA16)		157	109D			$+274_H$	0 to 15 (15)	
	PWM Generator channel 17 interrupt (INTPWGA17)		158	109E			$+278_H$	0 to 15 (15)	
	PWM Generator channel 18 interrupt (INTPWGA18)		159	109F			$+27C_H$	0 to 15 (15)	
	PWM Generator channel 19 interrupt (INTPWGA19)		160	10A0			$+280_H$	0 to 15 (15)	
	PWM Generator channel 20 interrupt (INTPWGA20)		161	10A1			$+284_H$	0 to 15 (15)	
	PWM Generator channel 21 interrupt (INTPWGA21)		162	10A2			$+288_H$	0 to 15 (15)	
	PWM Generator channel 22 interrupt (INTPWGA22)		163	10A3			$+28C_H$	0 to 15 (15)	
	PWM Generator channel 23 interrupt (INTPWGA23)		164	10A4			$+290_H$	0 to 15 (15)	
ICU-S2	CMD registers ready to write interrupt (INTICUSTWRDY)		165	10A5			$+294_H$	0 to 15 (15)	
	CMD registers ready to read interrupt (INTICUSTRRDY)		166	10A6			$+298_H$	0 to 15 (15)	
–	Reserved	√	167	10A7			$+29C_H$	0 to 15 (15)	
–	Reserved		168	10A8			$+2A0_H$	0 to 15 (15)	
–	Reserved		169	10A9			$+2A4_H$	0 to 15 (15)	
–	Reserved	√	170	10AA			$+2A8_H$	0 to 15 (15)	
Flash control	Flash access error (INTFLERR)	√	171	10AB			$+2AC_H$	0 to 15 (15)	
Flash control	Flash sequencer ready (INTFLENDNM)		172	10AC			$+2B0_H$	0 to 15 (15)	
–	Reserved	√	173	10AD			$+2B4_H$	0 to 15 (15)	
LCBI0	Read data ready interrupt (INTLCBI0RDY)		174	10AE			$+2B8_H$	0 to 15 (15)	
	Write buffer empty interrupt (INTLCBI0EMPTY)		175	10AF			$+2BC_H$	0 to 15 (15)	
	Write buffer half full interrupt (INTLCBI0HALF)		176	10B0			$+2C0_H$	0 to 15 (15)	
	Write buffer full interrupt (INTLCBI0FULL)		177	10B1			$+2C4_H$	0 to 15 (15)	
	Write buffer quarter full interrupt (INTLCBI0QTR)		178	10B2			$+2C8_H$	0 to 15 (15)	
	Write buffer three quarters full interrupt (INTLCBI03QTR)		179	10B3			$+2CC_H$	0 to 15 (15)	
RSCAN2	Channel 2 error interrupt (INTRCAN2ERR)	√	180	10B4			$+2D0_H$	0 to 15 (15)	
	Channel 2 COM RX FIFO interrupt (INTRCAN2REC)	√	181	10B5			$+2D4_H$	0 to 15 (15)	
	Channel 2 TX interrupt (INTRCAN2TRX)	√	182	10B6			$+2D8_H$	0 to 15 (15)	
PCMP0	FIFO buffer fill interrupt (INTPCMP0FFIL)		183	10B7			$+2DC_H$	0 to 15 (15)	
	Error interrupt (INTPCMP0FERR)		184	10B8			$+2E0_H$	0 to 15 (15)	

Table 7.13 D1M1-V2 Interrupt Exception Handler and Priority (6/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
ISM0	Target position reached interrupt (INTISM0REACHED)		185	10B9	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+2E4_H$	0 to 15 (15)	High
	Idle state interrupt (INTISM0DONE)		186	10BA			$+2E8_H$	0 to 15 (15)	
	ZPD measurement started interrupt (INTISM0ZPDAD)		187	10BB			$+2EC_H$	0 to 15 (15)	
–	Reserved	√	188	10BC			$+2F0_H$	0 to 15 (15)	
SSIF0	Multi-purpose interrupt (INTSSIF0)	√	189	10BD			$+2F4_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF0RX)	√	190	10BE			$+2F8_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF0TX)	√	191	10BF			$+2FC_H$	0 to 15 (15)	
SSIF1	Multi-purpose interrupt (INTSSIF1)	√	192	10C0			$+300_H$	0 to 15 (15)	
	Reception data full interrupt (INTSSIF1RX)	√	193	10C1			$+304_H$	0 to 15 (15)	
	Transmission data empty interrupt (INTSSIF1TX)	√	194	10C2			$+308_H$	0 to 15 (15)	
–	Reserved	√	195	10C3			$+30C_H$	0 to 15 (15)	
GPU2D	Pause interrupt (INTGPU2D0PAUSE)	√	196	10C4			$+310_H$	0 to 15 (15)	
	SYNC interrupt (INTGPU2D0SYNC)	√	197	10C5			$+314_H$	0 to 15 (15)	
	Stop / Stall / MBI Error interrupt (INTGPU2D0SP)	√	198	10C6			$+318_H$	0 to 15 (15)	
VDCE0	Error interrupt (INTVDCE0ERR)	√	199	10C7			$+31C_H$	0 to 15 (15)	
	VBLANK detection at Graphics 3 (INTVDCE0GR3VBLANK)		200	10C8			$+320_H$	0 to 15 (15)	
	VSYNC input at Scaler 0 interrupt (INTVDCE0S0VIVSYNC)	√	201	10C9			$+324_H$	0 to 15 (15)	
	VSYNC output at Scaler 0 interrupt (INTVDCE0S0LOVSYNC)	√	202	10CA			$+328_H$	0 to 15 (15)	
	Line detection of designated line at Graphics3 interrupt (INTVDCE0GR3VLINE)	√	203	10CB			$+32C_H$	0 to 15 (15)	
	End of field for record function at Scaler 0 interrupt (INTVDCE0S0VFIELD)	√	204	10CC			$+330_H$	0 to 15 (15)	
	Reserved	√	205	10CD			$+334_H$	0 to 15 (15)	
	VSYNC output at Scaler 1 interrupt (INTVDCE0S1LOVSYNC)	√	206	10CE			$+338_H$	0 to 15 (15)	
	Reserved	√	207	10CF			$+33C_H$	0 to 15 (15)	
	VSYNC input at Output Image Render interrupt (INTVDCE0OIRVIVSYNC)	√	208	10D0			$+340_H$	0 to 15 (15)	
	VSYNC output at Output Image Render interrupt (INTVDCE0OIRLOVSYNC)	√	209	10D1			$+344_H$	0 to 15 (15)	
	Line detection of VSYNC at Output Image Render interrupt (INTVDCE0OIRVLINE)	√	210	10D2			$+348_H$	0 to 15 (15)	
–	Reserved	√	211	10D3			$+34C_H$	0 to 15 (15)	
–	Reserved		212	10D4			$+350_H$	0 to 15 (15)	
–	Reserved		213	10D5			$+354_H$	0 to 15 (15)	
–	Reserved		214	10D6			$+358_H$	0 to 15 (15)	
RLIN32	Status interrupt interrupt (INTRLIN32UR2)		215	10D7			$+35C_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN32UR1)		216	10D8			$+360_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN32UR0)		217	10D9			$+364_H$	0 to 15 (15)	
RLIN33	Status interrupt (INTRLIN33UR2)		218	10DA			$+368_H$	0 to 15 (15)	
	Reception complete interrupt (INTRLIN33UR1)		219	10DB			$+36C_H$	0 to 15 (15)	
	Transmission interrupt (INTRLIN33UR0)		220	10DC			$+370_H$	0 to 15 (15)	

Table 7.13 D1M1-V2 Interrupt Exception Handler and Priority (7/7)

Functional Module	Interrupt Source Name	Level Interrupt *1	Number of Interrupt Channel (EIINT)	Source Code	Offset Address: Direct Branch RINT = 0	Offset Address: Direct Branch RINT = 1	Offset Address: Table Reference	Interrupt Priority (Initial Value)	Default Priority
–	Reserved	√	221	10DD	The offset address is the same in all channels and is determined between $+100_H$ and $+1F0_H$ according to priority.	The offset address is always $+100_H$ regardless of priority to reduce the offset address.	$+374_H$	0 to 15 (15)	High
–	Reserved		222	10DE			$+378_H$	0 to 15 (15)	
–	Reserved	√	223	10DF			$+37C_H$	0 to 15 (15)	
–	Reserved	√	224	10E0			$+380_H$	0 to 15 (15)	
–	Reserved	√	225	10E1			$+384_H$	0 to 15 (15)	
–	Reserved	√	226	10E2			$+388_H$	0 to 15 (15)	
–	Reserved	√	227	10E3			$+38C_H$	0 to 15 (15)	
–	Reserved	√	228	10E4			$+390_H$	0 to 15 (15)	
–	Reserved	√	229	10E5			$+394_H$	0 to 15 (15)	
–	Reserved	√	230	10E6			$+398_H$	0 to 15 (15)	
–	Reserved	√	231	10E7			$+39C_H$	0 to 15 (15)	
–	Reserved	√	232	10E8			$+3A0_H$	0 to 15 (15)	
–	Reserved	√	233	10E9			$+3A4_H$	0 to 15 (15)	
–	Reserved		234	10EA			$+3A8_H$	0 to 15 (15)	
–	Reserved		235	10EB			$+3AC_H$	0 to 15 (15)	
–	Reserved		236	10EC			$+3B0_H$	0 to 15 (15)	
–	Reserved		237	10ED			$+3B4_H$	0 to 15 (15)	
–	Reserved		238	10EE			$+3B8_H$	0 to 15 (15)	
–	Reserved		239	10EF			$+3BC_H$	0 to 15 (15)	
VOWE	Common interrupt (INTVOWE0)	√	240	10F0			$+3C0_H$	0 to 15 (15)	
–	Reserved		241	10F1			$+3C4_H$	0 to 15 (15)	
JCUA	Compression/decompression process interrupt (INTJCUA0EDI)	√	242	10F2			$+3C8_H$	0 to 15 (15)	
	Data transfer interrupt (INTJCUA0DTI)		243	10F3			$+3CC_H$	0 to 15 (15)	
ETNB0	Interrupt request line 0 (INTETNB0LINE0)	√	244	10F4			$+3D0_H$	0 to 15 (15)	
	Interrupt request line 1 (INTETNB0LINE1)	√	245	10F5			$+3D4_H$	0 to 15 (15)	
	Interrupt request line 2 (INTETNB0LINE2)	√	246	10F6			$+3D8_H$	0 to 15 (15)	
	Interrupt request line 3 (INTETNB0LINE3)	√	247	10F7			$+3DC_H$	0 to 15 (15)	
–	Reserved		248	10F8			$+3E0_H$	0 to 15 (15)	
–	Reserved		249	10F9			$+3E4_H$	0 to 15 (15)	
–	Reserved		250	10FA			$+3E8_H$	0 to 15 (15)	
–	Reserved		251	10FB			$+3EC_H$	0 to 15 (15)	
–	Reserved	√	252	10FC			$+3F0_H$	0 to 15 (15)	
–	Reserved		253	10FD			$+3F4_H$	0 to 15 (15)	
–	Reserved	√	254	10FE			$+3F8_H$	0 to 15 (15)	
Internal bus errors	This interrupt is a logical OR combination of various internal bus error indications. The exact cause of this error interrupt can be evaluated by use of the bus error status registers. Refer to Section 7.4.9, Bus error status registers for details.		255	10FF			$+3FC_H$	0 to 15 (15)	

Note 1. For a level interrupt, the status register in each module should be cleared during interrupt processing by software. The EICTn bit in EICn is set to 1. The EIRFn bit in EICn cannot be cleared by software.

7.4.9 Bus error status registers

7.4.9.1 BERR0ST0 — Bus error status register 0

This register holds error flags, that indicated the cause of an assertion of the internal bus error interrupt (EIINT channel number 255).
Other bus errors are indicated in the bus error staus register 1 BERR0ST1.

The error flags of this register can be cleared by use of the bus error status clear register 0 BERR0STC0.

Access: This register can be accessed in 32-bit units.
Address: FFC0 7000_H
Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	BERR0 SSF018	0	BERR0 SSF016
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	BERR0 SSF008	BERR0 SSF007	BERR0 SSF006	BERR0 SSF005	BERR0 SSF004	BERR0 SSF003	BERR0 SSF002	BERR0 SSF001	BERR0 SSF000
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

All error flags in this register indicate the following:

- BERR0SSF0x = 0: respective error has not occured
- BERR0SSF0x = 1: respective error has occured

Table 7.14 BERR0ST0 register contents

Bit position	Bit name	Function
31 to 19	Reserved	Reading returns 0. When written, write the initial value.
18	BERR0SSF018	HBUS Error (BRG_GFX)
17	Reserved	Reading returns 0. When written, write the initial value.
16	BERR0SSF016	Media Local Bus module (MLBB) cross-connect error An error occurred during an MLBB transaction via the XC1 cross-connect.
15 to 9	Reserved	Reading returns 0. When written, write the initial value.
8	BERR0SSF008	PBUS Slave Error (PBUSBUSIP_EEP)
7	BERR0SSF007	PBUS Slave Error (PBUSBUSIP3)
6	BERR0SSF006	PBUS Slave Error (PBUSBUSIP2)
5	BERR0SSF005	PBUS Slave Error (PBUSBUSIP1)
4	BERR0SSF004	PBUS Slave Error (PBUSBUSIP_SYS)
3	BERR0SSF003	PBUS Slave Error (PBUSBUSIP_SN)
2	BERR0SSF002	PBUS Slave Error (PBUSBUSIP_COM)
1	BERR0SSF001	PBUS Slave Error (PBUSBUSIP_PWM)
0	BERR0SSF000	PBUS Slave Error (PBUSBUSIP_AWO)

7.4.9.2 BERR0ST1 — Bus error status register 1

This register holds error flags, that indicated the cause of an assertion of the internal bus error interrupt (EIINT channel number 255).
Other bus errors are indicated in the bus error staus register 0 BERR0ST0.

The error flags of this register can be cleared by use of the bus error status clear register 1 BERR0STC1.

Access: This register can be accessed in 32-bit units.

Address: FFC0 7004_H

Initial value: 0000 00xx 0000 0000 0000 0000 0000 0000_B

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BERR0 SSF131	BERR0 SSF130	BERR0 SSF129	BERR0 SSF128	0	0	x*1	x*1	0	0	0	0	0	0	0	BERR0 SSF116
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	BERR0 SSF112	BERR0 SSF111	BERR0 SSF110	BERR0 SSF09	BERR0 SSF108	0	0	BERR0 SSF105	BERR0 SSF104	BERR0 SSF103	BERR0 SSF102	BERR0 SSF101	BERR0 SSF100
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The value of this bit is undefined.

All error flags in this register indicate the following:

– BERR0SSF1x = 0: respective error has not occured

– BERR0SSF1x = 1: respective error has occured

Table 7.15 BERR0ST1 register contents (1/2)

Bit position	Bit name	Function
31	BERR0SSF131	Video Output Warping Engine (VOWE) cross-connect transaction protocol error BERR0SSF131 write transaction error BERR0SSF130 read transaction error
30	BERR0SSF130	
29	BERR0SSF129	JPEG Codec Unit (JCUA) cross-connect transaction protocol error BERR0SSF129 write transaction error BERR0SSF128 read transaction error
28	BERR0SSF128	
27 to 17	Reserved	Reading returns 0. When written, write the initial value.
16	BERR0SSF116	CPU Subssytem cross-connect error during • XC1 cross-connect to Global Interconnect (GVCI) transaction • Global Interconnect (GVCI) to XC1 cross-connect transaction • Global Interconnect (GVCI) to PBUS Interconnect (VPI)
15 to 13	Reserved	Reading returns 0. When written, write the initial value.
12	BERR0SSF112	Video Data Controller 1 (VDCE1) cross-connect transaction protocol error BERR0SSF112 : Image Synthesizer13 read transaction error BERR0SSF111 : Image Synthesizer11 read transaction error BERR0SSF110 : Image Synthesizer12 read transaction error BERR0SSF109 : Video Input1 write transaction error BERR0SSF108 : Image Synthesizer10 read transaction error
11	BERR0SSF111	
10	BERR0SSF110	
9	BERR0SSF109	
8	BERR0SSF108	
7 to 6	Reserved	Reading returns 0. When written, write the initial value.

Table 7.15 BERR0ST1 register contents (2/2)

Bit position	Bit name	Function
5	BERR0SSF105	Video Data Controller 0 (VDCE0) cross-connect transaction protocol error BERR0SSF105 : Output image generator read transaction error BERR0SSF104 : Image Synthesizer03 read transaction error BERR0SSF103 : Image Synthesizer01 read transaction error BERR0SSF102 : Image Synthesizer02 read transaction error BERR0SSF101 : Video Input0 write transaction error BERR0SSF100 : Image Synthesizer00 read transaction error
4	BERR0SSF104	
3	BERR0SSF103	
2	BERR0SSF102	
1	BERR0SSF101	
0	BERR0SSF100	

7.4.9.3 BERR0STC0 — Bus error status clear register 0

This register is used to clear the bus error flags of BERR0ST0.

Access: This register can be accessed in 32-bit units.

Address: FFC0 7010_H

Initial value: Reading this register returns always 0000 0000_H.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	BERR0 CL018	0	BERR0 CL016
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	BERR0 CL008	BERR0 CL007	BERR0 CL006	BERR0 CL005	BERR0 CL004	BERR0 CL003	BERR0 CL002	BERR0 CL001	BERR0 CL000
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

All clear bits in this register effect the following, when written:

- BERR0CL0x = 0: no function
- BERR0CL0x = 1: respective error flag in the bus error status register 0 is cleared

Table 7.16 BERR0STC0 register contents

Bit position	Bit name	Function
31 to 19	Reserved	Reading returns 0. When written, write the initial value.
18	BERR0CL018	BERR0ST0.BERR0SSF018 error flag clear
17	Reserved	Reading returns 0. When written, write the initial value.
16	BERR0CL016	BERR0ST0.BERR0SSF016 error flag clear
15 to 9	Reserved	Reading returns 0. When written, write the initial value.
8	BERR0CL008	BERR0ST0.BERR0SSF008 error flag clear
7	BERR0CL007	BERR0ST0.BERR0SSF007 error flag clear
6	BERR0CL006	BERR0ST0.BERR0SSF006 error flag clear
5	BERR0CL005	BERR0ST0.BERR0SSF005 error flag clear
4	BERR0CL004	BERR0ST0.BERR0SSF004 error flag clear
3	BERR0CL003	BERR0ST0.BERR0SSF003 error flag clear
2	BERR0CL002	BERR0ST0.BERR0SSF002 error flag clear
1	BERR0CL001	BERR0ST0.BERR0SSF001 error flag clear
0	BERR0CL000	BERR0ST0.BERR0SSF000 error flag clear

7.4.9.4 BERR0STC1 — Bus error status clear register 1

This register is used to clear the bus error flags of BERR0ST1.

Access: This register can be accessed in 32-bit units.

Address: FFC0 7014_H

Initial value: Reading this register returns always 0000 0000_H.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BERR0CL131	BERR0CL130	BERR0CL129	BERR0CL128	0	0	0	0	0	0	0	0	0	0	0	BERR0CL116
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	BERR0CL112	BERR0CL111	BERR0CL110	BERR0CL109	BERR0CL108	0	0	BERR0CL105	BERR0CL104	BERR0CL103	BERR0CL102	BERR0CL101	BERR0CL100
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

All clear bits in this register effect the following, when written:

- BERR0CL1x = 0: no function
- BERR0CL1x = 1: respective error flag in the bus error status register 1 is cleared

Table 7.17 BERR0STC1 register contents

Bit position	Bit name	Function
31	BERR0CL131	BERR0ST1.BERR0SSF131 error flag clear
30	BERR0CL130	BERR0ST1.BERR0SSF130 error flag clear
29	BERR0CL129	BERR0ST1.BERR0SSF129 error flag clear
28	BERR0CL128	BERR0ST1.BERR0SSF128 error flag clear
27 to 17	Reserved	Reading returns 0. When written, write the initial value.
16	BERR0CL116	BERR0ST1.BERR0SSF116 error flag clear
15 to 13	Reserved	Reading returns 0. When written, write the initial value.
12	BERR0CL112	BERR0ST1.BERR0SSF112 error flag clear
11	BERR0CL111	BERR0ST1.BERR0SSF111 error flag clear
10	BERR0CL110	BERR0ST1.BERR0SSF110 error flag clear
9	BERR0CL109	BERR0ST1.BERR0SSF109 error flag clear
8	BERR0CL108	BERR0ST1.BERR0SSF108 error flag clear
7 to 6	Reserved	Reading returns 0. When written, write the initial value.
5	BERR0CL105	BERR0ST1.BERR0SSF105 error flag clear
4	BERR0CL104	BERR0ST1.BERR0SSF104 error flag clear
3	BERR0CL103	BERR0ST1.BERR0SSF103 error flag clear
2	BERR0CL102	BERR0ST1.BERR0SSF102 error flag clear
1	BERR0CL101	BERR0ST1.BERR0SSF101 error flag clear
0	BERR0CL100	BERR0ST1.BERR0SSF100 error flag clear1:clears error flag

7.4.10 DMA Bus Error Control Module (BECM)

The Bus Error Control Module detects error responses during DMA Controller tranaction attempts via the cross-connect.

In case of an transaction error detection an interrupt is generated.

Following errors can be detected:

- illegal ID error response from cross-connect
- error response from targetted cross-connect slave module

Table 7.18 DMA bus error status registers overview

Register Name	Symbol	Address
Access protect control for BECM registers	BECMBUSERRKEYCODE	FFFF 7400$_H$
DMAC bus error mask register	BECMBUSERRMASK	FFFF 7404$_H$
DMAC bus error flag register	BECMBUSERRFLAG	FFFF 7408$_H$
DMAC bus error clear register	BECMBUSERRCLEAR	FFFF 740C$_H$
DMAC bus error index register	BECMBUSERRINDEX	FFFF 7410$_H$
DMAC bus error info register 0	BECMBUSERRINFO0	FFFF 7414$_H$
DMAC bus error info register 1	BECMBUSERRINFO1	FFFF 7418$_H$

7.4.10.1 BECMBUSERRKEYCODE — Access protect control for BECM registers

This register enables write access to the BECM registers.

Access: This register can be accessed in 32-bit units.

Address: FFFF 7400$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	BECMKEY[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 7.19 BECMBUSERRKEYCODE register contents

Bit position	Bit name	Function
31 to 8	Reserved	Reading returns 0. When written, write the initial value.
7 to 0	BECMKEY[7:0]	When AA$_H$ is written, write access to the BECM registers is enabled. BECMKEY[0] will become 0. Writing any other value than AA$_H$ prohibits write access to the BECM registers. BECMKEY[0] will become 1. When access is prohibited, write access to the BECM registers is invalid and reading of the BECM registers returns 0000 0000$_H$. Reading this register returns 0 of the bits 31 to 1 and bit 0 (BECMKEY[0]) indicates BECM registers access status: 1: BECM registers access prohibited 0: BECM registers access enabled

7.4.10.2 BECMBUSERRMASK — DMAC bus error mask register

This register controls the generation of interrupts in case of error detections.

Access: This register can be accessed in 32-bit units.

Address: FFFF 7404_H

Initial value: 0000 $FFA6_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	1	1	BECMMASK12	BECMMASK11	1	1	1	1	0	1	0	0	1	1	0
Initial value	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	0
R/W	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R

Table 7.20 BECMBUSERRMASK register contents

Bit position	Bit name	Function
31 to 13	Reserved	Reading returns the initial value. When written, write the initial value.
12	BECMMASK12	DMA Controller post write response error interrupt generation control 0: interrupt upon post write response error generated 1: no interrupt upon post write response error generated
11	BECMMASK11	DMA Controller response error interrupt generation control 0: interrupt upon response error generated 1: no interrupt upon response error generated
10 to 0	Reserved	Reading returns the initial value. When written, write the initial value.

7.4.10.3 BECMBUSERRFLAG — DMAC bus error flag register

This register holds error flags, that indicated the cause of an assertion of the internal bus error interrupt (EIINT channel number 255).

The error flags of this register can be cleared via the DMAC bus error clear register BECMBUSERRCLEAR.

The settings of the DMAC bus error mask register BECMBUSERRMASK does not affect this register.

Access: This register can be accessed in 32-bit units.

Address: FFFF 7408$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	BECMFLG12	BECMFLG11	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R

Table 7.21 BECMBUSERRFLAG register contents

Bit position	Bit name	Function
31 to 13	Reserved	Reading returns the initial value. When written, write the initial value.
12	BECMFLG12	DMA Controller post write response status flag 0: DMA Controller post write response error not detected 1: DMA Controller post write response error detected
11	BECMFLG11	DMA Controller response error status flag 0: DMA Controller response error not detected 1: DMA Controller response error detected
10 to 0	Reserved	Reading returns the initial value. When written, write the initial value.

7.4.10.4 BECMBUSERRCLEAR — DMAC bus error clear register

This register is used to clear the error flags of the DMA Controller bus error flag register BECMBUSERRFLAG.

For clearing an error flag BECMFLGx in the BECMBUSERRFLAG register, proceed as follows:

1. set BECMBUSERRCLEAR.BECMCLRx = 1
2. set BECMBUSERRCLEAR.BECMCLRx = 0
3. set BECMBUSERRFLAG.BECMFLGx = 0

Access: This register can be accessed in 32-bit units.

Address: FFFF 740C$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	BECM CLR12	BECM CLR11	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R

Table 7.22 BECMBUSERRCLEAR register contents

Bit position	Bit name	Function
31 to 13	Reserved	Reading returns the initial value. When written, write the initial value.
12	BECMCLR12	DMA Controller post write response status flag clear 0: no function 1: clear post write response error flag BECMBUSERRFLAG.BECMFLG12
11	BECMCLR11	DMA Controller response status flag clear 0: no function 1: clear response error flag BECMBUSERRFLAG.BECMFLG11
10 to 0	Reserved	Reading returns the initial value. When written, write the initial value.

7.4.10.5 BECMBUSERRINDEX — DMAC bus error index register

Access: This register can be accessed in 32-bit units.

Address: FFFF 7410$_H$

Initial value: undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	BECMECODE[6:0]						
Initial value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 7.23 BECMBUSERRINDEX register contents

Bit position	Bit name	Function
31 to 7	Reserved	Reading returns 0. When written, write the initial value.
6 to 0	BECM ECODE[6:0]	Error number of BECMBUSERRINFO0 and BECMBUSERRINFO1 0C$_H$: DMA Controller post write response error 0B$_H$: DMA Controller response error 00$_H$: no error detected Other: reserved

7.4.10.6 BECMBUSERRINFO0 — DMAC bus error info register 0

Access: This register can be accessed in 32-bit units.

Address: FFFF 7414_H

Initial value: undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	BECMVCID[6:0]						
Initial value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.24 BECMBUSERRINFO0 register contents

Bit position	Bit name	Function
31 to 7	Reserved	Reading returns 0. When written, write the initial value.
6 to 0	BECMVCID[6:0]	VCID DMA Controller post write response error occurred. The DMA Controller uses four IDs (200_H, 201_H, 220_H, 221_H) for transactions on the CPU Subsystem's internal global interconnect (GVCI). BECMVCID[6:0] holds one of these IDs, when a DMA Controller GVCI transaction error occurs.

7.4.10.7 BECMBUSERRINFO1 — DMAC bus error info register 1

Access: This register can be accessed in 32-bit units.

Address: FFFF 7418$_H$

Initial value: undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BECMEADR[31:16]															
Initial value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BECMEADR[15:0]															
Initial value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 7.25 BECMBUSERRINFO1 register contents

Bit position	Bit name	Function
31 to 0	BECMEADR[31:0]	Address DMA Controller post write response error occurred.

7.5 Operation

7.5.1 External Interrupts (NMI / INTP)

NMI and INTP interrupts are input from external devices.

The NMI is input to the Error Control Module. For details see Section 48, Error Control Module (ECM).

The INTP interrupts are can directly generate EI level interrupts.

Figure 7.1 shows an IRQ (external interrupt) detection flow.

- Select an IRQ detecting method (rising edge, falling edge, and both edges) by setting the FCLAnCTLm register.
- After an IRQ has been detected, an interrupt request is sent to the INTC.

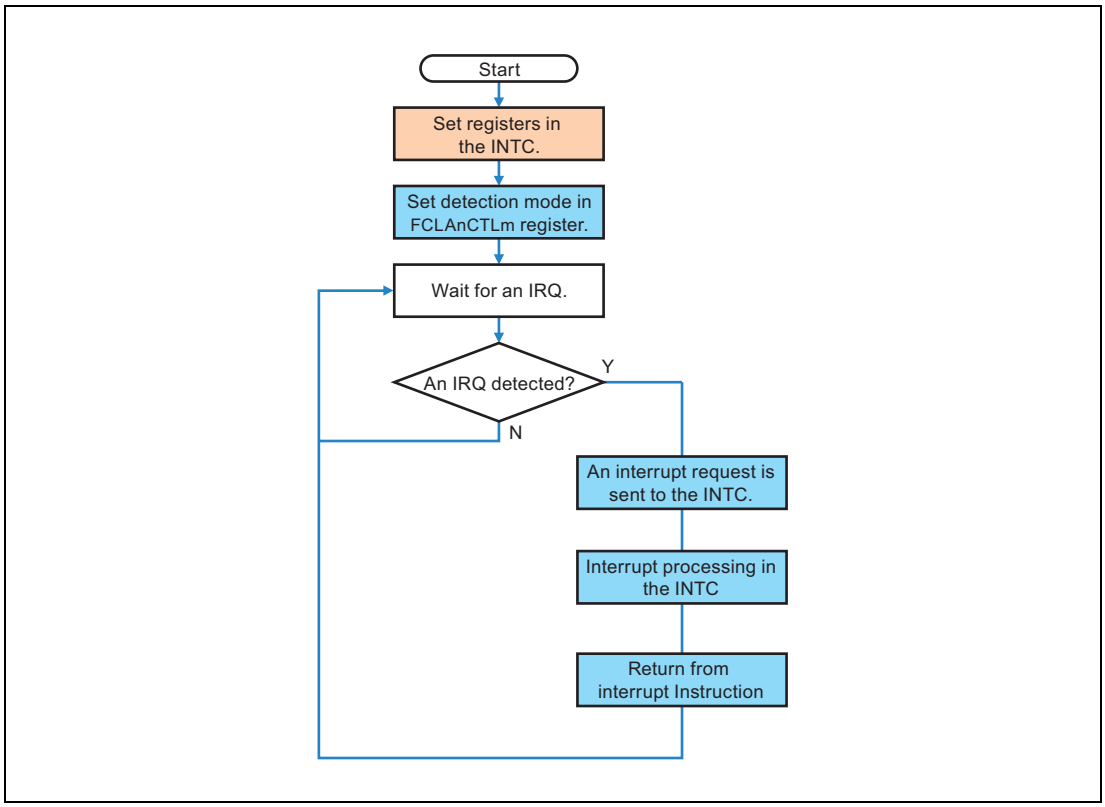


Figure 7.1 External Interrupt Processing Flow

7.6 Interrupt Response Times

Table 7.26 Minimum interrupt Response Times to be Considered

Interrupt Request Source	Number of Cycles for Processing			
INTC Connection	INTC2	INTC1	in CPU	Total (in the case of edge detection by fixed vector method a)
Directly input to INTC1	—	2 x CPUCLK <1 x CPUCLK>	Refer to the description under CPU below	7 x CPUCLK
Input via INTC2	3 x XCCLK + 1 x CPUCLK <2 x XCCLK + 1 x CPUCLK>	—		3 x XCCLK + 6 x CPUCLK

Note 1. The numbers in < > indicate the numbers of cycles in the case of level detection.

Vector method	Cache HIT/MISS	in CPU		Vector method
		D1M2(H), D1M1A	Other devices	
Fixed vector method	a) ISR entry is hit	5 x CPUCLK		a) Fixed vector method
	b) ISR entry is missed	10 x CPUCLK	8 x CPUCLK	
Vector table reference method	c) Vector code Flash assigned, ISR entry is hit	14 x CPUCLK	12 x CPUCLK	b) Vector table reference method Code flash assigned
	d) Vector code Flash assigned, ISR entry is missed	19 x CPUCLK	15 x CPUCLK	
	e) Vector Local RAM assigned, ISR entry is hit	9 x CPUCLK		c) Vector table reference method Local RAM assigned
	f) Vector Local RAM assigned, ISR entry is missed	14 x CPUCLK	12 x CPUCLK	

7.7 Using Interrupt Request Signals to Initiate Data Transfer

An interrupt request signal activates the DMAC to perform data transfer. For details, see **Section 8, DMA**.

Section 8 DMA

8.1 Overview

8.1.1 Overview

Direct memory access (DMA) is used to access data without going through the CPU.

A DMAC includes registers for storing transfer information. DMA has two 8-channel DMAC modules.

In this manual, DTFR denotes the function to select among hardware DMA transfer sources for a DMAC and retain the DMA transfer request. The DTFR can handle 128 types of hardware DMA transfer sources.

The address space that can be used for DMA transfer is a 4 GB address space represented by a 32-bit address. For information about which resource is assigned to a particular area in the 4 GB address space and which area is accessible from DMA, see Section 5, Address Spaces.

8.1.2 Term Definition

Table 8.1 shows the terms used in this section.

Table 8.1 List of Term Definitions

Term	Meaning
DMA transfer	A general term for data transfer carried out by DMA.
DMA cycle	A series of actions that consist of reading an amount of data specified by the transfer size (8/16/32/64/128 bits) from the address specified by the source address and writing it to the address specified by the destination address. The first half of the DMA cycle (reading part) is called a read cycle, and the second half (writing part) is called a write cycle.
Hardware DMA transfer source	A trigger for a DMA transfer request given by an internal peripheral device.
Hardware DMA transfer request	A DMA transfer request generated by a hardware DMA transfer source
Software DMA transfer request	A DMA transfer request generated by software writing to a register.
DMA transfer request	A trigger to start DMA transfer with a DMAC.
Single transfer	DMA transfer consisting of one DMA cycle started by one DMA transfer request.
Block transfer 1	DMA transfer consisting of a number of DMA cycles specified by the transfer count in the transfer information, started by one DMA transfer request.
Block transfer 2	DMA transfer consisting of a number of DMA cycles specified by the address reload count in the transfer information, started by one DMA transfer request.
Block transfer	A general term for both block transfer 1 and block transfer 2.
Last transfer	The DMA cycle carried out when the transfer count in the transfer information is 1.
Address reload transfer	The DMA cycle carried out when the address reload count in the transfer information is 1 if the reload function 2 is used.
Suspension	An action of pausing DMA transfer during block transfer. You can resume DMA transfer after suspension.
Resume	An action of resuming suspended DMA transfer.
Transfer abort	An action of aborting DMA transfer in the middle. You cannot resume DMA transfer after that.

8.2 DMA Function

8.2.1 Basic Operation of DMA Transfer

8.2.1.1 Transfer Mode

DMA has three transfer modes.

Single Transfer

One DMA cycle is executed when a DMA transfer request is accepted.

Block Transfer 1

A number of DMA cycles specified in the transfer count register are executed when a DMA transfer request is accepted.

Block Transfer 2

A number of DMA cycles specified by the address reload count are executed when a DMA transfer request is accepted. If the address reload count is larger than the value in the transfer count register, a number of DMA cycles specified in the transfer count register are executed.

8.2.1.2 Executing a DMA Cycle

Dual-address (2-cycle) transfer operation is adopted to DMA cycle.

DMA always executes a write cycle after a read cycle is complete. For example, if the transfer data size is 128 bits, a write cycle is executed after a read cycle for the 128-bit data is finished. A write cycle never starts in the middle of a read cycle.

8.2.1.3 Updating Transfer Information

When a DMA cycle is executed, DMA updates transfer information as follows.

Source Address and Destination Address

Transfer information will be updated as described in **Table 8.2** according to the settings for the source address and destination address and the settings in the transfer control register such as the count directions of source address and destination address and transfer data size.

Table 8.2 Updating the Source Address and the Destination Address (1/2)

Direction of Count	Transfer Data Size	Address after Update
Increment	8 bit	(address before update) + 0000 0001_H
	16 bit	(address before update) + 0000 0002_H
	32 bit	(address before update) + 0000 0004_H
	64 bit	(address before update) + 0000 0008_H
	128 bit	(address before update) + 0000 0010_H

Table 8.2 Updating the Source Address and the Destination Address (2/2)

Direction of Count	Transfer Data Size	Address after Update
Decrement	8 bit	(address before update) – 0000 0001_H
	16 bit	(address before update) – 0000 0002_H
	32 bit	(address before update) – 0000 0004_H
	64 bit	(address before update) – 0000 0008_H
	128 bit	(address before update) – 0000 0010_H
Fixed	—	Same as the address before update.

When you use the reload function, a special update rule other than the one described in **Table 8.2** is applied for the last transfer and the address reload transfer. For details, see Section 8.2.3, Reload Function.

Transfer Count/Address Reload Count

The transfer count is decremented by one for every DMA cycle.

The address reload count is decremented by one for every DMA cycle when the reload function 2 or block transfer 2 is used. When the reload function 2 or block transfer 2 is not used, it is not updated.

When you use the reload function, a special update rule is applied for the last transfer and the address reload transfer. For details, see Section 8.2.3, Reload Function.

Other transfer information

Other transfer information is not updated during execution of a DMA cycle.

8.2.1.4 Last Transfer and Address Reload Transfer

The last transfer means a DMA cycle executed when the value in the transfer count register, which shows the remaining number of transfers, is one. The last transfer differs in operation compared to other DMA cycles as follows.

- The transfer completion flag (DCSTn.TC) is set when the last transfer is complete. (Only applicable for a DMAC)
- The channel operation enable (DCENn.DTE) bit is cleared when the last transfer is complete. (Only applicable for a DMAC. When the continuous transfer is disabled.)
- When the transfer completion interrupt output enable is set, a transfer completion interrupt is output when the last transfer is complete.
- When the reload function 1 is enabled, the reload function 1 is executed at the timing of the last transfer. For details, see Section 8.2.3, Reload Function.

The address reload transfer means a DMA cycle executed when the reload function 2 is enabled and the address reload count is one. The reload function 2 is executed during the address reload transfer. For details, see Section 8.2.3, Reload Function.

8.2.1.5 Transfer Completion Interrupt and Transfer Count Match Interrupt Outputs

DMA can output a transfer completion interrupt and a transfer count match interrupt to external devices.

Transfer Completion Interrupt Output

When the transfer completion interrupt output enable (DTCTn.TCE) is set in the transfer control register, a DMAC requests a transfer completion interrupt when the last transfer is complete.

Transfer Count Match Interrupt Output

When the transfer count match interrupt enable (DTCTn.CCE) is set in the transfer control register, a DMAC requests a DMAC transfer count match interrupt when the DMA cycle in which the transfer count compare register and the transfer count have the same value is complete.

Figure 8.1 shows the operation of the transfer completion interrupt and the transfer count match interrupt.

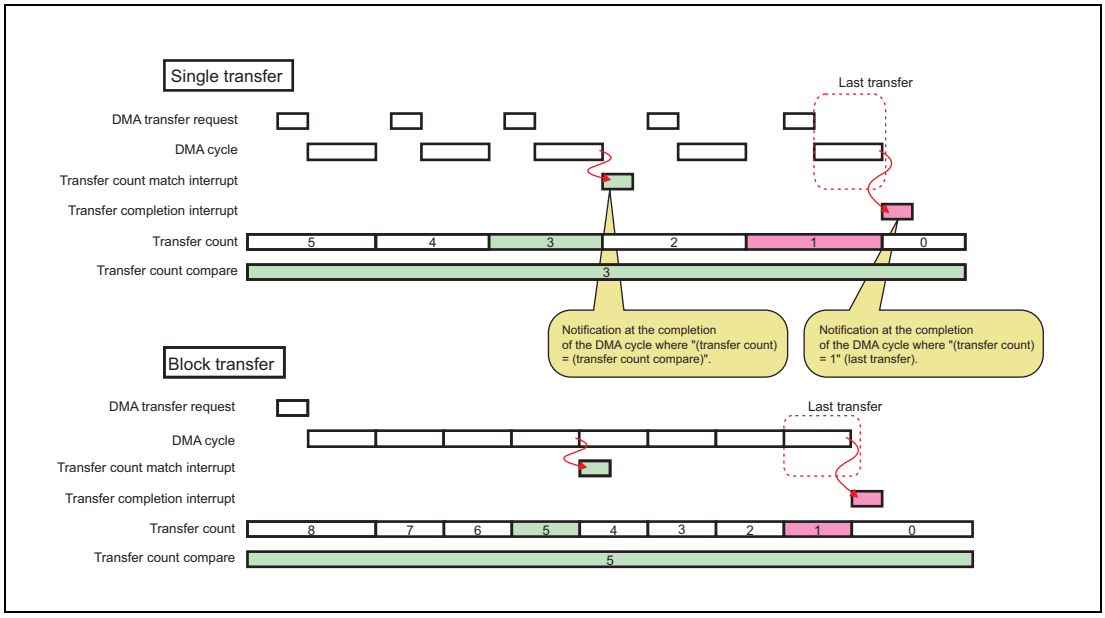


Figure 8.1 Transfer Completion Interrupt and Transfer Count Match Interrupt

8.2.1.6 Continuous Transfer

If the continuous transfer is not used, a DMAC sets the transfer completion flag (DCSTn.TC) and clears the channel operation enable (DCENn.DTE) bit when the last transfer is complete. In this case, a DMA transfer request is not accepted when the request is generated after the completion of the last transfer.

If the continuous transfer is used, the channel operation enable (DCENn.DTE) bit is not cleared when the last transfer is complete, and a DMA transfer request can be accepted even when the transfer completion flag is set. If DMA is used for a case where a specified number of DMA transfers are executed repetitively, software overhead associated with clearing the transfer completion flag and setting the channel operation enable bit after the completion of the last transfer can be reduced by using the continuous transfer.

The continuous transfer is enabled by setting the continuous transfer enable (DTCTn.MLE) in the DMAC transfer control register.

The continuous transfer is designed to work with the reload function 1. The continuous transfer function itself does not update the source address register, destination address register, and transfer count register. If, after the last transfer is complete, you want to restore the source address register, destination address register, and transfer count register to their state before the DMA transfer started, use the reload function 1 and set the values of the source address register, destination address register,

and the transfer count register before the DMA transfer starts to the reload source address register, reload destination address register, and reload transfer count register respectively.

Figure 8.2 shows an operation of continuous transfer by a DMAC.

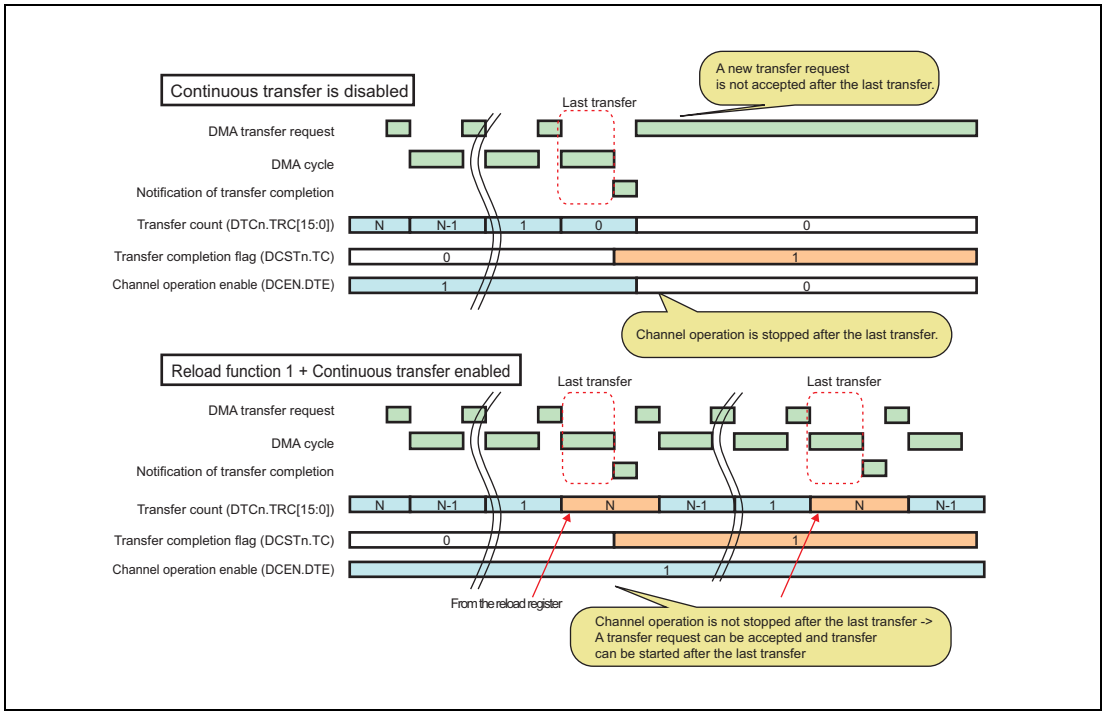


Figure 8.2 Operation of Continuous Transfer by a DMAC

8.2.2 Channel Priority Order

This subsection explains arbitration between multiple DMA channels.

8.2.2.1 DMAC Channel Arbitration

A DMAC selects one channel out of eight channels by arbitration. Arbitration is done according to a fixed priority order. The priority order is “channel 0 > channel 1 > channel 2 > channel 3 > channel 4 > channel 5 > channel 6 > channel 7” for DMAC0, and “channel 8 > channel 9 > channel 10 > channel 11 > channel 12 > channel 13 > channel 14 > channel 15” for DMAC1.

Arbitration is done for every DMA cycle. No arbitration occurs between the read and write of a DMA cycle.

If, at the time when one DMA cycle in the middle of a block transfer of a channel is complete, there is a DMA transfer request from a channel with a higher priority than the channel, a DMA cycle of the channel with the higher priority will be executed next as the result of arbitration.

If a DMAC executes the block transfer 1 or block transfer 2, DMAC channel arbitration is done for every DMA cycle, and possibly a DMA cycle of another DMAC channel with a higher priority can cut in.

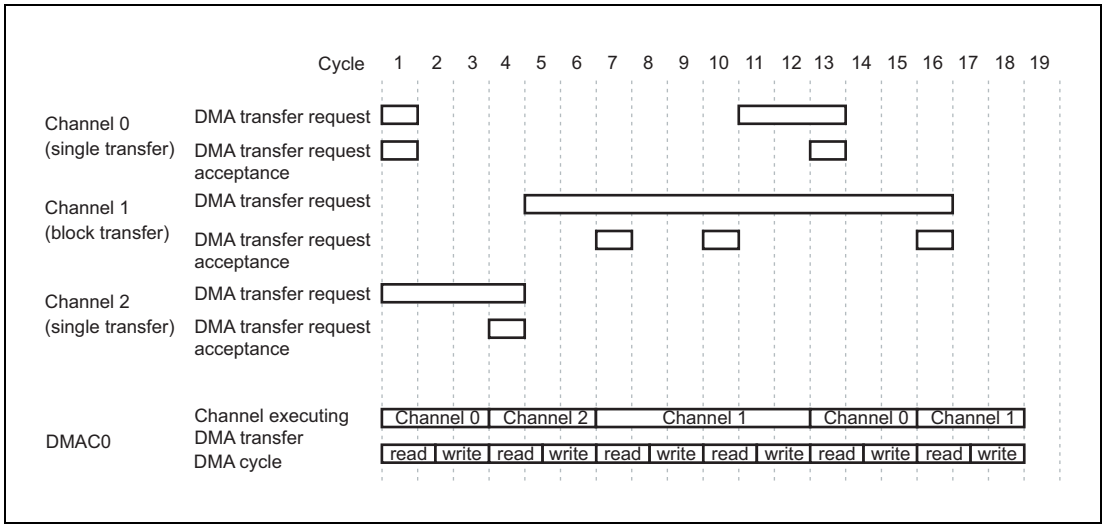


Figure 8.3 DMAC Channel Arbitration

Cycle numbers shown in **Figure 8.3** are for explanation purpose only. They do not indicate an actual number of cycles necessary for executing DMA transfer.

In **Figure 8.3**, DMA transfer requests for channels 0 and 2 are generated at Cycle 1. As a result of arbitration, a DMA cycle for channel 0 starts because its priority is higher. At Cycle 4, a DMA cycle for channel 2 starts. At Cycle 5, a DMA transfer request for channel 1 is generated. The DMA cycle for channel 2 is still ongoing and no arbitration is done at this point. At Cycle 7, a DMA cycle for channel 1 starts. Channel 1 uses block transfer. Another DMA cycle continues at Cycle 10 because there are no DMA transfer requests from other channels. At Cycle 11, a DMA transfer request for channel 0 is generated. The DMA cycle for channel 1 is still ongoing and no arbitration is done at this point. At Cycle 12, the DMA cycle for channel 1 is complete. At Cycle 13, a DMA cycle for channel 0 starts as a result of arbitration between DMA channels 0 and 1.

It should be noted that, even though a block transfer of channel 1 has been already started, a DMA cycle of not channel 1 but channel 0 is executed at Cycle 13 because the priority of the latter is higher. At Cycle 15, the DMA cycle for channel 0 is complete. At Cycle 16, a DMA cycle for channel 1 starts again. At Cycle 18, the last DMA cycle of the block transfer of channel 1 is complete.

8.2.2.2 Interface Arbitration

The DMA units DMAC0 and DMAC1 execute DMA transfer independently.
That means, two DMA transactions, one on each DMA unit, can be processed concurrently.

Each channel of DMAC0 and DMAC1 has the capability to store one DMA transfer request, while it is already processing the previous request of that channel.

CAUTION

If a DMA transfer request occurs, while the previous one is pending, the new request gets lost.

8.2.3 Reload Function

8.2.3.1 Overview of the Reload Function

The reload function updates a portion of transfer information, more specifically, the source address, destination address, transfer count, and address reload count, to the predefined values during DMA transfer.

The reload function has two types of functions: reload function 1 and reload function 2.

8.2.3.2 Operation of Reload Function 1

When the reload function 1 is enabled, actions described in **Table 8.3** are executed at the time of the last transfer according to the reload function 1 setting.

If the last transfer is generated at the same time as the address reload transfer is generated, reload is executed according to the reload function 1 setting.

Table 8.3 Operation of Reload Function 1

Reload Function 1 Setting (DTCTn.RLD1M[1:0])	Register	Action at the Last Transfer
00 (Reload function 1 disabled.)	Source address	Not reloaded.
	Destination address	Not reloaded.
	Transfer count	Not reloaded.
	Address reload count	Not reloaded.
01 (Reload function 1 enabled. Reloading source address and transfer count.)	Source address	The reload source address is copied to this.
	Destination address	Not reloaded.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	• If the reload function 2 is disable: Not reloaded. • If the reload function 2 is enabled: The reload address reload count is copied to this.
10 (Reload function 1 enabled. Reloading destination address and transfer count.)	Source address	Not reloaded.
	Destination address	The reload destination address is copied to this.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	• If the reload function 2 is disable: Not reloaded. • If the reload function 2 is enabled: The reload address reload count is copied to this.
11 (Reload function 1 enabled. Reloading source address, destination address, and transfer count.)	Source address	The reload source address is copied to this.
	Destination address	The reload destination address is copied to this.
	Transfer count	The reload transfer count is copied to this.
	Address reload count	• If the reload function 2 is disable: Not reloaded • If the reload function 2 is enabled: The reload address reload count is copied to this.

Figure 8.4 shows an operation of the reload function 1.

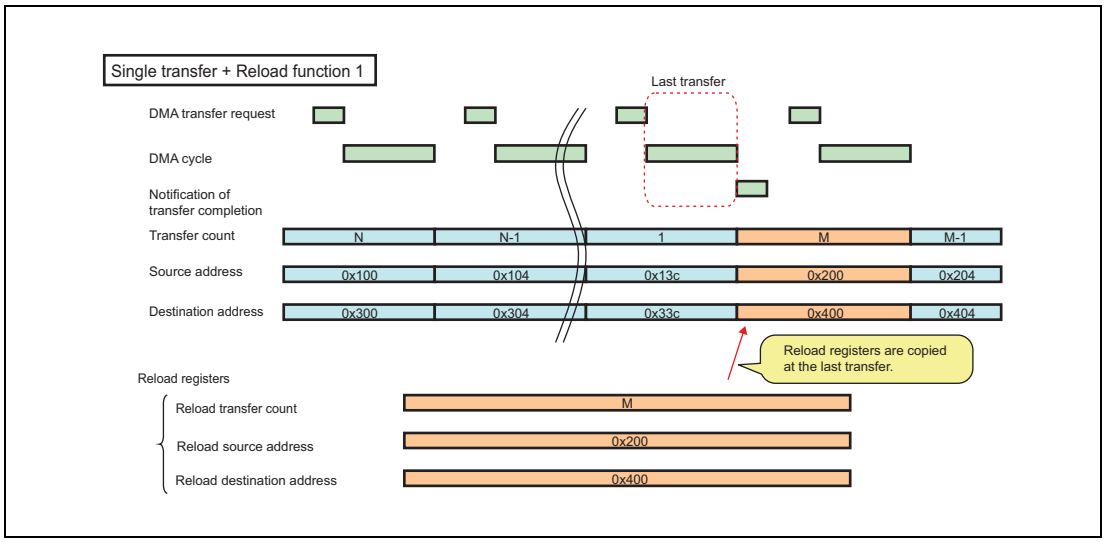


Figure 8.4 Operation of Reload Function 1

8.2.3.3 Reload Function 2

When the reload function 2 is enabled, actions described in **Table 8.4** are executed at the time of the address reload transfer according to the reload function 2 setting.

Table 8.4 Operation of Reload Function 2

Reload Function 2 Setting (DTCTn.RLD2M[1:0])	Register	Action at the Address Reload Transfer
00 (Reload function 2 disabled.)	Source address	Not reloaded.
	Destination address	Not reloaded.
	Address reload count	Not reloaded.
01 (Reload function 2 enabled. Reloading source address.)	Source address	The reload source address is copied to this.
	Destination address	Not reloaded.
	Address reload count	The reload address reload count is copied to this.
10 (Reload function 2 enabled. Reloading destination address.)	Source address	Not reloaded.
	Destination address	The reload destination address is copied to this.
	Address reload count	The reload address reload count is copied to this.
11 (Reload function 2 enabled. Reloading source address and destination address.)	Source address	The reload source address is copied to this.
	Destination address	The reload destination address is copied to this.
	Address reload count	The reload address reload count is copied to this.

Figure 8.5 shows an operation of the reload function 2.

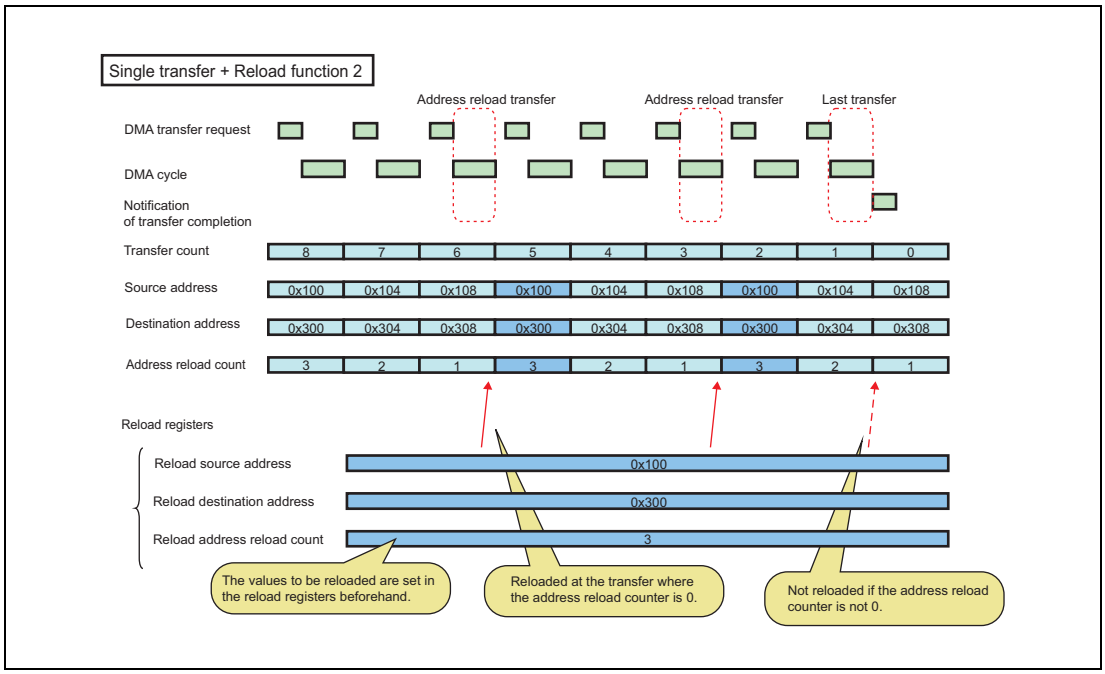


Figure 8.5 Operation of Reload Function 2

Figure 8.6 shows an operation when both the reload function 1 and the reload function 2 are used simultaneously.

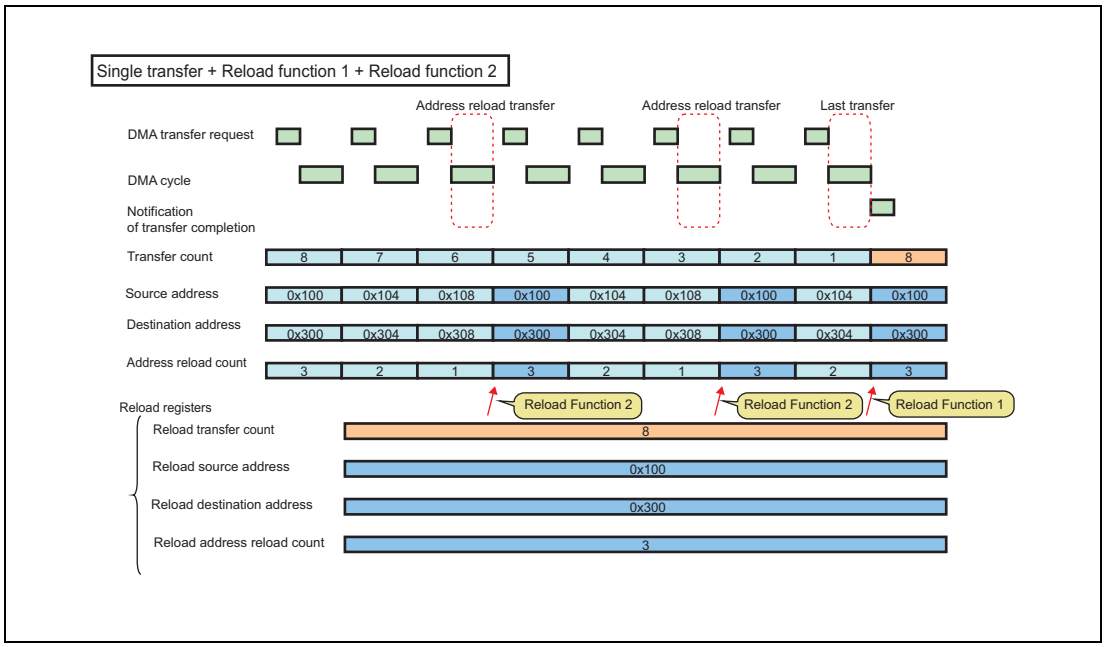


Figure 8.6 Operation when combining the reload function 1 and the reload function 2

8.2.3.4 Timing of Setting DMAC Reload Registers

You can set up the reload source address register, reload destination address register, and reload transfer count register any time (even during DMA transfer). As an exception, if you update the reload source address register, reload destination address register, and reload transfer count register during DMA transfer, there may be a conflict between reloading at the last transfer or address reload transfer and updating the reload register. In order to avoid this conflict, setting of reload registers must be completed before the last transfer or address reload transfer starts.

If you need to update the reload source address register, reload destination address register, and reload transfer count register during DMA transfer, one way to know the right timing to update is to use a DMA transfer count match interrupt. In this case, you must set up the DMA transfer count compare register (DTCCn) so that you can have enough margin for the time necessary to update the reload registers.

8.2.4 Chain Function

8.2.4.1 Overview

DMA offers a function called chain function. If you use the chain function, the completion of the DMA cycle or last transfer for one channel can trigger a DMA transfer request for another channel. A DMA transfer request for another channel initiated by the chain function is called a chain request.

You can select the condition for generating a chain request from the following two options.

- Always chain: A chain request is generated at the completion of every DMA cycle.
- Chain at the last transfer: A chain request is generated at the completion of the last transfer.

Figure 8.7 shows a case of “always chain” operation.

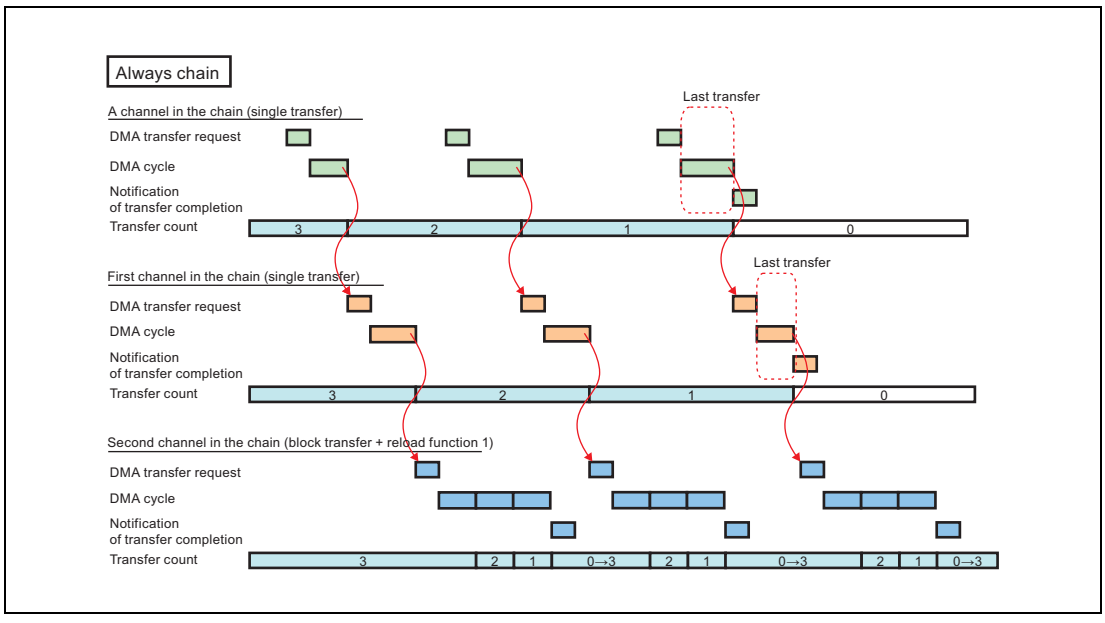


Figure 8.7 Operation of “Always Chain”

Figure 8.8 shows a care of "chain at the last transfer" operation.

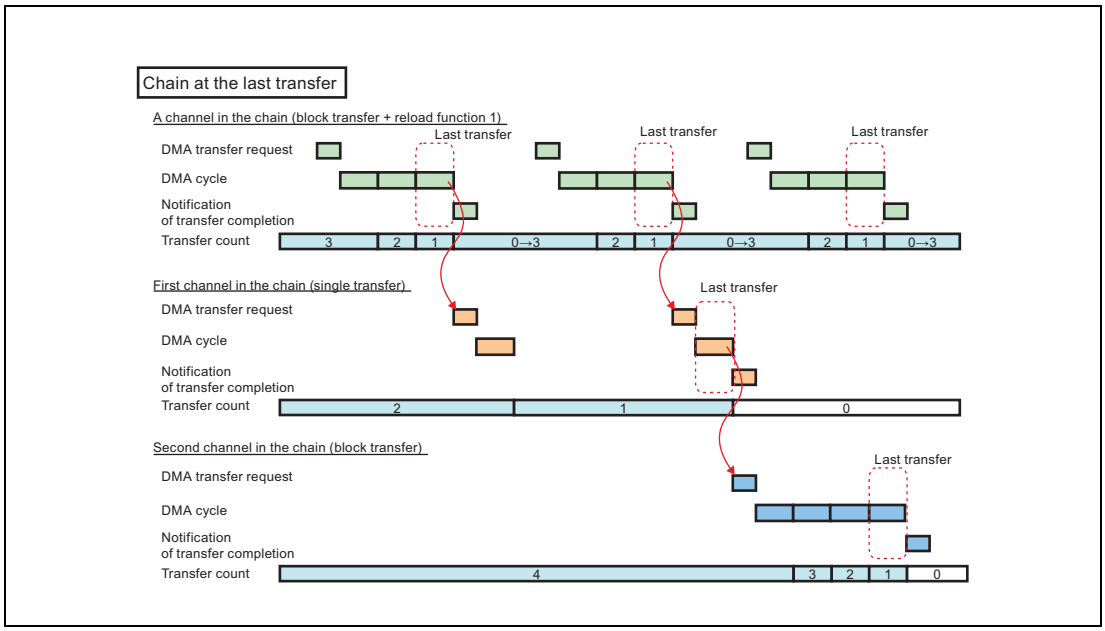


Figure 8.8 Operation of "Chain at the Last Transfer"

8.2.4.2 Setting Up the Chain Function

For a DMAC, you need to write to the chain enable (DTCTn.CHNE[1:0]) and the next channel in the chain selection (DTCTn.CHNSEL[2:0]) in the DMAC transfer control register in order to set up the type of chain function and the next channel number in the chain.

8.2.4.3 Caution for Using the Chain Function

The chain function sets the software DMA transfer request flag of the next channel in the chain as a part of its function. Therefore, you need to set up the channel settings of the next channel in the chain in the same way as when the software DMA transfer request is used. If you specify a channel using the hardware DMA transfer request for the next channel in the chain, the chain function does not work.

A channel and its next channel in the chain must belong to the same module (DMAC0 and DMAC1). You cannot specify a channel in another module for its next channel in the chain.

8.2.5 DMAC Operation

8.2.5.1 Types of DMA Transfer Requests and Assigning DMA Transfer Requests

A DMAC starts DMA transfer by accepting a hardware DMA transfer request or software DMA transfer request. The DMA transfer request selection assignment (DRS) bit in the DMAC transfer control register (DTCTn) determines whether a hardware DMA transfer request or a software DMA transfer request is used.

In the case of a hardware DMA transfer request for a DMAC, one out of 128 hardware DMA transfer sources is selected and assigned for each channel of the DMAC in the DTFR. This assignment is configured in the DTFR setting registers.

8.2.5.2 Generating and Accepting a Software DMA Transfer Request

By setting the software DMA transfer request flag (SR) in the DMAC transfer status register (DCSTn) using the DMAC transfer status set register (DCSTSn), a software DMA transfer request can be generated.

The software DMA transfer request flag is automatically cleared when the DMAC processes the DMA transfer request. The timing at which the software DMA transfer request flag is automatically cleared differs depending on the transfer mode of the DMA transfer to be executed.

- In the single transfer mode, the software DMA transfer request flag is cleared whenever the software DMA transfer request is accepted.
- In the block transfer 1 mode, the software DMA transfer request flag is cleared when the last transfer starts.
- In the block transfer 2 mode, the software DMA transfer request flag is cleared when the last transfer or address reload transfer starts.

The software DMA transfer request flag can also be cleared by software using the DMAC transfer status clear register (DCSTCn). When you abort a DMA transfer of a DMAC channel, you must clear the software DMA transfer request flag.

8.2.5.3 Generating and Accepting a Hardware DMA Transfer Request

Upon detection of a rising edge on the configured DMA transfer source a DMA transfer request is generated, indicated by the hardware DMA transfer request flag DCSTn.DR = 1.

The hardware DMA transfer request flag is automatically cleared when the DMAC processes the DMA transfer request. The timing at which the hardware DMA transfer request flag is automatically cleared differs depending on the transfer mode of the DMA transfer to be executed.

- In the single transfer mode, the hardware DMA transfer request flag is cleared whenever the hardware DMA transfer request is accepted.
- In the block transfer 1 mode, the hardware DMA transfer request flag is cleared when the last transfer starts.
- In the block transfer 2 mode, the hardware DMA transfer request flag is cleared when the address reload transfer starts.

8.2.5.4 Execution time of DMA transfers

A DMA cycle is triggered by the detection of a transfer request (hardware or software). The DMA cycle starts one DMA clock later. The read request is issued 2 DMA clocks later and the read data is provided after the latency time of the read bus has expired. After the read data has been provided in a read response, the DMAC needs two more DMA clock cycles to issue a write request. The whole DMA cycle is completed after expiration of the write latency time.

8.3 Suspension, Resume, Transfer Abort, and Clearing a DMA Transfer Request

8.3.1 DMA Suspension and Resume by Software Control

The DMA control register (DMACTL) is used to suspend DMA transfer for all channels.

When the DMA suspension bit (DMASPD) in the DMA control register is set, DMA puts all channels into the suspended state, If all channels are in the suspended state and the DMA suspension bit (DMASPD) in the DMA control register is cleared, DMA restores all channels from the suspended state to the normal state and resumes the DMA transfer of the suspended channel.

When all channels are put into the suspended state, DMA transfer is suspended for all channels without changing the value of the DTE bit of each DMAC channel.

8.3.2 Suspension, Resume, and Transfer Abort of a DMAC Channel

You can suspend the DMA transfer of a DMAC channel by clearing the channel operation enable bit (DCENn.DTE) in the DMAC channel operation enable setting register for the channel. If a DMA cycle is ongoing, the DMA transfer of a channel is suspended after the currently ongoing DMA cycle is finished. If you set the DTE bit for a channel while the DMA transfer of the channel is suspended, the DMA transfer of the channel is resumed.

If you want to abort the currently ongoing DMA transfer of a DMAC channel, similarly clear the channel operation enable bit (DCENn.DTE) in the DMAC channel operation enable setting register, and then clear the hardware DMA transfer request in the DTFR in the case of a hardware DMA transfer request, and clear the software DMA transfer request flag (DCSTn.SR) using the DMAC transfer request flag clear bit (DCSTCn.SRC) in the DMAC transfer status clear register in the case of a software DMA transfer request.

In case that the continuous transfer enable bit (DTCTn.MLE) is set, the channel operation enable bit (DCENn.DTE) is kept to be set. Even though the channel operation enable bit (DCENn.DTE) is cleared by software during a DMA cycle in a last transfer, the function of the continuous transfer enable bit (DTCTn.MLE) is given high priority and the channel operation enable bit (DCENn.DTE) is set after completion of the last transfer. If you want to abort an ongoing DMA transfer of a DMAC channel when continuous transfer function is enabled, please clear the continuous transfer enable bit (DTCTn.MLE) first and then clear the channel operation enable bit (DCENn.DTE) to abort DMA transfer of the DMAC channel. Only for the operation, DMAC Transfer Control Register (DTCTn) can be written under the channel operation is enabled (DCENn.DTE = 1).

Figure 8.9 shows an example of suspension, resume, and transfer abort of a DMAC channel.

In **Figure 8.9**, both channels 0 and 1 are executing block transfer. At time tick 1, DMA transfer of channel 1 starts. At time tick 2, a DMA transfer request for channel 0 is accepted. As a result of DMAC channel arbitration, DMA transfer of channel 0 starts because channel 0 has a higher priority than channel 1. At time tick 3, the last transfer of channel 0 is complete, and the remaining DMA transfer in the block transfer of channel 1 starts. At time tick 4, the last transfer of channel 1 is complete. After time tick 5, DMA transfer of channel 0 and DMA transfer of channel 1 are executed similarly. At time tick 7, the DMA transfer of channel 0 is suspended and, as a result of DMAC channel arbitration, the DMA transfer of channel 1 starts. At time tick 8, the last transfer of channel 1 is complete, and then, at time tick 9, the DMA transfer of channel 0 resumes. At time tick 10, the last transfer of channel 0 is suspended again, and then, at time tick 11, the DMA transfer of channel 0 is aborted. At time tick 12, the suspended state for channel 0 is cleared, but DMA transfer is not executed because the DMA transfer is aborted at time tick 11.

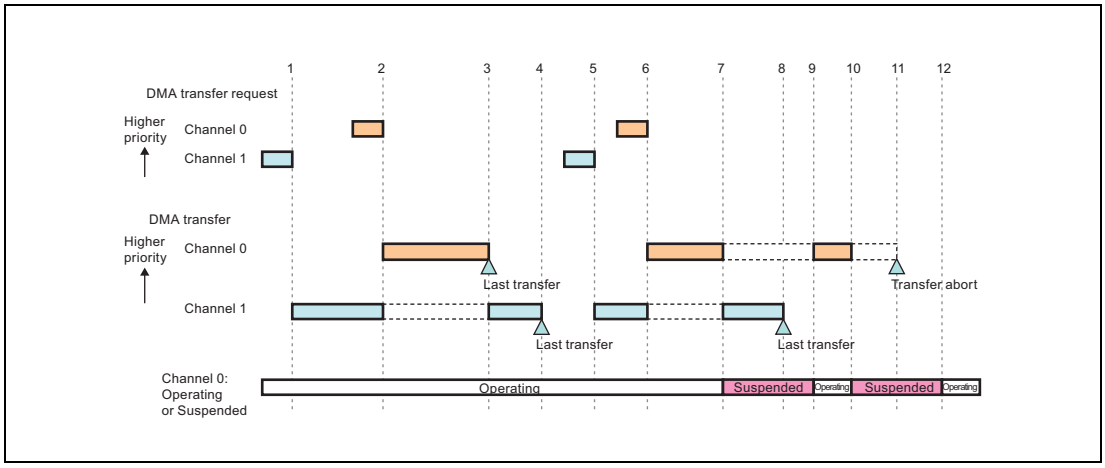


Figure 8.9 Example of Suspension, Resume, and Transfer Abort of a DMAC Channel

8.3.3 Masking and Clearing a Hardware DMA Transfer Request by the DTFR

If a DMAC uses a hardware DMA transfer request, you can temporarily disable (mask) the hardware DMA transfer request output from the DTFR to the DMAC by clearing the hardware DMA transfer source selection enable bit (DTFRn.REQEN) in the DTFR setting register.

Also, if a hardware DMA transfer source is used, you can clear a hardware DMA transfer request retained in the DTFR by using the hardware DMA transfer request clear (DTFRRQCn.DRQC) bit in the DTFR transfer request clear register.

Even if you suspend or abort DMA transfer of a DMAC channel, the hardware DMA transfer request selection/hold circuit of the DTFR is still running, and consequently, the DTFR may retain a hardware DMA transfer request that came to the DTFR during the suspension or transfer abort period of the DMAC channel. When you resume or start DMA transfer of a DMAC channel, clear the hardware DMA transfer request retained in the DTFR as required.

8.3.4 List of Suspend, Resume, and Transfer Abort Functions

Table 8.5 List of Suspend, Resume, and Transfer Abort Functions

Function	How to execute the function	Operation	Possibility of DMA transfer abort	Master that can execute the function (See Section 8.5, Reliability Function.)
DMA suspension and resume by software control	Setting and clearing the DMACTL.DMASPD.	All channels are in the suspended state.	Not possible*1	Special master
Suspension and resume of a DMAC channel	Clearing and setting the DCENn.DTE in each channel register.*2	DMA transfer of a channel is suspended.	Possible (by clearing the DMA transfer request flag during suspension)	Special master, and general master assigned to the channel.

Note 1. In order to abort DMA transfer, you need to abort transfer for the DMAC channel.

Note 2. In case that the continuous transfer enable bit (DTCTm.MLE) is set, please clear (or set) the continuous transfer enable bit (DTCTm.MLE) first.

8.4 Error Control

8.4.1 Type of Error

DMA can generate the following error.

- DMA Transfer Error
  This error is generated when error is detected in the read cycle or write cycle in a DMA cycle.
  This error can be generated in all DMAC channels during execution of DMA transfer.

8.4.2 DMA Transfer Error

When a DMA transfer error occurs, DMA asserts the DMA Transfer Error (INTDMAERR).

8.4.2.1 Operation of a DMAC When DMA Transfer Error Occurs

When DMA transfer error occurs in a DMAC, the transfer error flag (DCSTn.ER) in the DMAC transfer status register of the channel with the DMA transfer error. The DMAC error register (DMACER) shows the transfer error flags of all 16 DMAC channels.

While the transfer error flag of a channel is set, a new DMA cycle is not executed if the transfer error case DMA transfer disable setting (DTCTn.ESE) bit is set. On the other hand, a DMA cycle is executed regardless of the value of the transfer error flag if the transfer error case DMA transfer disable setting (DTCTn.ESE) bit is cleared.

If you want to abort the DMA transfer of a channel with DMA transfer error, follow the procedure to abort DMA transfer of the DMAC channel.

If DMA transfer error occurs during the read cycle of a DMA cycle, the write cycle is not executed. If DMA transfer error occurs during the write cycle of a DMA cycle, the validity of the write is not guaranteed.

Regardless of whether DMA transfer error occurs in the read cycle or write cycle of a DMA cycle, the source address, destination address, transfer count, and address reload count registers are updated.

8.5 Reliability Function

8.5.1 Overview

In this product, DMA offers the following reliability functions.

- Register access protection function
- Master information inherit function

8.5.2 Register Access Protection Function

The register access protection function allows access to the transfer information of each DMA channel from the master assigned to the channel but prohibits access from other masters.

The register access protection function enables you, for example, to prevent the settings of the channel from being read or updated by masters other than the one assigned to the channel.

8.5.2.1 Identifying the Accessing Master

DMA identifies a master based on the ID of the accessing CPU (PEID) and whether the CPU is in the supervisor mode (PSW.UM = 0) or the user mode (PSW.UM = 1).

8.5.2.2 Special Master Access

DMA treats access from the CPU supervisor mode as special master access.

8.5.2.3 General Master Access

DMA treats accesses except special master access as general master access.

In general master access, access to the following registers is allowed.

- The DMACER global registers
- Channel registers of the channels assigned by the channel assignment. (For details, see Section 8.5.2.4, Channel Assignment.)

In general master access, access to registers other than the above is not allowed.

8.5.2.4 Channel Assignment

To each channel, DMA can assign a master so that the master is allowed to use the channel. Channel assignment is configured in the channel master setting register (DMnnCM) by the CPU in the supervisor mode.

In general master access, the master assigned to a channel by channel assignment is allowed to access the channel registers of the channel. If the channel registers of a channel is accessed by a master other than the master assigned to the channel, the access is called illegal access. For information about illegal access, see Section 8.5.2.5, Illegal Access.

8.5.2.5 Illegal Access

DMA handles the following access as illegal access.

(a) General master access to the global registers
Except the DMACER register

(b) General master access to the channel registers of a channel by a master other than the master assigned to the channel

DMA's actions against illegal access are as follows.

For both cases (a) and (b),

- Write access is ignored.
- Read access returns 0 as read data.

Only for case (b),

- The information about the illegal access is stored in a register access protection violation register. The DMAC0 and DMAC1 have their own register access protection violation registers (DM0CMV and DM1CMV respectively).
- A generation of illegal access is notified to ECM.

Only the special master can access the register access protection violation registers. The special master can check whether illegal access has occurred by checking the register access protection violation registers periodically.

In addition, it is recommended that, when a master tries to use DMA and configures transfer information in the channel registers, the master should check whether the configuration has been successfully completed without illegal access by, for example, reading back the settings.

8.5.3 Master Information Inherit Function

In this product, DMA access inherits master information that is equivalent to the master information of the CPU assigned to the DMA channel.

The master information that is output from DMA is as in **Table 8.6**.

Table 8.6 Master Information That Is Output from DMA

Item	Value that is output from DMA
UM	Same as the UM bit value in the channel master setting register
SPID	Same as the SPID[1:0] bit value in the channel master setting register
PEID	Same as the PEID[2:0] bit value in the channel master setting register
DMA access	1

8.5.4 Other Reliability Functions

8.5.4.1 Restriction on the Next Channel in the Chain

The reliability function limits the channels you can select as the next channel in the chain.

When you use the chain function, the channel master settings of a channel and its next channel in the chain must be the same.

The chain function is designed so that a channel and its next channel in the chain are managed by the same master.

When DMA detects that different masters are assigned to a channel and its next channel in the chain, it is deemed illegal and the chain function is suppressed. More specifically, when DMA tries to execute the chain function, DMA compares the chain master settings of the channel and its next channel in the chain, and if the settings are the same for both PEID and UM, the chain function is allowed and a chain request is sent to the next channel. If the settings are not the same for either PEID or UM, a chain request is not sent.

8.6 Setting Up DMA Transfer

8.6.1 Overview of Setting Up DMA

Table 8.7 Channel Assignment

No.	Master that configures the setting	Description	Register		Necessity of the setting
1	Special master (CPU in the supervisor mode)	Overall DMA operation setting	DM00CM to DM07CM, DM10CM to DM17CM	DMAC channel master setting	Mandatory (if a DMAC is used)
2		Status clear	CMVC	Channel protection violation clear register	Recommended
3	Master assigned to the DMAC channel	Channel setting	DSAn	DMAC source address	Mandatory
4			DDAn	DMAC destination address	Mandatory
5			DTCn	DMAC transfer count	Mandatory
6			DTCTn	DMAC transfer control	Mandatory
7			DRSAn	DMAC reload source address	Mandatory if the reload function is used
8			DRDAn	DMAC reload destination address	Mandatory if the reload function is used
9			DRTCn	DMAC reload transfer count	Mandatory if the reload function is used
10			DTCCn	DMAC transfer count compare	Mandatory if the transfer count match interrupt is used
11			DTFRn	DTFR setting register	Mandatory
12		Status clear	DCSTCn	DMAC transfer status clear	Mandatory
13			DTFRRQCn	DTFR transfer request clear	Recommended
14		Channel operation enable	DCENn	DMAC channel operation enable setting	Mandatory

8.6.2 Setting Up the Overall DMA Operation

You need to set up the overall DMA operation before you start using DMA.

To configure the overall DMA operation, the special master (a CPU in the supervisor mode) needs to set up global registers. Global registers can be set up only by special master access. For details, see Section 8.5, Reliability Function.

The following register must be set up to configure the overall DMA operation.

- DMAC channel master setting registers (DMnnCM)

If the DMAC channel master setting registers are not properly set, DMA channel setting and DMA transfer cannot be executed properly.

Also, if errors are detected in the following registers while the overall DMA operation is set up, clearing the errors is recommended.

- DMAC0 register access protection violation register (DM0CMV)
- DMAC1 register access protection violation register (DM1CMV)

8.6.3 Setting Up the DMA Channel Setting

The DMA channel setting defines the transfer information and transfer source for each DMAC channel.

To configure the DMA channel setting, the master assigned to each channel by the channel assignment needs to set up channel registers.

Follow the procedure below to set up the DMAC channel setting and use the DMAC.

(1) Disabling the DMAC Channel Operation

If the channel operation enable (DTE) in the DMAC channel operation enable setting register (DCENn) is set, clear the DTE bit to disable the channel operation.

(2) Setting Up the Transfer Information

When you set up the transfer information of the DMAC, the following registers need to be set up.

- DMAC source address register (DSAn)
- DMAC destination address register (DDAn)
- DMAC transfer count register (DTCn)
- DMAC transfer control register (DTCTn)
- DMAC reload source address register (DRSAn)
- DMAC reload destination address register (DRDAn)
- DMAC reload transfer count register (DRTCn)
- DMAC transfer count compare register (DTCCn)

(3) Setting Up the DMA Transfer Request

While setting the transfer information, you need to set up the DMA transfer request selection assignment (DTCTn.DRS) bit in the DMAC transfer control register (DTCTn) to define whether the hardware or software DMA transfer request is used.

You cannot use both the hardware and software DMA transfer requests for the same channel at the same time.

If you use the hardware DMA transfer request, you need to select one source used as the hardware DMA transfer request out of 128 hardware DMA transfer sources using the hardware DMA transfer source selection (DTFRn.REQSEL[6:0]) in the DTFR setting register. Also, you need to enable the hardware DMA transfer source selection (DTFRn.REQEN) in the same register.

The DTFR may retain a hardware DMA transfer request that came before the hardware DMA transfer source is selected. Clear the hardware DMA transfer request (DTFRRQn.DRQ) retained in the DTFR using the DTFR transfer request clear register (DTFRRQCn) if necessary.

If you use the software DMA transfer request, disable the hardware DMA transfer source selection (DTFRn.REQEN) in the DTFR setting register.

(4) Clearing the Transfer Status

The DMAC transfer status register (DCSTn) may retain the result of the previous DMA transfer. You need to clear the flags in the DMAC transfer status register using the DMAC transfer status clear register (DCSTCn).

(5) Enabling the DMAC Channel Operation

Set the channel operation enable (DCENn.DTE) bit in the DMAC channel operation enable setting register to enable the channel operation.

After the channel operation enable bit is set, the DMAC can accept a DMA transfer request and start DMA transfer.

8.7 DMA Trigger Source

The DMA trigger source ID assignment for DMA channel n is set in the DTFR setting register (DTFRn).

Table 8.8 List of DMA Trigger Sources (1/4)

DMA trigger ID	DMA trigger factor	Module	DMA Trigger Source
0	INTP0	Port	External interrupts
1	INTP1		
2	INTP2		
3	INTP3		
4	INTP4		
5	INTP5		
6	INTP6		
7	INTP7		
8	INTP8		
9	INTP9		
10	INTP10		
11	Reserved	–	–
12	INTLCBI0RDY	LCBI0	Read data ready interrupt
13	INTLCBI0EMPTY		Write buffer empty interrupt
14	INTLCBI0HALF		Write buffer half full interrupt
15	INTLCBI0FULL		Write buffer full interrupt
16	INTLCBI0QTR		Write buffer quarter full interrupt
17	INTLCBI03QTR		Write buffer three quarters full interrupt

Table 8.8 List of DMA Trigger Sources (2/4)

DMA trigger ID	DMA trigger factor	Module	DMA Trigger Source
18	INTPWGA0	PWGA	PWM Generator 0 interrupt
19	INTPWGA1		PWM Generator 1 interrupt
20	INTPWGA2		PWM Generator 2 interrupt
21	INTPWGA3		PWM Generator 3 interrupt
22	INTPWGA4		PWM Generator 4 interrupt
23	INTPWGA5		PWM Generator 5 interrupt
24	INTPWGA6		PWM Generator 6 interrupt
25	INTPWGA7		PWM Generator 7 interrupt
26	INTPWGA8		PWM Generator 8 interrupt
27	INTPWGA9		PWM Generator 9 interrupt
28	INTPWGA10		PWM Generator 10 interrupt
29	INTPWGA11		PWM Generator 11 interrupt
30	INTPWGA12		PWM Generator 12 interrupt
31	INTPWGA13		PWM Generator 13 interrupt
32	INTPWGA14		PWM Generator 14 interrupt
33	INTPWGA15		PWM Generator 15 interrupt
34	INTPWGA16		PWM Generator 16 interrupt
35	INTTAUB0I0	TAUB0	Channel 0 interrupt
36	INTTAUB0I1		Channel 1 interrupt
37	INTTAUB0I2		Channel 2 interrupt
38	INTTAUB0I3		Channel 3 interrupt
39	INTTAUB0I4		Channel 4 interrupt
40	INTTAUB0I5		Channel 5 interrupt
41	INTTAUB0I6		Channel 6 interrupt
42	INTTAUB0I7		Channel 7 interrupt
43	INTTAUB0I8		Channel 8 interrupt
44	INTTAUB0I9		Channel 9 interrupt
45	INTTAUB0I10		Channel 10 interrupt
46	INTTAUB0I11		Channel 11 interrupt
47	INTTAUB0I12		Channel 12 interrupt
48	INTTAUB0I13		Channel 13 interrupt
49	INTTAUB0I14		Channel 14 interrupt
50	INTTAUB0I15		Channel 15 interrupt

Table 8.8 List of DMA Trigger Sources (3/4)

DMA trigger ID	DMA trigger factor	Module	DMA Trigger Source
51	INTTAUB1I0	TAUB1	Channel 0 interrupt
52	INTTAUB1I1		Channel 1 interrupt
53	INTTAUB1I2		Channel 2 interrupt
54	INTTAUB1I3		Channel 3 interrupt
55	INTTAUB1I4		Channel 4 interrupt
56	INTTAUB1I5		Channel 5 interrupt
57	INTTAUB1I6		Channel 6 interrupt
58	INTTAUB1I7		Channel 7 interrupt
59	INTTAUB1I8		Channel 8 interrupt
60	INTTAUB1I9		Channel 9 interrupt
61	INTTAUB1I10		Channel 10 interrupt
62	INTTAUB1I11		Channel 11 interrupt
63	INTTAUB1I12		Channel 12 interrupt
64	INTTAUB1I13		Channel 13 interrupt
65	INTTAUB1I14		Channel 14 interrupt
66	INTTAUB1I15		Channel 15 interrupt
67	INTTAUB2I0	TAUB2	Channel 0 interrupt
68	INTTAUB2I1		Channel 1 interrupt
69	INTTAUB2I2		Channel 2 interrupt
70	INTTAUB2I3		Channel 3 interrupt
71	INTTAUB2I4		Channel 4 interrupt
72	INTTAUB2I5		Channel 5 interrupt
73	INTTAUB2I6		Channel 6 interrupt
74	INTTAUB2I7		Channel 7 interrupt
75	INTTAUB2I8		Channel 8 interrupt
76	INTTAUB2I9		Channel 9 interrupt
77	INTTAUB2I10		Channel 10 interrupt
78	INTTAUB2I11		Channel 11 interrupt
79	INTTAUB2I12		Channel 12 interrupt
80	INTTAUB2I13		Channel 13 interrupt
81	INTTAUB2I14		Channel 14 interrupt
82	INTTAUB2I15		Channel 15 interrupt
83	INTTAUJ0I0	TAUJ0	Channel 0 interrupt
84	INTTAUJ0I1		Channel 1 interrupt
85	INTTAUJ0I2		Channel 2 interrupt
86	INTTAUJ0I3		Channel 3 interrupt
87	INTRLIN30UR0	RLIN30	Transmit interrupt
88	INTRLIN30UR1		Receive completion interrupt
89	INTRLIN31UR0	RLIN31	Transmit interrupt
90	INTRLIN31UR1		Receive completion interrupt
91	INTRLIN32UR0	RLIN32	Transmit interrupt
92	INTRLIN32UR1		Receive completion interrupt

Table 8.8 List of DMA Trigger Sources (4/4)

DMA trigger ID	DMA trigger factor	Module	DMA Trigger Source
93	INTRLIN33UR0	RLIN33	Transmit interrupt
94	INTRLIN33UR1		Receive completion interrupt
95	INTCSIG0IC	CSIG0	Communication status interrupt
96	INTCSIG0IR		Reception status interrupt
97	INTCSIG1IC	CSIG1	Communication status interrupt
98	INTCSIG1IR		Reception status interrupt
99	INTCSIG2IC	CSIG2	Communication status interrupt
100	INTCSIG2IR		Reception status interrupt
101	INTCSIG3IC	CSIG3	Communication status interrupt
102	INTCSIG3IR		Reception status interrupt
103	INTCSIH0IC	CSIH0	Communication status interrupt
104	INTCSIH0IR		Reception status interrupt
105	INTCSIH0IJC		Job completion interrupt
106	INTCSIH1IC	CSIH1	Communication status interrupt
107	INTCSIH1IR		Reception status interrupt
108	INTCSIH1IJC		Job completion interrupt
109	INTRIIC0TI	RIIC0	Data buffer empty interrupt
110	INTRIIC0RI		Receive end interrupt
111	INTRIIC1TI	RIIC1	Transmit end interrupt
112	INTRIIC1RI		Receive end interrupt
113	INTSSIF0TX	SSIF0	Transmission data empty
114	INTSSIF0RX		Reception data full
115	INTSSIF1TX	SSIF1	Transmission data empty
116	INTSSIF1RX		Reception data full
117	INTPCMP0FFIL	PCMP0	FIFO buffer fill interrupt
118	INTSG0TI	SG0	Threshold interrupt
119	INTSG1TI	SG1	Threshold interrupt
120	INTSG2TI	SG2	Threshold interrupt
121	INTSG3TI	SG3	Threshold interrupt
122	INTSG4TI	SG4	Threshold interrupt
123	INTADCE0I1	ADCE0	Scan group 1 interrupt
124	INTADCE0I2		Scan group 2 interrupt
125	INTADCE0I3		Scan group 3 interrupt
126	DMA_WR_REQ	ICUS2	CMD registers are ready to write
127	DMA_RD_REQ		CMD registers are ready to read

8.8 Global Register

CAUTION

The DMA global registers are not accessible per default as these registers are guarded by the PBUS Guard PBG0B.

Before accessing any DMA global register the PBG0B protection register FSGD0BPROT1 has to be set to 07FF FFFF$_H$.

See Section 14.6.4, PBUS Guards (PBG) for further details.

8.8.1 List of Global Register Addresses

Address = Base address "FFFF 8000$_H$" + Offset address

Table 8.9 List of Global Register Addresses

Module Name	Offset Address	Symbol	Meaning	Access	
				Special Master	General Master
PDMA	0000$_H$	DMACTL	DMA control register	Allowed	Not allowed
PDMA	0020$_H$	DMACER	DMAC error register	Allowed	Allowed
PDMA	0030$_H$	DM0CMV	DMAC0 register access protection violation register	Allowed	Not allowed
PDMA	0034$_H$	DM1CMV	DMAC1 register access protection violation	Allowed	Not allowed
PDMA	003C$_H$	CMVC	Register access protection violation clear register	Allowed	Not allowed
PDMA	0100$_H$	DM00CM	DMAC0 channel 0 channel master setting	Allowed	Not allowed
PDMA	0104$_H$	DM01CM	DMAC0 channel 1 channel master setting	Allowed	Not allowed
PDMA	0108$_H$	DM02CM	DMAC0 channel 2 channel master setting	Allowed	Not allowed
PDMA	010C$_H$	DM03CM	DMAC0 channel 3 channel master setting	Allowed	Not allowed
PDMA	0110$_H$	DM04CM	DMAC0 channel 4 channel master setting	Allowed	Not allowed
PDMA	0114$_H$	DM05CM	DMAC0 channel 5 channel master setting	Allowed	Not allowed
PDMA	0118$_H$	DM06CM	DMAC0 channel 6 channel master setting	Allowed	Not allowed
PDMA	011C$_H$	DM07CM	DMAC0 channel 7 channel master setting	Allowed	Not allowed
PDMA	0120$_H$	DM10CM	DMAC1 channel 0 channel master setting	Allowed	Not allowed
PDMA	0124$_H$	DM11CM	DMAC1 channel 1 channel master setting	Allowed	Not allowed
PDMA	0128$_H$	DM12CM	DMAC1 channel 2 channel master setting	Allowed	Not allowed
PDMA	012C$_H$	DM13CM	DMAC1 channel 3 channel master setting	Allowed	Not allowed
PDMA	0130$_H$	DM14CM	DMAC1 channel 4 channel master setting	Allowed	Not allowed
PDMA	0134$_H$	DM15CM	DMAC1 channel 5 channel master setting	Allowed	Not allowed
PDMA	0138$_H$	DM16CM	DMAC1 channel 6 channel master setting	Allowed	Not allowed
PDMA	013C$_H$	DM17CM	DMAC1 channel 7 channel master setting	Allowed	Not allowed

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

8.8.2 Details of Global Registers

8.8.2.1 DMA Control Register (DMACTL)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8000_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMA SPD
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.10 DMACTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the initial value is read. When written, write the initial value.
0	DMASPD	DMA suspension This bit shows whether DMA transfer for all channels is suspended. If a user writes 1 to this bit, DMA transfer for all channels can be suspended. If a user writes 0 to this bit, suspension of DMA transfer for all channels can be cleared. The suspension controlled by this bit is independent from the suspension controlled by the transfer enable bit (DTE) of each DMAC channel. That means, if this bit is 1, all DMA transfers are suspended regardless of the values of the DTE bit of each channel. Writing to this bit does not affect the DTE bit of each channel. 0: DMA suspension cleared 1: DMA suspension request/DMA suspension ongoing

8.8.2.2 DMAC Error Register (DMACER)

Access: This register can be read in 32-bit units.

Address: FFFF 8020_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DM1 ER7	DM1 ER6	DM1 ER5	DM1 ER4	DM1 ER3	DM1 ER2	DM1 ER1	DM1 ER0	DM0 ER7	DM0 ER6	DM0 ER5	DM0 ER4	DM0 ER3	DM0 ER2	DM0 ER1	DM0 ER0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.11 DMACER register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the initial value is read. When written, write the initial value.
15 to 8	DM1ER[7:0]	DMAC1 DMA transfer error status These bits show the DMA transfer error status of channels 0 through 7 of the DMAC1. Each bit is mapped from the DCSTn.ER bit of each channel of the DMAC1 and is read-only. 0: DMA transfer error is not generated 1: DMA transfer error is generated
7 to 0	DM0ER[7:0]	DMAC0 DMA transfer error status These bits show the DMA transfer error status of channels 0 through 7 of the DMAC0. Each bit is mapped from the DCSTn.ER bit of each channel of the DMAC0 and is read-only. 0: DMA transfer error is not generated 1: DMA transfer error is generated

8.8.2.3 DMAC0 Register Access Protection Violation Register (DM0CMV)

Access: This register can be read in 32-bit units.

Address: FFFF 8030_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	MINF[6:4]			MINF[3:2]		MINF1	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	VCH2	VCH1	VCH0	—	—	—	VF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.12 DM0CMV register contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the initial value is read. When written, write the initial value.
22 to 17	MINF[6:1]	Illegal access master information These bits show the accessing master information of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. MINF6-4 retains the PEID of the accessing master, MINF3-2 retains the SPID of the accessing master, and MINF1 retains the UM of the accessing master.
16 to 7	Reserved	When read, the initial value is read. When written, write the initial value.
6 to 4	VCH[2:0]	Illegal access channel These bits show the channel number (0-7) of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared.
3 to 1	Reserved	When read, the initial value is read. When written, write the initial value.
0	VF	Illegal access flag This bit shows whether illegal access occurs in the DMAC0. 0: No illegal access has occurred in the DMAC0 1: Illegal access has occurred in the DMAC0 If illegal access occurs in the DMAC0 while this bit is 0, this bit is set, and MINF6 to 1 and VCH2 to 0 store their respective information. If illegal access occurs in the DMAC0 while this bit is 1, this bit remains 1, and MINF6 to 1 and VCH2 to 0 do not change. This bit can be cleared by using the CMVC register.

8.8.2.4 DMAC1 Register Access Protection Violation Register (DM1CMV)

Access: This register can be read in 32-bit units.

Address: FFFF 8034_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	MINF[6:4]			MINF[3:2]		MINF1	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	VCH2	VCH1	VCH0	—	—	—	VF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.13 DM1CMV register contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the initial value is read. When written, write the initial value.
22 to 17	MINF[6:1]	Illegal access master information These bits show the accessing master information of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared. MINF6-4 retains the PEID of the accessing master, MINF3-2 retains the SPID of the accessing master, and MINF1 retains the UM of the accessing master.
16 to 7	Reserved	When read, the initial value is read. When written, write the initial value.
6 to 4	VCH[2:0]	Illegal access channel These bits show the channel number (0-7) of the first illegal access after the VF bit is cleared to 0. If illegal access occurs while the VF bit is 1, these bits do not change. These bits are read-only and cannot be cleared.
3 to 1	Reserved	When read, the initial value is read. When written, write the initial value.
0	VF	Illegal access flag This bit shows whether illegal access occurs in the DMAC1. 0: No illegal access has occurred in the DMAC1 1: Illegal access has occurred in the DMAC1 If illegal access occurs in the DMAC1 while this bit is 0, this bit is set, and MINF 6 to 1 and VCH2 to 0 store their respective information. If illegal access occurs in the DMAC1 while this bit is 1, this bit remains 1, and MINF6 to 1 and VCH2 to 0 do not change. This bit can be cleared by using the CMVC register.

8.8.2.5 Register Access Protection Violation Clear Register (CMVC)

Access: This register can be read/written in 32-bit units.

Address: FFFF 803C$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DM1VC	DM0VC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 8.14 CMVC register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the initial value is read. When written, write the initial value.
1	DM1VC	DMAC1 illegal access flag clear The DMAC1 illegal access flag (DM1CMV.VF) can be cleared by writing 1 to this bit. The read value of this bit is always 0.
0	DM0VC	DMAC0 illegal access flag clear The DMAC0 illegal access flag (DM0CMV.VF) can be cleared by writing 1 to this bit. The read value of this bit is always 0.

8.8.2.6 DMAC Channel Master Setting (DMnnCM) (nn = 00 - 07, 10 - 17)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8100_H + 4 × Ch. No. n (n = 0 to 7)
FFFF 8120_H + 4 × Ch. No. n - 10 (n = 10 to 17)

Initial value: 0000 0010_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	PEID2	PEID1	PEID0	SPID1	SPID0	UM	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 8.15 DMnnCM register contents

Bit Position	Bit Name	Function ion
31 to 7	Reserved	When read, the initial value is read. When written, write the initial value.
6 to 4	PEID[2:0]	Channel master PEID setting Specifies the PEID information of the master assigned to the channel. This register must be set to a value other than 0.
3, 2	SPID[1:0]	Channel master SPID setting Specifies the SPID information used by the master assigned to the channel.
1	UM	Channel master UM setting Specifies the UM information of the master assigned to the channel.
0	Reserved	When read, the initial value is read. When written, write the initial value.

CAUTION

DM00CM - DM07CM configure the channel master information of the DMAC0 channel 0-7 respectively.
DM10CM - DM17CM configure the channel master information of the DMAC1 channel 0-7 respectively.

For information about the functions this register offers, see Section 8.5, Reliability Function.

8.9 DMAC Channel Register

CAUTION

The DMA channel registers are not accessible per default as these registers are guarded by the PBUS Guard PBG0B.

Before accessing any DMA channel register the PBG0B protection register FSGD0BPROT1 has to be set to 07FF FFFF$_H$.

See Section 14.6.4, PBUS Guards (PBG) for further details.

8.9.1 DMAC Channel Register Address

Address = Base address "FFFF 8000$_H$" + Offset address

Table 8.16 DMAC Channel Register Address

Module Name	Offset Address	Symbol	Meaning	Access	
				Special Master	General Master
PDMA	0400$_H$ + 40$_H$ × [channel number]	DSAn	DMAC source address	Allowed	Allowed
PDMA	0404$_H$ + 40$_H$ × [channel number]	DDAn	DMAC destination address	Allowed	Allowed
PDMA	0408$_H$ + 40$_H$ × [channel number]	DTCn	DMAC transfer count	Allowed	Allowed
PDMA	040C$_H$ + 40$_H$ × [channel number]	DTCTn	DMAC transfer control	Allowed	Allowed
PDMA	0410$_H$ + 40$_H$ × [channel number]	DRSAn	DMAC reload source address	Allowed	Allowed
PDMA	0414$_H$ + 40$_H$ × [channel number]	DRDAn	DMAC reload destination address	Allowed	Allowed
PDMA	0418$_H$ + 40$_H$ × [channel number]	DRTCn	DMAC reload transfer count	Allowed	Allowed
PDMA	041C$_H$ + 40$_H$ × [channel number]	DTCCn	DMAC transfer count compare	Allowed	Allowed
PDMA	0420$_H$ + 40$_H$ × [channel number]	DCENn	DMAC channel operation enable setting	Allowed	Allowed
PDMA	0424$_H$ + 40$_H$ × [channel number]	DCSTn	DMAC transfer status	Allowed	Allowed
PDMA	0428$_H$ + 40$_H$ × [channel number]	DCSTSn	DMAC transfer status set	Allowed	Allowed
PDMA	042C$_H$ + 40$_H$ × [channel number]	DCSTCn	DMAC transfer status clear	Allowed	Allowed
PDMA	0430$_H$ + 40$_H$ × [channel number]	DTFRn	DTFR setting	Allowed	Allowed
PDMA	0434$_H$ + 40$_H$ × [channel number]	DTFRRQn	DTFR transfer request status	Allowed	Allowed
PDMA	0438$_H$ + 40$_H$ × [channel number]	DTFRRQCn	DTFR transfer request clear	Allowed	Allowed

Note 1. The [channel number] in the offset addresses and "n" in the register symbols are numbers in the range from 0 to 15, and the correspondence between the channel number n and the channel is as follows.

Channel number n	Channel
0	DMAC0 channel 0
1	DMAC0 channel 1
2	DMAC0 channel 2
3	DMAC0 channel 3
4	DMAC0 channel 4
5	DMAC0 channel 5
6	DMAC0 channel 6
7	DMAC0 channel 7
8	DMAC1 channel 0
9	DMAC1 channel 1
10	DMAC1 channel 2

Channel number n	Channel
11	DMAC1 channel 3
12	DMAC1 channel 4
13	DMAC1 channel 5
14	DMAC1 channel 6
15	DMAC1 channel 7

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

8.9.2 Details of DMAC Channel Registers

The “n” in the register symbols indicates the DMAC channel number (n = 0 to 15).

8.9.2.1 DMAC Source Address Register (DSAn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8400_H + 40_H × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SA31	SA30	SA29	SA28	SA27	SA26	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.17 DSAn register contents

Bit Position	Bit Name	Function
31 to 0	SA[31:0]	Source address Specifies the DMA transfer source address. Those bits are updated whenever a DMA cycle is executed. If you read from those bits, the transfer source address for the next DMA cycle is read.

CAUTIONS

1. It is forbidden to write to those bits when the channel operation is enabled (DTE bit = 1). If you do, the correct operation is not guaranteed.
2. The address must be set up while the DTE bit is 0.
3. DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (× denotes an arbitrary one bit.)
   The correct operation is not guaranteed if you select any other setting.

Data Size	SA3	SA2	SA1	SA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
64 bits	×	0	0	0
128 bits	0	0	0	0

8.9.2.2 DMAC Destination Address Register (DDAn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8404_H + 40_H × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DA31	DA30	DA29	DA28	DA27	DA26	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.18 DDAn register contents

Bit Position	Bit Name	Function
31 to 0	DA[31:0]	Destination address Specifies the DMA transfer destination address. Those bits are updated whenever a DMA cycle is executed. If you read from those bits, the transfer destination address for the next DMA cycle is read.

CAUTIONS

1. It is forbidden to write to those bits when the channel operation is enabled (DTE bit = 1). If you do, the correct operation is not guaranteed.
2. The address must be set up while the DTE bit is 0.
3. If transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the destination address is updated.
4. DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (× denotes an arbitrary one bit.)
   The correct operation is not guaranteed if you select any other setting.

Data Size	DA3	DA2	DA1	DA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
64 bits	×	0	0	0
128 bits	0	0	0	0

8.9.2.3 DMAC Transfer Count Register (DTCn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8408_H + 40_H × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ARC15	ARC14	ARC13	ARC12	ARC11	ARC10	ARC9	ARC8	ARC7	ARC6	ARC5	ARC4	ARC3	ARC2	ARC1	ARC0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TRC15	TRC14	TRC13	TRC12	TRC11	TRC10	TRC9	TRC8	TRC7	TRC6	TRC5	TRC4	TRC3	TRC2	TRC1	TRC0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.19 DTCn register contents

Bit Position	Bit Name	Function
31 to 16	ARC[15:0]	Address reload count Specifies the number of transfers until the address reload when the reload function 2 is used, and also specifies the number of transfers when the block transfer 2 is used. If you read from those bits during DMA transfer, the address reload count for the next DMA cycle is read. When the reload function 2 or block transfer 2 is used, ARC[15:0] is decremented by one for every DMA cycle. When the reload function 2 or block transfer 2 is not used, ARC[15:0] is not updated. If the value is 0000_H, it means that the number of transfers until the address reload when the reload function 2 is used and the number of transfers when the block transfer 2 is used are 65536.
15 to 0	TRC[15:0]	Transfer count Configures the number of transfers. TRC[15:0] is decremented by one whenever a DMA cycle is executed. If you read from those bits, the remaining number of transfers for the next DMA cycle is read. If the reload function is not used, after the last transfer is complete, the value at the completion (0000_H) is retained.

TRC15-0	Operation
0000_H	The number of transfers is 65536, or the transfer is complete.
0001_H	The number of transfers or remaining transfers is 1.
:	:
$FFFF_H$	The number of transfers or remaining transfers is 65535.

CAUTIONS

1. It is forbidden to write to those bits when the channel operation is enabled (DTE bit = 1). If you do, the correct operation is not guaranteed.
2. If transfer error is generated in the read cycle of DMA transfer, the write cycle is not executed, but the transfer count and the address reload count are updated.

8.9.2.4 DMAC Transfer Control Register (DTCTn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 840C$_H$ + 40$_H$ × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	ESE	DRS	—	—	—	—	—	CHNSEL2	CHNSEL1	CHNSEL0	CHNE1	CHNE0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CCE	TCE	MLE	RLD2M1	RLD2M0	RLD1M1	RLD1M0	DACM1	DACM0	SACM1	SACM0	DS2	DS1	DS0	TRM1	TRM0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.20 DTCTn register contents (1/3)

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, the initial value is read. When written, write the initial value.
27	ESE	Transfer error case DMA transfer disable setting Configures whether a DMA cycle is executed when the DCSTn.ER bit is set due to DMA transfer error. If this bit is cleared to 0, even when the DCSTn.ER bit is set due to DMA transfer error, the following DMA cycles can be executed. If this bit is set to 1, the following DMA cycles are not executed when the DCSTn.ER bit is set due to DMA transfer error. 0: DMA cycles are executed while the DCSTn.ER bit is set. 1: DMA cycles are not executed while the DCSTn.ER bit is set.
26	DRS	DMA transfer request selection assignment Selects the type of DMA transfer requests to be accepted. 0: Software DMA transfer request 1: Hardware DMA transfer request
25 to 21	Reserved	When read, the initial value is read. When written, write the initial value.
20 to 18	CHNSEL[2:0]	Next channel in the chain Specifies the next channel in the chain. The next channel must be another channel in the same DMAC. You cannot specify a channel in the different DMAC. You cannot specify the same channel for the next channel. (If you do, the correct operation is not guaranteed.)
17, 16	CHNE[1:0]	Chain enable Selects the chain function. 00: Disabled 01: Chain at the last transfer A chain request is generated at the completion of the DMA cycle in which the remaining transfer count is one. 10: (Forbidden. No guarantee of operation) 11: Always chain A chain request is generated at the completion of every DMA cycle.
15	CCE	Transfer count match interrupt enable If this bit is set, a transfer count match interrupt is generated at the completion of the DMA cycle in which the transfer count compare register and the remaining transfer count have the same value.
14	TCE	Transfer completion interrupt enable If this bit is set, a transfer completion interrupt is generated at the completion of the last transfer. A transfer completion notification interrupt is also generated at the completion of the last transfer when DCSTn.TC = 1.

Table 8.20 DTCTn register contents (2/3)

Bit Position	Bit Name	Function
13	MLE	Continuous transfer enable If this bit is set, the DTE bit is not cleared at the completion of DMA transfer. In addition, even if the TC bit is not cleared, DMA transfer starts when there is a DMA transfer request. 0: The DTE bit is cleared at the completion of DMA transfer. In addition, the next DMA transfer can start only after the TC bit is cleared. 1: The DTE bit is not cleared at the completion of DMA transfer. In addition, even if the TC bit is not cleared, DMA transfer starts when there is a DMA transfer request.
12, 11	RLD2M[1:0]	Reload function 2 setting Configures the reload function 2. 00: Reload function 2 is disabled. 01: Reload function 2 is enabled. The source address and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1. 10: Reload function 2 is enabled. The destination address and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1. 11: Reload function 2 is enabled. The source address, destination address, and address reload count are reloaded at the completion of the DMA cycle in which the address reload count is 1.
10, 9	RLD1M[1:0]	Reload function 1 setting Configures the reload function 1. 00: Reload function 1 is disabled. 01: Reload function 1 is enabled. The source address and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.) 10: Reload function 1 is enabled. The destination address and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.) 11: Reload function 1 is enabled. The source address, destination address, and transfer count are reloaded at the completion of the DMA cycle in which the remaining transfer count is 1. (If the reload function 2 is enabled, the address reload count is also reloaded.)
8, 7	DACM[1:0]	Destination address count direction Specifies the count direction of the destination address.

DACM1	DACM0	Direction of Count
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Forbidden (No guarantee of operation)

6, 5	SACM[1:0]	Source address count direction Specifies the count direction of the source address.

SACM1	SACM0	Direction of Count
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Forbidden (No guarantee of operation)

Table 8.20 DTCTn register contents (3/3)

Bit Position	Bit Name	Function
4 to 2	DS[2:0]	Transfer data size Specifies the transfer data size.
1, 0	TRM[1:0]	Transfer mode Specifies the DMA transfer mode. 00: Single transfer 01: Block transfer 1 (The number of transfers is specified by the transfer count.) 10: Block transfer 2 (The number of transfers is specified by the address reload count.) 11: Forbidden (No guarantee of operation)

DS2	DS1	DS0	Transfer Data Size
0	0	0	8 bits
0	0	1	16 bits
0	1	0	32 bits
0	1	1	64 bits
1	0	0	128 bits
Other than the above			Forbidden (No guarantee of operation)

CAUTIONS

1. It is forbidden to write to those bits when the channel operation is enabled (DTE bit = 1), only MLE bit can be changed '1' to '0'. (If other setting done, the correct operation is not guaranteed.)
2. If forbidden settings are used for some of the bits, the correct operation is not guaranteed.

8.9.2.5 DMAC Reload Source Address Register (DRSAn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8410_H + 40_H × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RSA31	RSA30	RSA29	RSA28	RSA27	RSA26	RSA25	RSA24	RSA23	RSA22	RSA21	RSA20	RSA19	RSA18	RSA17	RSA16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RSA15	RSA14	RSA13	RSA12	RSA11	RSA10	RSA9	RSA8	RSA7	RSA6	RSA5	RSA4	RSA3	RSA2	RSA1	RSA0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.21 DRSAn register contents

Bit Position	Bit Name	Function
31 to 0	RSA[31:0]	Reload source address Specifies the source address to be reloaded to the DMA source address register when the reload function 1 or reload function 2 is used.

CAUTION

DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (× denotes an arbitrary one bit.) The correct operation is not guaranteed if you select any other setting.

Data Size	RSA3	RSA2	RSA1	RSA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
64 bits	×	0	0	0
128 bits	0	0	0	0

8.9.2.6 DMAC Reload Destination Address Register (DRDAn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8414_H + 40_H × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDA31	RDA30	RDA29	RDA28	RDA27	RDA26	RDA25	RDA24	RDA23	RDA22	RDA21	RDA20	RDA19	RDA18	RDA17	RDA16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDA15	RDA14	RDA13	RDA12	RDA11	RDA10	RDA9	RDA8	RDA7	RDA6	RDA5	RDA4	RDA3	RDA2	RDA1	RDA0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.22 DRDAn register contents

Bit Position	Bit Name	Function
31 to 0	RDA[31:0]	Reload destination address Specifies the destination address to be reloaded to the DMA destination address register when the reload function 1 or reload function 2 is used.

CAUTION

DMA transfer for misaligned data is not supported. The possible lowest four bits of the address for each transfer data size is as follows. (× denotes an arbitrary one bit.) The correct operation is not guaranteed if you select any other setting.

Data Size	RDA3	RDA2	RDA1	RDA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
64 bits	×	0	0	0
128 bits	0	0	0	0

8.9.2.7 DMAC Reload Transfer Count Register (DRTCn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8418_H + 40_H × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RARC15	RARC14	RARC13	RARC12	RARC11	RARC10	RARC9	RARC8	RARC7	RARC6	RARC5	RARC4	RARC3	RARC2	RARC1	RARC0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTRC15	RTRC14	RTRC13	RTRC12	RTRC11	RTRC10	RTRC9	RTRC8	RTRC7	RTRC6	RTRC5	RTRC4	RTRC3	RTRC2	RTRC1	RTRC0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.23 DRTCn register contents

Bit Position	Bit Name	Function
31 to 16	RARC[15:0]	Reload address reload count Specifies the value to be loaded to the address reload count in the transfer count register at the time of reload when the reload function 2 is used.
15 to 0	RTRC[15:0]	Reload transfer count Specifies the value to be loaded to the transfer count in the transfer count register at the time of reload when the reload function 1 or reload function 2 is used.

8.9.2.8 DMAC Transfer Count Compare Register (DTCCn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 841C$_H$ + 40$_H$ × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMC15	CMC14	CMC13	CMC12	CMC11	CMC10	CMC9	CMC8	CMC7	CMC6	CMC5	CMC4	CMC3	CMC2	CMC1	CMC0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.24 DTCCn register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the initial value is read. When written, write the initial value.
15 to 0	CMC[15:0]	Transfer count compare Configures the transfer count to be compared to the transfer count register. At the completion of the DMA cycle in which the remaining transfer count is the same as the value in this register, the transfer count match flag (DCSTn.CC) in the DMAC transfer status register is set. Furthermore, if the transfer count match interrupt enable (DTCTn.CCE) bit is 1, a transfer count match interrupt is generated. If 0000$_H$ is set, comparison with the transfer count is disabled. In this case, the transfer count match flag in the DMAC transfer status register is never set, and a transfer count match interrupt is never generated.

CAUTION

It is forbidden to write to those bits when the channel operation is enabled (DTE bit = 1). If you do, the correct operation is not guaranteed.

8.9.2.9 DMAC Channel Operation Enable Setting Register (DCENn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8420_H + 40_H × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.25 DCENn register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the initial value is read. When written, write the initial value.
0	DTE	Channel operation enable Specifies whether to enable or disable the transfer operation of the channel. If the DTE bit is 1, DMA transfer starts when there is a DMA transfer request. If the MLE bit is 0, the DTE bit is cleared automatically at the completion of the DMA transfer. In addition, if 0 is written to the DTE bit during DMA transfer, the DMA transfer is suspended. If 1 is written to the DTE bit during suspension, the suspension is cleared and the DMA transfer resumes. 0: Channel operation is disabled/Channel suspended 1: Channel operation is enabled/Channel suspension cleared

8.9.2.10 DMAC Transfer Status Register (DCSTn)

Access: This register can be read in 32-bit units.

Address: FFFF 8424_H + 40_H × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	ERWR	—	—	CY	ER	—	CC	TC	—	—	DR	SR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.26 DCSTn register contents (1/2)

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, the initial value is read. When written, write the initial value.
11	ERWR	DMA Transfer Error occurring cycle This bit is updated at the same time as setting of the DMA transfer error flag (ER), indicating which cycle of read or write the DMA transfer error occurs in. This bit is not updated when a new DMA transfer error occurs after the ER bit has been set. If the ER bit is cleared, this bit is also cleared to 0. 0: DMA transfer error occurs in read cycle. 1: DMA transfer error occurs in write cycle.
10, 9	Reserved	When read, the initial value is read. When written, write the initial value.
8	CY	DMA cycle execution state This bit shows whether a DMA cycle is ongoing in this channel. 0: DMA cycle is not ongoing. 1: DMA cycle is ongoing.
7	ER	Transfer error flag This bit is set when DMA transfer error is generated. If this bit is 1 and the DTCTn.ESE bit is set, a DMA cycle is not executed when a DMA transfer request is generated. 0: No DMA transfer error is generated 1: DMA transfer error is generated
6	Reserved	When read, the initial value is read. When written, write the initial value.
5	CC	Transfer count match flag This bit is set at the completion of the DMA cycle in which the remaining transfer count is the same as the value set in the transfer compare register. 0: No compare match has occurred with the transfer count compare register. 1: Compare match has occurred with the transfer count compare register.
4	TC	Transfer completion flag This bit is set at the completion of the last transfer and shows whether the DMA transfer is complete. If the MLE bit is 0 and this bit is 1, a DMA cycle is not executed when a DMA transfer request is generated. 0: DMA transfer incomplete 1: DMA transfer complete
3, 2	Reserved	When read, the initial value is read. When written, write the initial value.

Table 8.26 DCSTn register contents (2/2)

Bit Position	Bit Name	Function
1	DR	Hardware DMA transfer request status This bit shows whether there is a hardware DMA transfer request (DMARQ) from the DTFR. This bit changes regardless of the value of the DTE bit when a hardware DMA transfer request from the DTFR is generated. If the software DMA transfer request is selected by the transfer request selection bit (DRS) in the DMAC transfer control register, this bit is not set even when a hardware DMA transfer request is input from the DTFR. 0: There is no hardware DMA transfer request 1: There is a hardware DMA transfer request
0	SR	Software DMA transfer request flag This bit shows whether there is a software DMA transfer request (DMARQ). This bit is automatically cleared at the completion of the last transfer. A user can set this bit by writing 1 to the SRS bit in the DMAC transfer status set register. In addition, a user can clear this bit by writing 1 to the SRC bit in the DMAC transfer status clear register, but if this is done, the ongoing DMA transfer is aborted and cannot be resumed. 0: There is no software DMA transfer request 1: There is a software DMA transfer request

8.9.2.11 DMAC Transfer Status Set Register (DCSTSn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8428_H + 40_H × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SRS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.27 DCSTSn Set register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the initial value is read. When written, write the initial value.
0	SRS	Software DMA transfer request flag A user can set the software DMA transfer request flag (SR) by writing 1 to this bit. 0 is always read from this bit.

8.9.2.12 DMAC Transfer Status Clear Register (DCSTCn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 842C$_H$ + 40$_H$ × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ERC	—	CCC	TCC	—	—	—	SRC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R	R	R	R/W

Table 8.28 DCSTCn register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the initial value is read. When written, write the initial value.
7	ERC	Transfer error flag clear The DMA transfer error flag (ER) can be cleared by writing 1 to this bit. The read value of this bit is always 0.
6	Reserved	When read, the initial value is read. When written, write the initial value.
5	CCC	Transfer count match flag clear The transfer count match flag (CC) can be cleared by writing 1 to this bit. The read value of this bit is always 0.
4	TCC	Transfer completion flag clear The transfer completion flag (TC) can be cleared by writing 1 to this bit. The read value of this bit is always 0.
3 to 1	Reserved	When read, the initial value is read. When written, write the initial value.
0	SRC	Software DMA transfer request flag clear The software DMA transfer request flag (SR) can be cleared by writing 1 to this bit. The read value of this bit is always 0.

8.9.2.13 DTFR Setting Register (DTFRn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8430_H + 40_H × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	REQSEL6	REQSEL5	REQSEL4	REQSEL3	REQSEL2	REQSEL1	REQSEL0	REQEN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8.29 DTFRn register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the initial value is read. When written, write the initial value.
7 to 1	REQSEL[6:0]	Hardware DMA transfer source selection Selects one out of 128 hardware DMA transfer sources as the hardware DMA transfer request. 000_0000: Selecting the DMACTRG[0] input : 111_1111: Selecting the DMACTRG[127] input
0	REQEN	Hardware DMA transfer source selection enable This bit enables/disables the hardware DMA transfer source selection. 0: Hardware DMA transfer source selection is disabled. 1: Hardware DMA transfer source selection is enabled. If this bit is 0, even when the hardware DMA transfer source selected by the REQSEL6-0 bits is activated, it is not recognized as a hardware DMA transfer request, and a hardware DMA transfer request is not generated.

8.9.2.14 DTFR Transfer Request Status Register (DTFRRQn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8434_H + 40_H × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQ
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8.30 DTFRRQn register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the initial value is read. When written, write the initial value.
0	DRQ	Hardware DMA transfer request status If this bit is set, it means that a hardware DMA transfer request exists or is retained. • If the hardware DMA transfer request is an edge detection type*1 This bit shows whether a hardware DMA transfer request generated by edge detection is retained. When the DMA transfer request acceptance signal (DMAAKn) from the DMAC is asserted, this bit is automatically cleared. A user can clear this bit by writing 1 to the DTFRRQCn.DRQC bit. • If the hardware DMA transfer request is a level input type*1 This bit shows whether there is a hardware DMA transfer request input from the outside. Even when the DMA transfer request acceptance signal (DMAAKn) from the DMAC is asserted, this bit is not automatically cleared. In addition, this bit is not cleared even when a user writes to the DTFRRQCn.DRQC bit. This bit changes regardless of the value of the DTFRn.REQEN bit when a hardware DMA transfer request from the outside is generated. 0: There is no hardware DMA transfer request 1: There is a hardware DMA transfer request

Note 1. Whether the hardware DMA transfer request is an edge detection type or level input type depends on the hardware DMA transfer source selected by DTFRn.REQSEL[6:0]. This product's hardware DMA transfer request is only an edge detection type.

8.9.2.15 DTFR Transfer Request Clear Register (DTFRRQCn)

Access: This register can be read/written in 32-bit units.

Address: FFFF 8438_H + 40_H × Ch. No. n (n = 0 to 15)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRQC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 8.31 DTFRRQCn register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the initial value is read. When written, write the initial value.
0	DRQC	Hardware DMA transfer request clear If the hardware DMA transfer request is an edge detection type*1, a user can clear the DTFRRQn.DRQ bit by writing 1 to this bit. If the hardware DMA transfer request is a level input type*1, the DTFRRQn.DRQ bit cannot be cleared by writing to this bit. The read value of this bit is always 0.

Note 1. Whether the hardware DMA transfer request is an edge detection type or level input type depends on the hardware DMA transfer source selected by DTFRn.REQSEL[6:0]. This product's hardware DMA transfer request is only an edge detection type.

Section 9 Reset Controller

9.1 Overview

Several system reset functions are provided in order to initialize the microcontroller hardware and its registers.

A reset can be caused by the following events:

- External reset signal $\overline{\text{RESET}}$
  Noise in the external reset signal is eliminated by an analog filter.
- Power-On-Clear 0 (POC0RES)
- Power-On-Clear 1 (POC1RES) (not for D1M2(H))
- External power-good indicator signal (PWRGD)
- Overflow of the Watchdog Timers (WDTA0RES, WDTA1RES)
- Clock Monitor 0 reset ($\overline{\text{CLMA0RES}}$)
- Error Control Module reset output (ECMRES)
- Software reset (SWRES)
- Debugger reset ($\overline{\text{DBRES}}$)
- DEEPSTOP mode (ISORES)

9.1.1 Clock Supply

The Reset Controller clock supply is shown in the following table.

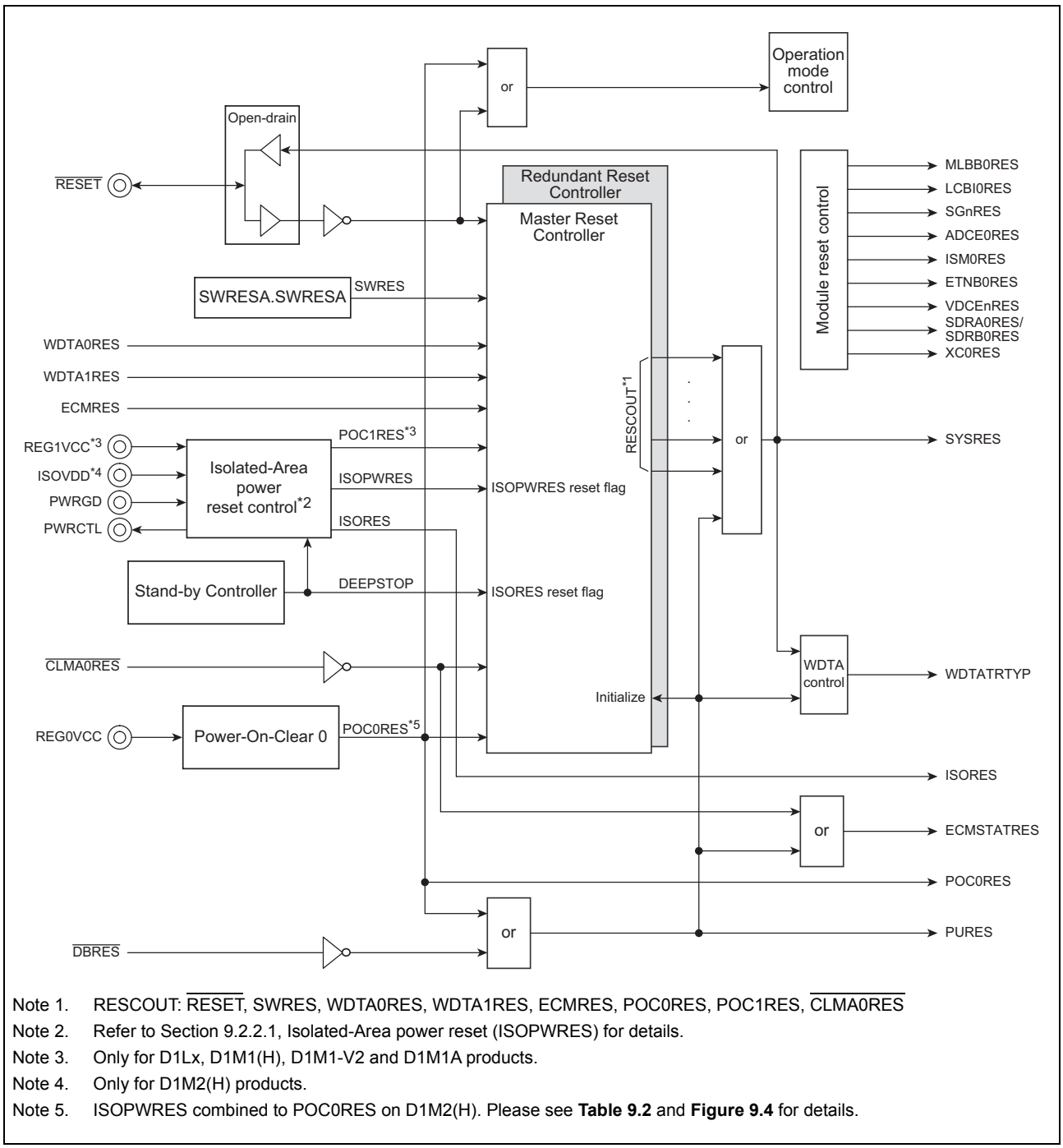
Table 9.1 Clock Supply

Module	Clock	Connected to
Reset Controller	Register access clock	Clock Controller • C_ISO_PCLK in RUN mode • EMCLK in DEEPSTOP mode

9.2 Configuration

9.2.1 Block Diagram

The block diagram of reset circuitry is shown below.



Note 1. RESCOUT: $\overline{\text{RESET}}$, SWRES, WDTA0RES, WDTA1RES, ECMRES, POC0RES, POC1RES, $\overline{\text{CLMA0RES}}$

Note 2. Refer to Section 9.2.2.1, Isolated-Area power reset (ISOPWRES) for details.

Note 3. Only for D1Lx, D1M1(H), D1M1-V2 and D1M1A products.

Note 4. Only for D1M2(H) products.

Note 5. ISOPWRES combined to POC0RES on D1M2(H). Please see **Table 9.2** and **Figure 9.4** for details.

Figure 9.1 Block diagram of the Reset Controller

9.2.2 Reset sources and targets

Table 9.2 Reset sources and targets overview

Power domain	Reset target	Reset source				
		Reset level 1 (PURES)		Reset level 2 (SYSRES)		Reset level 3 (ISORES)
		POC0RES, ISOPWRES*3	$\overline{\text{DBRES}}$	$\overline{\text{CLMA0RES}}$	POC1RES*2, $\overline{\text{RESET}}$, WDTAnRES, ECMRES, SWRES, ISOPWRES*4	ISORES
AWO	Watchdog Timer WDTA0	reset	reset	reset	reset	not reset
	Real-Time Clock RTCA0	reset	reset	reset	reset	not reset
	Always-On-Area Timer AWOT0	reset	reset	reset	reset	not reset
	Clock Monitors CLMA0 to CLMA2	reset	reset	reset	reset	not reset
	Error Control Module, except error status registers	reset	reset	reset	reset	not reset
	Error Control Module error status registers	reset	reset	reset	not reset	not reset
	Low Speed IntOsc	reset	reset	not reset	not reset	not reset
	High Speed IntOsc	reset	reset	reset	not reset	not reset
	MainOsc, SubOsc	reset	reset	not reset	not reset	not reset
	Always-On-Area ports	reset	reset	reset	reset	not reset
	RRAM content	not reset	not reset	not reset	not reset	not reset
ISO	All Isolated-Area modules*1	reset	reset*5	reset*5	reset*5	reset*5
	Isolated-Area ports	reset	reset	reset	reset	reset

Note 1. Including CPU Subsystem, PLL clock generators.
Note 2. If POC1RES is asserted during DEEPSTOP mode (i.e. because REG1VCC was switched off), it has no effect.
Note 3. In D1M2(H)
Note 4. In D1Lx, D1M1-V2, D1M1(H) and D1M1A
Note 5. LRAM content keep in this reset (except POC1RES and ISOPWRES)

The various reset sources are assigned to different hierarchical reset levels.
The reset sources of each reset level affect different reset targets:

- Reset level 1: power-up reset PURES
  Level 1 resets initialize basically the entire microcontroller.
- Reset level 2: system reset SYSRES
  Level 2 resets initialize all microcontroller modules and ports, except the oscillator circuits. This ensures fast return to normal operation, since no waiting for oscillators stabilization necessary after level 2 reset release.
  A level 2 reset also retains the content of the Error Control Module's error status register.
- Reset level 3: Isolated-Area reset ISORES
  A level 3 resets is asserted during DEEPSTOP mode and initialization the entire Isolated-Area.

9.2.2.1 Isolated-Area power reset (ISOPWRES)

The Isolated-Area power reset is asserted, when the power supply of the Isolated-Area fails.

Since switch-off of the Isolated-Area power supply during DEEPSTOP stand-by mode is a regular operation condition, ISOPWRES is never asserted during DEEPSTOP, i.e. while the Stand-by Controller activates the Isolated-Area reset ISORES. Refer also to the figures below.

(1) D1Lx, D1M1(H), D1M1-V2 and D1M1A Isolated-Area reset

If the device is not in DEEPSTOP mode (DEEPSTOP = 0) an Isolated-Area power reset (i.e. ISOPWRES = 1) is asserted if

- or POC1RES is active (POC1RES = 1) due to a fail of REG1VCC
- or the power-good signal PWRGD is externally set to low level, indicating unstable REG1VCC

Assertion and de-assertion of ISORES is managed by the Isolated-Area reset control, refer to Section (4), Isolated-Area reset control (D1L1, D1L2(H), D1M1(H), D1M1-V2, D1M1A) for details.

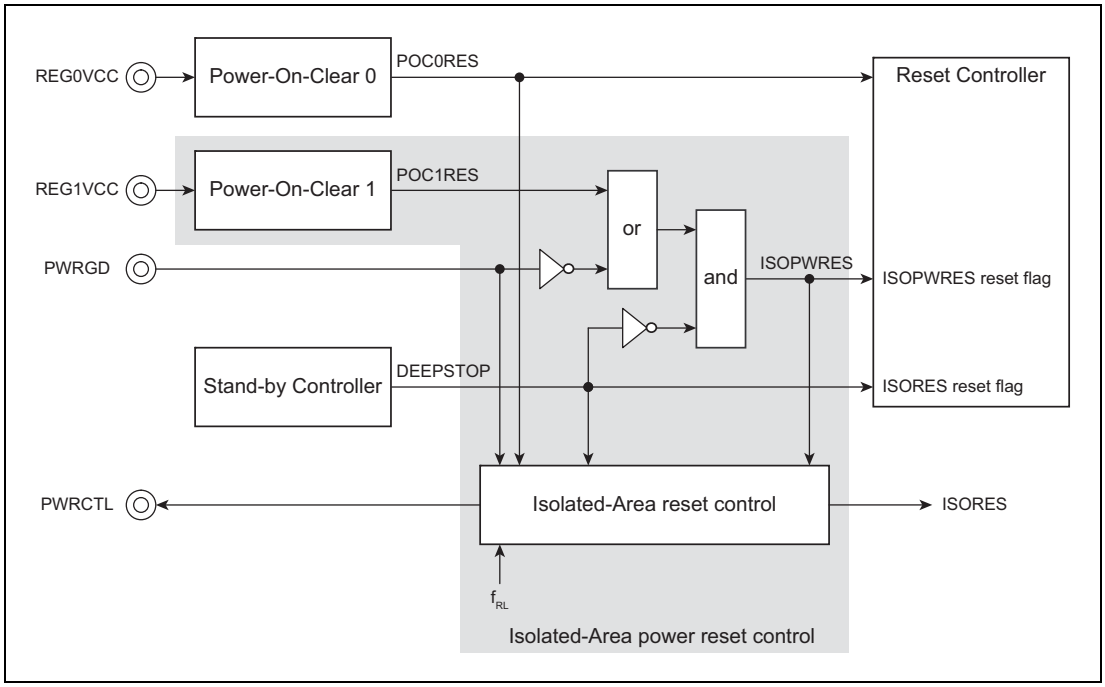


Figure 9.2 D1Lx, D1M1(H), D1M1-V2 and D1M1A Isolated-Area power reset

(2) D1M2(H) Isolated-Area power reset

If the device is not in DEEPSTOP mode (ISORES = 0) and D1M2(H) issues POC0RES when PWRGD signals set to low before PWRCTL change from high to low.

Assertion and de-assertion of ISORES is managed by the Isolated-Area reset control, refer to Section (3), Isolated-Area reset control (D1M2(H)) for details.

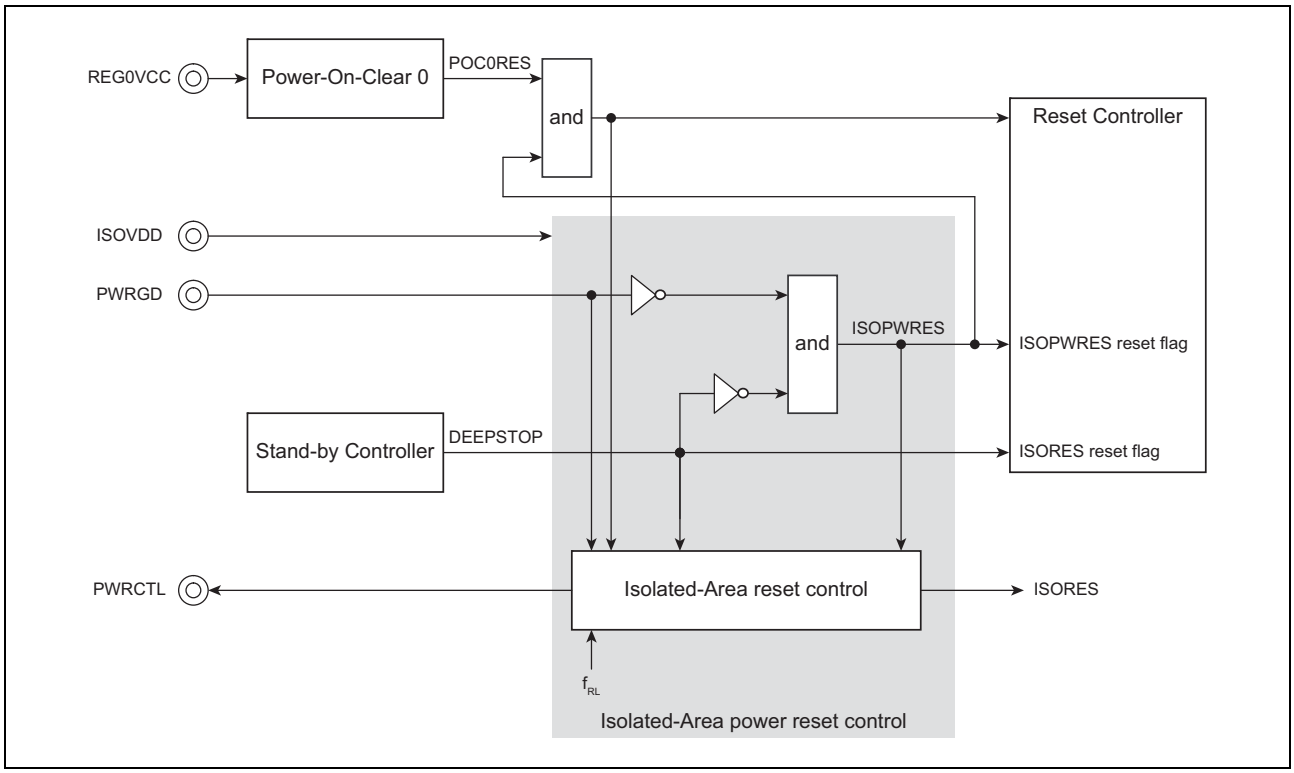


Figure 9.3 D1M2(H) Isolated-Area power reset

(3) Isolated-Area reset control (D1M2(H))

The Isolated-Area reset ISORES is release if

- the external REG1VCC/ISOVDD power supply sets PWRGD = H to confirm stable voltage
- the PWRGD_CNT counter has elapsed

The PWGD counter is used to ensure a minimum delay time from power supply switch on until ISORES release. It counts Low Speed IntOsc clocks f_{RL}.
The start value of the PWGD counter is defined by the PWRGD_CNT register.
The default value of the PWRGD counter after initial power-up is FFF_H, thus the initial ISORES release delay is nominal 4095 x 1/100 kHz = 41 ms.
In order to shorten the wake-up time from DEEPSTOP mode PWRGD_CNT can be set to the lowest acceptable value.

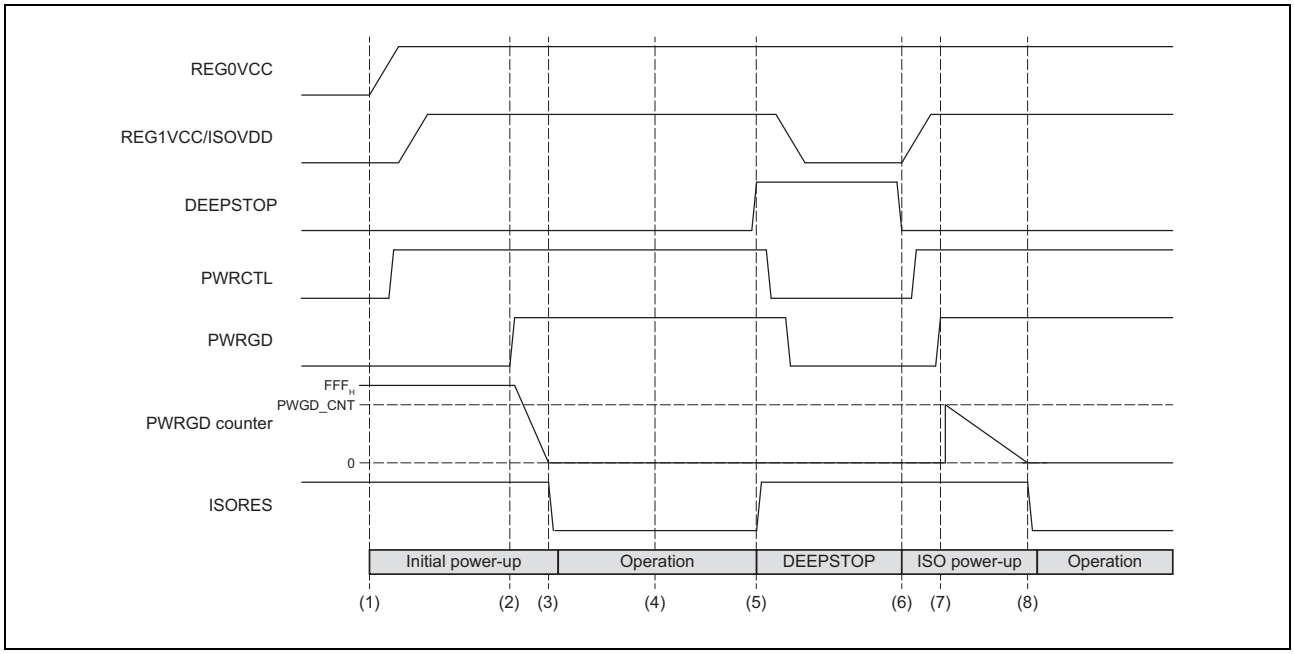


Figure 9.4 Isolated-Area reset control (D1M2(H))

(1) Initial power-up REG0VCC for Always-On-Area starts.
PWRCTL = H signals to external REG1VCC/ISOVDD supply to switch on.

(2) Stable REG1VCC/ISOVDD is signalled by PWRGD = H.
PWRGD counter starts count down from default counter value FFF_H.

NOTE

If PWRGD pin transits H → L → H during PWRGD counter is counting, the counter value is cleared and restarts counting.

(3) PWRGD counter has elapsed. Thus ISORES is released.

(4) New PWRGD_CNT < FFF_H is set up to minimize wake-up time.

(5) Transition to DEEPSTOP mode is entered, thus ISORES is asserted and PWRCTL = L signals to external REG1VCC/ISOVDD supply to switch off.
If REG1VCC/ISOVDD is switched during DEEPSTOP, PWRGD turns to L.

(6) Wake-up from DEEPSTOP
PWRCTL = H signals to external REG1VCC/ISOVDD supply to switch on again.

(7) Stable REG1VCC/ISOVDD is signalled by PWRGD = H.
PWRGD counter starts count down from new counter value < FFF_H.

(8) PWRGD counter has elapsed and ISORES is released.

(4) Isolated-Area reset control (D1L1, D1L2(H), D1M1(H), D1M1-V2, D1M1A)

The Isolated-Area reset ISORES is release after a fixed time*1, if the external REG1VCC power supply is confirmed stable by POC1RES = L and externally set PWRGD = H.

Note 1. – 0.8 ms, when High Speed IntOsc clock f_{RH} is running
– 1.2 ms, when High Speed IntOsc clock f_{RH} is stopping

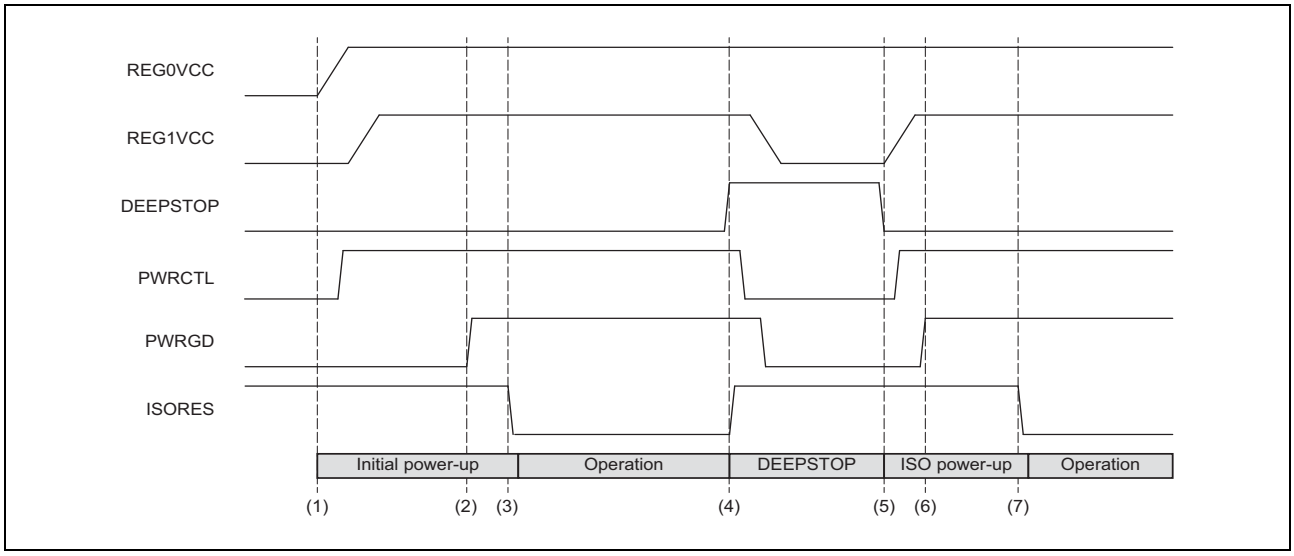


Figure 9.5 Isolated-Area reset control (D1L1, D1L2(H), D1M1(H), D1M1-V2, D1M1A)

(1) Initial power-up REG0VCC for Always-On-Area starts.
PWRCTL = H signals to external REG1VCC supply to switch on.

(2) Stable REG1VCC is signalled by POC1RES = L and PWRGD = H.

(3) ISORES is released after fixed time.

(4) Transition to DEEPSTOP mode is entered, thus ISORES is asserted and PWRCTL = L signals to external REG1VCC supply to switch off.
If REG1VCC is switched during DEEPSTOP, PWRGD turns to L.

(5) Wake-up from DEEPSTOP
PWRCTL = H signals to external REG1VCC supply to switch on again.

(6) Stable REG1VCC is signalled by POC1RES=L and PWRGD = H.
A Low → High signal transition at PWRGD pin is necessary to start the PWRGD counter after PWRCTL = H

(7) ISORES is released after fixed time.

NOTE

In D1M1(H) and D1Lx products, ISOPWRES occurs when REG1VCC is lower than 3.1 V after DEEPSTOP, under following conditions:

- PWRGD is high level before REG1VCC reached 3.1 V.
- Slew rate of REG1VCC is slow and noise is present on REG1VCC

To avoid unintended reset, adapt one of following workarounds.

1. Keep REG1VCC supplied during DEEPSTOP.
2. Rise PWRGD signal after REG1VCC is higher than 3.1 V.
3. Increase slew rate of REG1VCC to meet eliminable noise level.
4. Reduce noise on REG1VCC to meet slew rate.

Eliminable noise level is shown below.

Table 9.3 Eliminable noise level by slew rate

Slew rate for REG1VCC [V/ms]	Eliminable maximum peak to peak of noise on REG1VCC [mV]
≥ 20	220
≥ 10	110
≥ 5	70
≥ 3.3	70
≥ 2.5	50

9.2.2.2 Reset Controller reset output signals

The Reset Controller manages the generation of five reset signals upon occurrence of reset signals from various reset sources:

- Power-up reset PURES
  PURES is generated by the Power-On-Clear 0 circuit POC0RES or the debugger reset $\overline{\text{DBRES}}$.
  It basically initializes all microcontroller modules and ports.
- Power-On-Clear 0 reset POC0RES
  POC0RES is generated by the Power-On-Clear 0 circuit.
- System reset SYSRES
  SYSRES is generated by all reset sources except
  – ISORES, i.e. except during DEEPSTOP mode
  – POC1RES during DEEPSTOP mode
  At the generation of SYSRES, all the clock generation circuits operating up to that time continue to operate.
- ISO area reset ISORES
  ISORES is generated during DEEPSTOP mode and resets all modules in the Isolated-Area.
- Error Control Modules status reset ECMSTATRES
  ECMSTATRES is generated by the power-up PURES and the Clock Monitor 0 reset $\overline{\text{CLMA0RES}}$.
  It clears the error status registers of the Error Control Module.

9.2.2.3 External $\overline{\text{RESET}}$ input/output

Any assertion of a microcontroller reset is signalled via a reset output signal, that can be used to reset external devices concurrently to a microcontroller internal reset.

The internal SYSRES is combined with the external reset input via an open-drain buffer.
Thus any internal SYSRES or external reset sets the microcontroller as well as all other external devices, which are connected to $\overline{\text{RESET}}$, into reset state.

- POC RESET: output low
- Internal reset cause: output low
- External reset inputs: disable output enable

NOTE

In the self programming mode, reset output function can not be used.

9.2.2.4 Watchdog Timers start-up control WDTATRTYP

The Watchdog Timers can be started automatically or remain stopped after reset release. This is controlled by the WDTATRTYP signal and other Watchdog Timers start-up options.

For further information concerning the Watchdog Timers start-up options refer to **Section 26, Window Watchdog Timer (WDTA)**.

9.2.2.5 Operation mode control

The microcontroller's operation mode is determined upon release of the Power-On-Clear reset POC0RES or the external $\overline{\text{RESET}}$*1.

Note 1. The external reset includes all SYSRES in normal operation mode because RESET output is available.

Refer to Section 6, Operating Modes for details about the operation modes.

9.2.2.6 Module reset control

Following modules can be separately set into reset state by use of the reset control register MRSTC:

- MLBB0RES = 0 sets the Media Local Bus MLBB0 into reset
- LCBI0RES = 0 sets the LCD Bus I/F LCBI0 into reset
- SGnRES = 0 sets the Sound Generator SGn into reset
- ADCE0RES = 0 sets the A/D Converter ADCE0 into reset
- ISM0RES = 0 sets the Stepper Motor Driver ISM0 into reset
- ETNB0RES = 0 sets the Ethernet AVB MAC ETNB0 into reset
- VDCEnRES = 0 sets the Video Data Controller VDCEn into reset
- SDRB0RES = 0 sets the DDR2-SDRAM Memory Controller SDRB0 PHY into reset
  Note: In D1M1H and D1M1A, SDRB0RES bit corresponds to SDRA0RES signal, that reset SDR-SDRAM Memory Controller SDRA0.
- XC0RES = 0 sets XC0 cross-connect into reset

CAUTION

By default, i.e. after release of the system reset SYSRES, the above module resets are active.
Thus before accessing any of these modules its reset must be released by setting the respective reset bit to 1.

9.2.3 Reset Controller redundancy

For functional safety reasons two identical Reset Controllers are provided to ensure that a reset event performs the microcontroller initialization even in case of a Reset Controller failure. Both, the master and the redundant Reset Controller, have identical reset inputs. Their respective outputs are combined by a logical OR. Thus if one controller fails, the other still generates the reset.

Both Reset Controllers have an identical set of registers:

- master Reset Controller registers: reset factor RESF and reset factor clear RESFC
- redundant Reset Controller registers: reset factor RESFR and reset factor clear RESFCR

9.2.4 Masking of reset source in debugging mode

The following table informs, which reset sources can be masked, i.e. deactivated in debug mode. The reset mask is configurable via the debugger.

Table 9.4 Enabled or disabled reset sources in debug mode

Reset Source	Maskable
Power-On-Clear 0 reset (POC0RES)	no
Debugger reset ($\overline{\text{DBRES}}$)	no
Power-On-Clear 1 reset (POC1RES)	no
Isolated-Area reset (ISORES)	no
External reset ($\overline{\text{RESET}}$)	yes
Error Control Module reset (ECMRES)	yes
Clock Monitor 0 reset ($\overline{\text{CLMA0RES}}$)	yes
Watchdog Timer overflow (WDTA0RES, WDTA1RES)	yes
Software reset (SWRES)	yes

9.3 Registers

This section contains a description of all registers of the Reset Controller.

9.3.1 Reset Controller registers overview

The Reset Controller is controlled and operated by the following registers:

Table 9.5 Reset Controller registers overview

Module Name	Register Name	Symbol	Address
General reset flags registers			
SYS	Reset factor register	RESF	FFF8 0760$_H$
SYS	Reset factor clear register	RESFC	FFF8 0768$_H$
SYS	Redundant reset factor register	RESFR	FFF8 0860$_H$
SYS	Redundant reset factor clear register	RESFCR	FFF8 0868$_H$
Software reset control registers			
SYS	Software reset register	SWRESA	FFF8 0A04$_H$
SYS	Module reset control register	MRSTC	FFF8 F400$_H$
Isolated-Area reset control register			
PWRG	PWRGD counter register	PWRGD_CNT	FFF8 F500$_H$
Write protection registers			
SYS	Protection command register	PROTCMD0	FFF8 0000$_H$
SYS	Protection status register	PROTS0	FFF8 0004$_H$
SYS	MRSTC protection command register	PROTCMDMRST	FFF8 F480$_H$
SYS	MRSTC protection status register	PROTSMRST	FFF8 F484$_H$
PWRG	Protection command register	PROTCMDPWRGD	FFF8 F580$_H$
PWRG	Protection status register	PROTSPWRGD	FFF8 F584$_H$

NOTE

As for the protection registers, see Section 4, Write-Protected Registers.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> is defined are the above table.

9.3.2 General reset flag registers details

9.3.2.1 RESF, RESFR — Reset factor register

These registers contain information about which type of resets occurred since the last power-up reset PURES.

RESF is the reset factor register of the Master Reset Controller.
RESFR is the reset factor register of the Redundant Reset Controller.

NOTES

1. The following description refers only to the RESF bits.
   The corresponding bits of the RESFR register are named RESFRx.
2. For details, see Figure 9.7, When RESET is Released before the Flash Sequence is Completed.

In accordance with the setting conditions for each bit in the reset factor register, the same bits are set in this register.

Access: This register can be read in 32-bit units.

Address: RESF: FFF8 0760_H
RESFR: FFF8 0860_H

Initial value: 0000 0200_H/0000 0300_H *1 This register is initialized by a power-up reset PURES.

Note 1. For details, see Figure 9.7, When RESET is Released before the Flash Sequence is Completed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESF15	RESF14	0	0	0	RESF10	RESF9	RESF8	0	0	0	0	RESF3	RESF2	RESF1	RESF0
Initial value	0	0	0	0	0	0	1	1/0*1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. For details, see Figure 9.7, When RESET is Released before the Flash Sequence is Completed.

NOTE

The corresponding bits of the RESFR register are named RESFRx.

Table 9.6 RESF, RESFR register contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, an initial value is returned. The write value should be 0.
15	RESF15	Isolated-Area power reset (ISOPWRES) reset flag 0: No reset occurred 1: Reset has occurred
14	RESF14	Error Control Module (ECMRES) reset flag 0: No reset occurred 1: Reset has occurred
13 to 11	Reserved	When read, an initial value is returned. The write value should be 0.

Table 9.6 RESF, RESFR register contents (2/2)

Bit Position	Bit Name	Function
10	RESF10	Isolated-Area (ISORES) reset flag 0: No reset occurred 1: Reset has occurred
9	RESF9	Power-On-Clear 0 (POC0RES) reset flag 0: No reset occurred 1: Reset has occurred
8	RESF8	External reset ($\overline{\text{RESET}}$) flag 0: No reset occurred 1: Reset has occurred
7 to 4	Reserved	When read, an initial value is returned. The write value should be 0.
3	RESF3	CLMA0 ($\overline{\text{CLMA0RES}}$) reset flag 0: No reset occurred 1: Reset has occurred
2	RESF2	WDTA1 (WDTA1RES) reset flag 0: No reset occurred 1: Reset has occurred
1	RESF1	WDTA0 (WDTA0RES) reset flag 0: No reset occurred 1: Reset has occurred
0	RESF0	Software (SWRES) reset flag 0: No reset occurred 1: Reset has occurred

NOTE

In normal operation mode, reset factors (except for ISORES) are noticed with external $\overline{\text{RESET}}$ flag (RESF8). See Section 9.2.2.3, External RESET input/output for details.

9.3.2.2 RESFC, RESFCR — Reset factor clear register

RESFC is the reset factor clear register of the Master Reset Controller. This register clears the reset flags of the RESF register.

RESFCR is the reset factor clear register of the Redundant Reset Controller. This register clears the reset flags of the RESFR register.

NOTE

The following description refers only to the RESFC bits.
The corresponding bits of the RESFCR register are named RESFCRx.

Access: This register can be written in 32-bit units.

Address: RESFC: FFF8 0768_H
RESFCR: FFF8 0868_H

Initial value: Reading this register always returns 0000 0000_H.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RESFC15	RESFC14	0	0	0	RESFC10	RESFC9	RESFC8	0	0	0	0	RESFC3	RESFC2	RESFC1	RESFC0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	R	R	R	W	W	W	R	R	R	R	W	W	W	W

Table 9.7 RESFC, RESFCR register contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, an initial value is returned. The write value should be 0.
15	RESFC15	Isolated-Area power reset (ISOPWRES) reset flag clear 0: No function 1: Clear reset flag RESF.RESF15
14	RESFC14	Error Control Module (ECMRES) reset flag clear 0: No function 1: Clear reset flag RESF.RESF14
13 to 11	Reserved	When read, an initial value is returned. The write value should be 0.
10	RESFC10	Isolated-Area (ISORES) reset flag clear 0: No function 1: Clear reset flag RESF.RESF10
9	RESFC9	Power-On-Clear 0 (POC0RES) reset flag clear 0: No function 1: Clear reset flag RESF.RESF9
8	RESFC8	External reset ($\overline{\text{RESET}}$) flag clear 0: No function 1: Clear reset flag RESF.RESF8
7 to 4	Reserved	When read, an initial value is returned. The write value should be 0.
3	RESFC3	CLMA0 ($\overline{\text{CLMA0RES}}$) reset flag clear 0: No function 1: Clear reset flag RESF.RESF3

Table 9.7 RESFC, RESFCR register contents (2/2)

Bit Position	Bit Name	Function
2	RESFC2	WDTA1 (WDTA1RES) reset flag clear 0: No function 1: Clear reset flag RESF.RESF2
1	RESFC1	WDTA0 (WDTA0RES) reset flag clear 0: No function 1: Clear reset flag RESF.RESF1
0	RESFC0	Software (SWRES) reset flag clear 0: No function 1: Clear reset flag RESF.RESF0

9.3.3 Software reset control registers details

9.3.3.1 SWRESA — Software reset register

This register is used to generate a software reset SWRES.

Access: This register can be written in 32-bit units.
Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0. For details, refer to Section 4, Write-Protected Registers.

Address: FFF8 0A04$_H$

Initial value: Reading this register always returns 0000 0000$_H$.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SWRESA
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 9.8 SWRESA register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, an initial value is returned. The write value should be 0.
0	SWRESA	Software reset trigger 0: no function 1: Generate software reset SWRES

9.3.3.2 MRSTC — Module reset control register

This register is used to generate separate resets for various modules.

CAUTION

By default, i.e. after release of the power-up reset PURES, the system reset SYSRES and the Isolated-Area reset ISORES, the module resets are active.

Thus before accessing any of these modules its reset must be released by setting the respective reset bit to 1.

NOTE

Reset bits of unused modules should remain in reset state with exception of XC0RES and ETNB0RES.
Refer to Section 12.4, Clock Controller set-up for details.

Access: This register can be read/written in 32-bit units.
Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDMRST. For details, refer to Section 4, Write-Protected Registers.

Address: FFF8 F400$_H$

Initial value: 0000 0000$_H$. This register is initialized by a system reset SYSRES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	XC0RES	SDRB0RES	MLBB0RES	LCBI0RES	SG4RES	SG3RES	SG2RES	SG1RES	SG0RES	ADCE0RES	ISM0RES	ETNB0RES	VDCE1RES	VDCE0RES
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 9.9 MRSTC register contents (1/2)

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, an initial value is returned. The write value should be 0.
13	XC0RES	The XC0RES only need to be used when only setup C_ISO_XCCLK. Please write only "1" to XC0RES after setup C_ISO_XCCLK. 0: reset is active 1: reset is inactive
12	SDRB0RES	Software reset of DDR2-SDRAM Memory Controller SDRB0 PHY*1 (D1M2(H), D1M1H, D1M1Aonly) 0: reset is active 1: reset is inactive Note 1. In D1M1H and D1M1A, SDRB0RES bit corresponds to SDRA0RES signal, that reset SDR-SDRAM Memory Controller SDRA0.
11	MLBB0RES	Software reset of Media Local Bus MLBB0 (D1M2H only) 0: reset is active 1: reset is inactive

Table 9.9 MRSTC register contents (2/2)

Bit Position	Bit Name	Function
10	LCBI0RES	Software reset of LCD Bus I/F LCBI0 0: reset is active 1: reset is inactive
9	SG4RES	Software reset of Sound Generator SG4 0: reset is active 1: reset is inactive
8	SG3RES	Software reset of Sound Generator SG3 and Serial Sound Interface SSIF1 0: reset is active 1: reset is inactive
7	SG2RES	Software reset of Sound Generator SG2 and Serial Sound Interface SSIF0 0: reset is active 1: reset is inactive
6	SG1RES	Software reset of Sound Generator SG1 0: reset is active 1: reset is inactive
5	SG0RES	Software reset of Sound Generator SG0 0: reset is active 1: reset is inactive
4	ADCE0RES	Software reset of A/D Converter ADCE0 0: reset is active 1: reset is inactive
3	ISM0RES	Software reset of Intelligent Stepper Motor Driver ISM0 0: reset is active 1: reset is inactive
2	ETNB0RES	Software reset of Ethernet AVB MAC ETNB0 0: reset is active 1: reset is inactive
1	VDCE1RES	Software reset of Video Data Controller VDCE1 (D1M2(H), D1M1A only) 0: reset is active 1: reset is inactive
0	VDCE0RES	Software reset of Video Data Controller VDCE0 (D1L2(H), D1M1(H), D1M1-V2, D1M2(H), D1M1Aonly) 0: reset is active 1: reset is inactive

9.3.4 Isolated-Area reset control registers details

9.3.4.1 PWRGD_CNT — PWRGD counter register

This register is used to set PWRGD counter value.
It determines a minimum waiting period for PLLVDD and REG1VCC available after PWRGD is high.

Access: This register can be read/written in 32-bit units.
Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDPWRGD. For details, refer to Section 4, Write-Protected Registers.

Address: FFF8 F500$_H$

Initial value: 0000 0FFF$_H$. This register is initialized by POC0RES only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	PWRGD_CNT[11:0]											
Initial value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 9.10 PWRGD_CNT register contents

Bit Position	Bit Name	Function
31 to 12	Reserved	When read, an initial value is returned. The write value should be 0.
11 to 0	PWRGD_ CNT[11:0]	PWR_GD counter value PWR_GD delay time = (PWRGD_CNT[11:0] + 1) x T_{RL} T_{RL} denotes the cycle duration of the Low Speed IntOsc frequency f_{RL}.

NOTE

This register can be written in D1M2(H).
For D1L1, D1L2(H), D1M1(H), D1M1-V2, D1M1A, this register can not be written. (Fixed to initial value.)

9.4 Functional Description

All descriptions refer only to the master Reset Controller.
However they are valid also for the redundant Reset Controller, since both Reset Controllers behave identical.

9.4.1 Reset flags

The reset factor register (RESF) provides reset flags for each reset source.

If a reset has occurred, the assigned flag is set. This way the source of the reset can be evaluated.

RESF is initialized by a power-up reset PURES (POC0RES or $\overline{\text{DBRES}}$).

In addition, the flags in RESF can be cleared by the reset factor clear register (RESFC).

RESF8 and RESF9 is set when PURES is generated.

9.4.2 Power-On-Clear Resets (POC0RES, POC1RES)

NOTE

The RH850/D1L/D1M devices provide following Power-On-Clear circuits:

- All devices: Power-On-Clear circuit 0 supervises the external REG0VCC voltage
- D1Lx, D1M1(H), D1M1-V2 and D1M1A: Power-On-Clear circuit 1 supervises the external REG1VCC voltage

The following description refers to both Power-On-Clear circuits, denoted as POC, supervising the REGVCC voltage and generating the POCRES reset.

The Power-On-Clear circuit (POC) permanently compares the power supply voltage REGVCC with the internal reference voltage VPOC. It ensures that the microcontroller only operates as long as the power supply exceeds a well-defined limit.

If REGVCC falls below the internal reference voltage (REGVCC < VPOC),

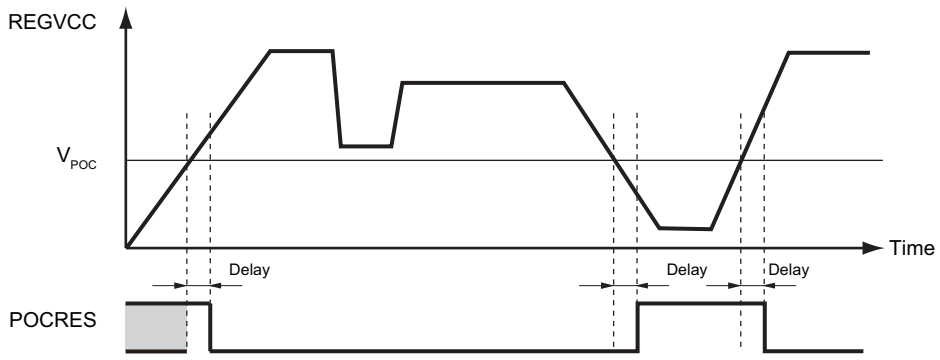
- POC0RES: Isolated-Area reset ISORES, system reset SYSRES, and a power-up reset PURES are generated.
- POC1RES: Isolated-Area reset ISORES and system reset SYSRES are generated.

For details on the specification of the internal voltage reference level VPOC, refer to the Data Sheet.

The reset factor register (RESF) is cleared by the power-on clear reset PURES.
RESF8 and RESF9 are set to 1 after the initialization.

The Power-On-Clear function holds the microcontroller in reset state as long as the power supply voltage does not exceed the threshold level VPOC.

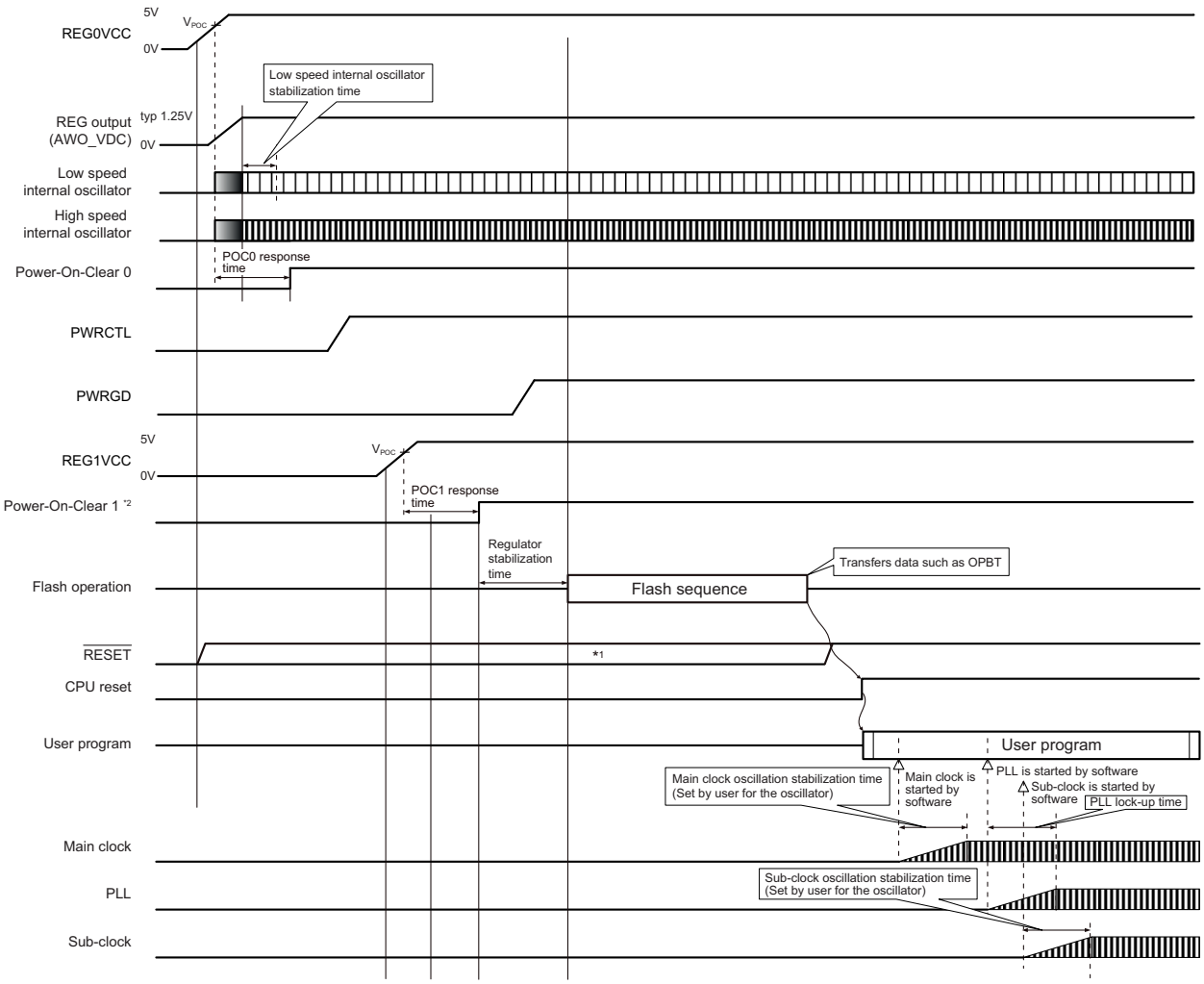
The following figure illustrates the timing of a POCRES.



Note: Delay:
When the REGVCC exceeds the VPOC, the POCRES is released after certain delay. When the REGVCC falls below the VPOC, the POCRES is asserted after certain delay.
See the Data Sheet for the delay time.

Figure 9.6 POC reset timing

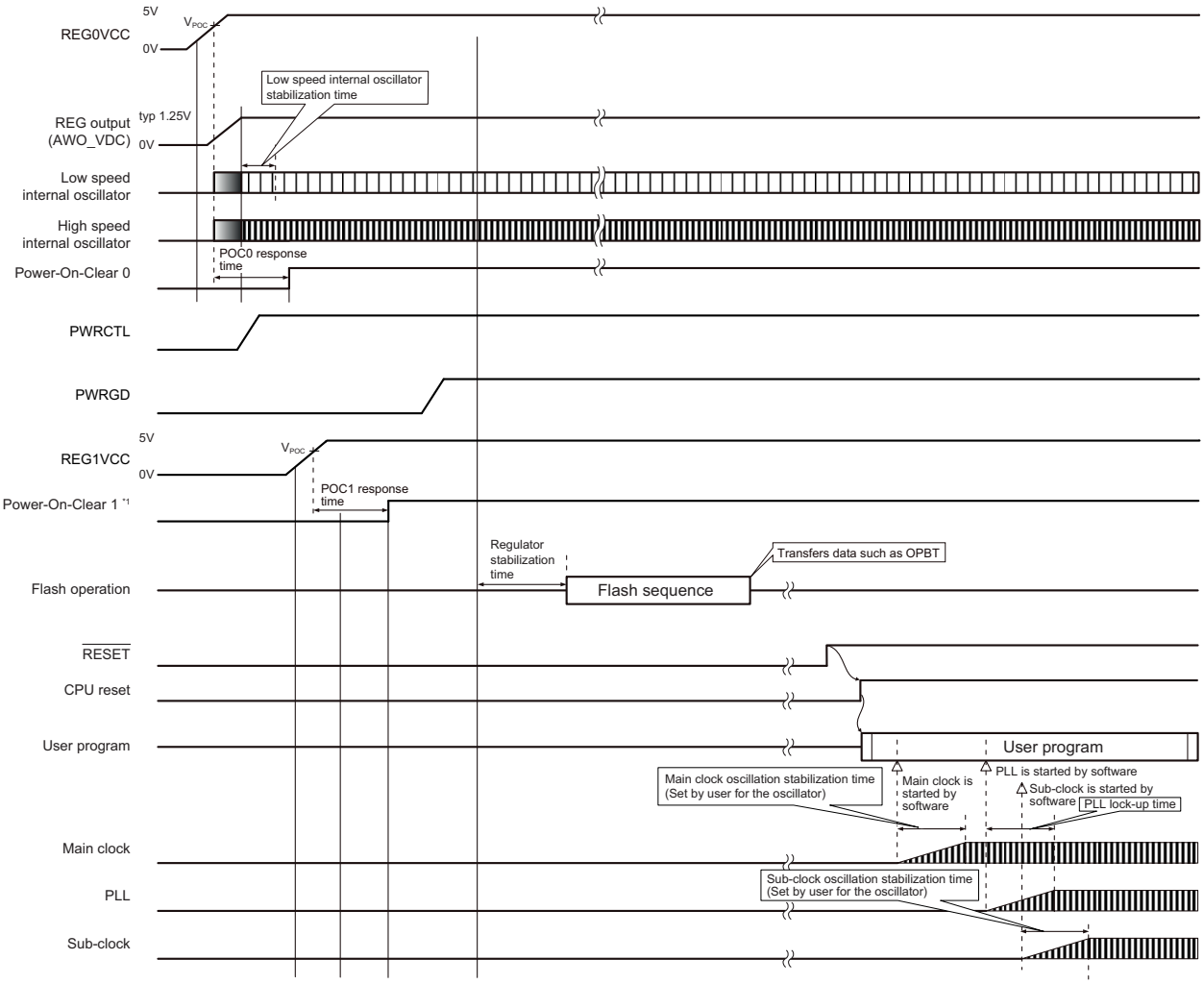
(1) Overview of CPU System Startup after Power-On-Clear



Note 1. After a power-on clear reset, the on-chip pull-down resistors are enabled until the flash sequence is completed. The timing of release from the reset state due to the $\overline{\text{RESET}}$ signal depends on the external circuit of the microcomputer.
If the $\overline{\text{RESET}}$ pin is at the low level, the values of both RESF8.RESF8 and RESFR.RESFR8 are set to 1. If the $\overline{\text{RESET}}$ pin is high level, the initial values of both RESF8.RESF8 and RESFR.RESFR8 are set to 0.

Note 2. In D1M2(H) Power-On-Clear 1 is not available.

Figure 9.7 When $\overline{\text{RESET}}$ is Released before the Flash Sequence is Completed



Note 1. In D1M2(H) Power-On-Clear 1 is not available.

Figure 9.8 When $\overline{\text{RESET}}$ is Released after the Flash Sequence is Completed

9.4.3 External Reset Signal ($\overline{\text{RESET}}$)

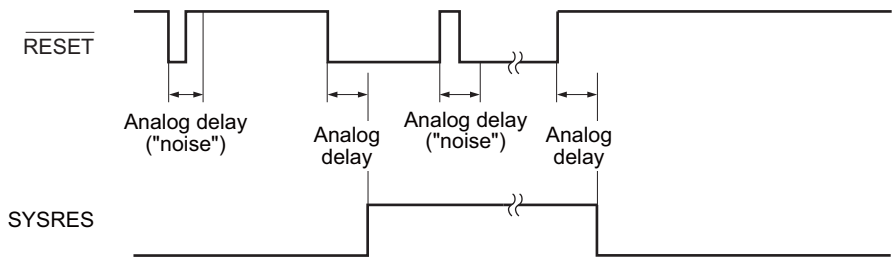
A reset is performed when a low level signal is applied to the $\overline{\text{RESET}}$ pin and the bit RESF.RESF8 is set.

After that, the bit RESF.RESF8 and RESFR.RESFR8 are not cleared automatically, even if the low-level input to the $\overline{\text{RESET}}$ pin is released. The bit RESF.RESF8 is cleared by

- setting the RESFC.RESFC8 bit to 1.
- power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

The $\overline{\text{RESET}}$ signal is passed through an analog noise filter to prevent erroneous resets due to noise.

The following figure shows the timing when SYSRES is generated by an external reset. This figure also shows the effect of the noise filter.



Note: The analog delay is caused by the analog filter. The filter regards pulses up to a certain width as noise and suppresses them.
For the minimum $\overline{\text{RESET}}$ pulse width refer to the Data Sheet.

Figure 9.9 External reset signal ($\overline{\text{RESET}}$)

9.4.4 Watchdog Timer Resets (WDTAnRES)

The Watchdog Timers can be configured to generate a reset if the watchdog time expires. After watchdog reset, the corresponding watchdog timer reset flag (the RESF.RESF1 bit for WDTA0RES, and the RESF.RESF2 bit for WDTA1RES, respectively) are set.

After that, the bit RESF.RESF1 (or the bit RESF.RESF2) is not cleared automatically, if WDTA0RES (or WDTA1RES) is stopped.

The bit RESF.RESF1 and the bit RESF.RESF2 are cleared by

- WDTA0RES:
  Setting the RESFC.RESFC1 bit to 1 clears the RESF.RESF1 bit.
- WDTA1RES:
  Setting the RESFC.RESFC2 bit to 1 clears the RESF.RESF2 bit.
- power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9.4.5 Error Control Module Reset (ECMRES)

The Error Control Module can be configured to generate the reset ECMRES upon various error events. Refer to Section 48, Error Control Module (ECM) for details about the possible error events.

After ECMRES reset, the corresponding Error Control Module reset flag (RESF.RESF14 bit) is set.

After that, the bit RESF.RESF14 is not cleared automatically, if ECMRES is stopped.

The bit RESF.RESF14 is cleared by

- setting the RESFC.RESFC14 bit to 1
- power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9.4.6 Software reset (SWRES)

The software reset SWRES can be asserted by setting

SWRESA.SWRESA = 1.

SWRES sets the reset flag RESF.RESF0 bit.

RESF.RESF0 is not cleared automatically. RESF.RESF0 is cleared by

- setting the RESFC.RESFC0 bit to 1
- power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9.4.7 Clock Monitor 0 Reset ($\overline{\text{CLMA0RES}}$)

The Clock Monitor 0 generates the reset $\overline{\text{CLMA0RES}}$, when a fail of the High Speed IntOsc is detected.

In addition, flag CLMA0RES is set.

This flag is not cleared automatically. It is cleared by

- setting the RESFC.RESFC3 bit to 1
- power-up reset PURES (POCRES or $\overline{\text{DBRES}}$)

9.4.8 Debugger reset ($\overline{\text{DBRES}}$)

Debugger reset ($\overline{\text{DBRES}}$) is generated via a debugger command. $\overline{\text{DBRES}}$ activates PURES, thus operates in the similar way like the Power-On-Clear reset POC0RES:

- It resets the MainOsc and SubOsc clock generators. Hence these clock generators stop operation and must be restarted.
- The reset factor register RESF is cleared (bit RESF9 is set to 1 after initialization.).
- It does not reset the Low Speed IntOsc and High Speed IntOsc clock generators.

Section 10 Power Supply and Power Domains

This section describes the power supply and power domains of the RH850/D1L/D1M.

10.1 Function

The device is separated into two independent power domains for internal circuits, the Always-On-Area (AWO) and the Isolated-Area (ISO).
The power supply of the Always-On-Area remains powered in all operating modes.
The power supply of the Isolated-Area can be turned off in DEEPSTOP mode, reducing the overall device power consumption.
The digital logic of the power domains is supplied by a nominal voltage of 1.25 V.
This voltage is generated by dedicated on-chip voltage regulators for the Always-On-Area and the Isolated-Area of the D1L1, D1L2(H), D1M1(H) and D1M1A devices.
The voltage for the Isolated-Area's digital logic of the D1M2(H) devices is supplied from external.

10.1.1 Power Supply Pins

For operation of the devices the following voltages need to be provided:

Table 10.1 Power Supply Pins Overview (1/2)

Power supply pins	D1L1	D1L2(H)	D1M1(H), D1M1-V2, D1M1A	D1M2(H)
Always-On-Area:				
REG0VCC REG0VSS	AWO digital circuits via on-chip voltage regulator; nominal 3.3 V and 5 V			
OSCVCC OSCVSS	MainOsc and SubOsc; nominal 3.3 V and 5 V			
EVCC EVSS	$\overline{\text{RESET}}$, FLMD0, PWRGD, PWRCTL, port buffers P0 and JP0; nominal 3.3 V and 5 V			
Isolated-Area:				
REG1VCC REG1VSS	ISO digital circuits via on-chip voltage regulator, flash memory and PLL circuits; nominal 3.3 V and 5 V		ISO digital circuits via on-chip voltage regulator, flash memory and PLL circuits; nominal 3.3 V	Flash memory, nominal 3.3 V and 5 V
ISOVDD ISOVSS	–	–	–	ISO digital circuits, nominal 1.25 V
PLLVCC PLLVSS	–	–	–	PLL circuits; nominal 3.3 V and 5 V
B0VCC B0VSS	Port buffers port group P1; nominal 3.3 V and 5 V Port buffers ports P42_7 to P42_10; nominal 3.3 V and 5 V	Port buffers port group P1; nominal 3.3 V and 5 V • D1L2: Port buffers port group P42_7 to P42_10; nominal 3.3 V and 5 V	Port buffers port group P1; nominal 3.3 V and 5 V	Port buffers port groups P1, P2; nominal 3.3 V and 5 V
B1VCC B1VSS	Port buffers port group P3; nominal 3.3 V and 5 V			
B2VCC B2VSS	–	–	–	Port buffers ports P43_0 to P43_6, port groups P46, P47; nominal 3.3 V and 5 V
B3VCC B3VSS	–	–	–	Port buffers ports P43_7 to P43_12; nominal 3.3 V and 5 V

Table 10.1 Power Supply Pins Overview (2/2)

Power supply pins	D1L1	D1L2(H)	D1M1(H), D1M1-V2, D1M1A	D1M2(H)
B4VCC B4VSS	–	• D1L2: – • D1L2H: Port buffers port group P42; nominal 3.3 V and 5 V	Port buffers port group P42; nominal 3.3 V and 5 V	• D1M2: – • D1M2H port buffers port group P42; nominal 3.3 V and 5 V
B5VCC B5VSS	Port buffers port groups P44, P45; nominal 3.3 V and 5 V Port buffers ports P43_0, P43_1; nominal 3.3 V and 5 V • D1M1A : Port buffers port group P45 (OpenLDI); nominal 3.3V			–
RVCC RVSS	–	–	–	Port buffers port groups P44, P45 (VO0 RSDS); nominal 3.3 V
MVCC MVSS	–	–	–	• D1M2: – • D1M2H: port buffers port group P40 (VI0 MIPI I/F); nominal 3.3 V
SFVCC SFVSS	Port buffers port group P21 (Serial Flash I/F); nominal 3.3 V and 5 V	Port buffers port group P21 (Serial Flash I/F); nominal 3.3 V • D1M1-V2, D1M1A: Port buffers port group P22 (Serial Flash I/F); nominal 3.3 V		Port buffers port group • D1M2(H): P21 (Serial Flash I/F) • D1M2H: P21 (Media Local Bus) nominal 3.3 V
ISMVCC ISMVSS	Port buffers port groups P16 and P17 (Stepper Motor Controller/Driver); • nominal 5 V when used for stepper motor operation • nominal 3.3 V and 5 V when not used for stepper motor operation			
ZPDVCC ZPDVSS	Zero point detection circuit; nominal 5 V			
ZPDVREF	Zero point detection external reference voltage; maximum ZPDVCC			
SDRAVCC SDRAVSS	–	–	• D1M1, D1M1-V2: – • D1M1H, D1M1A: SDR-SDRAM I/F port buffers; nominal 3.3 V	–
SDRBVCC SDRBVSS SDRBCKVCC SDRBCKVSS	–	–	–	DDR2-SDRAM I/F; nominal 1.8 V
A0VCC A0VSS	Ports buffers port groups P10 and P11 (A/D Converter analog circuits and input buffers); nominal 3.3 V and 5 V			
A0VREF	A/D Converter reference voltage; maximum A0VCC			

NOTE

For the exact voltage ranges, refer to the data sheet for the Electrical Characteristics.

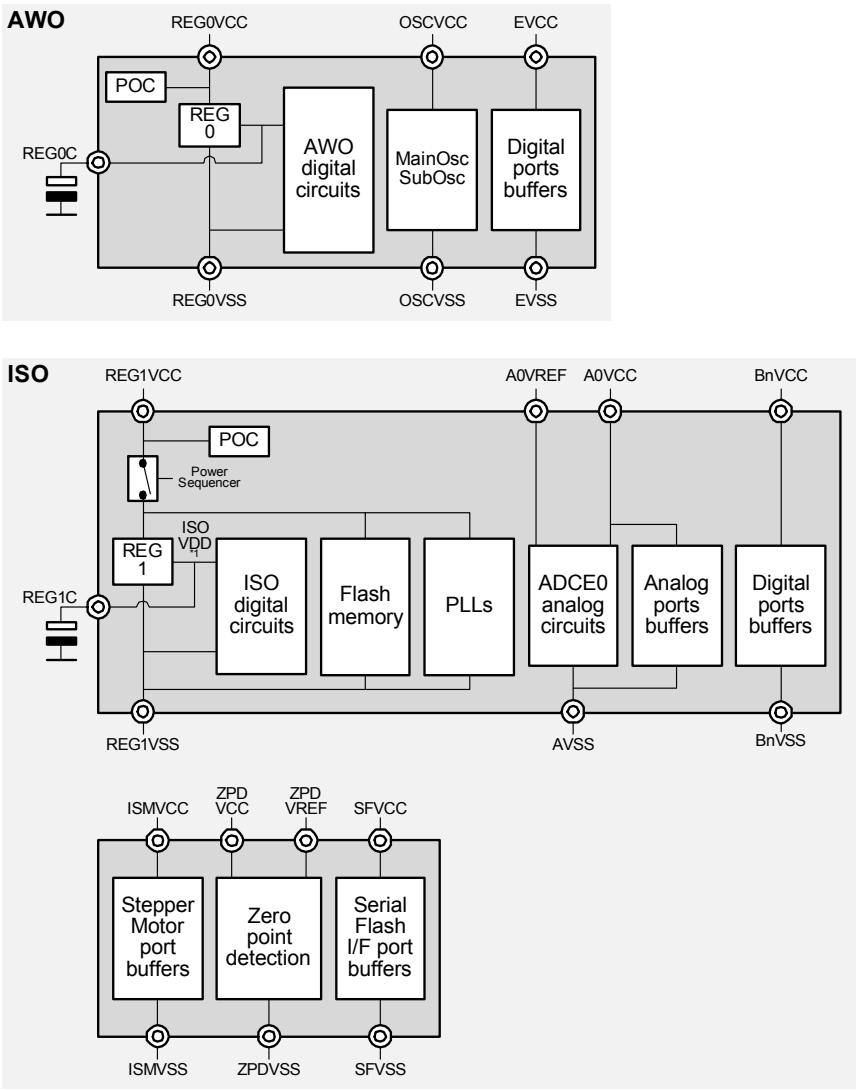
10.1.2 Power-On-Clear

The external power supplies

- REG0VCC
- REG1VCC of the D1L1, D1L2(H), D1M1(H), D1M1-V2 and D1M1A devices

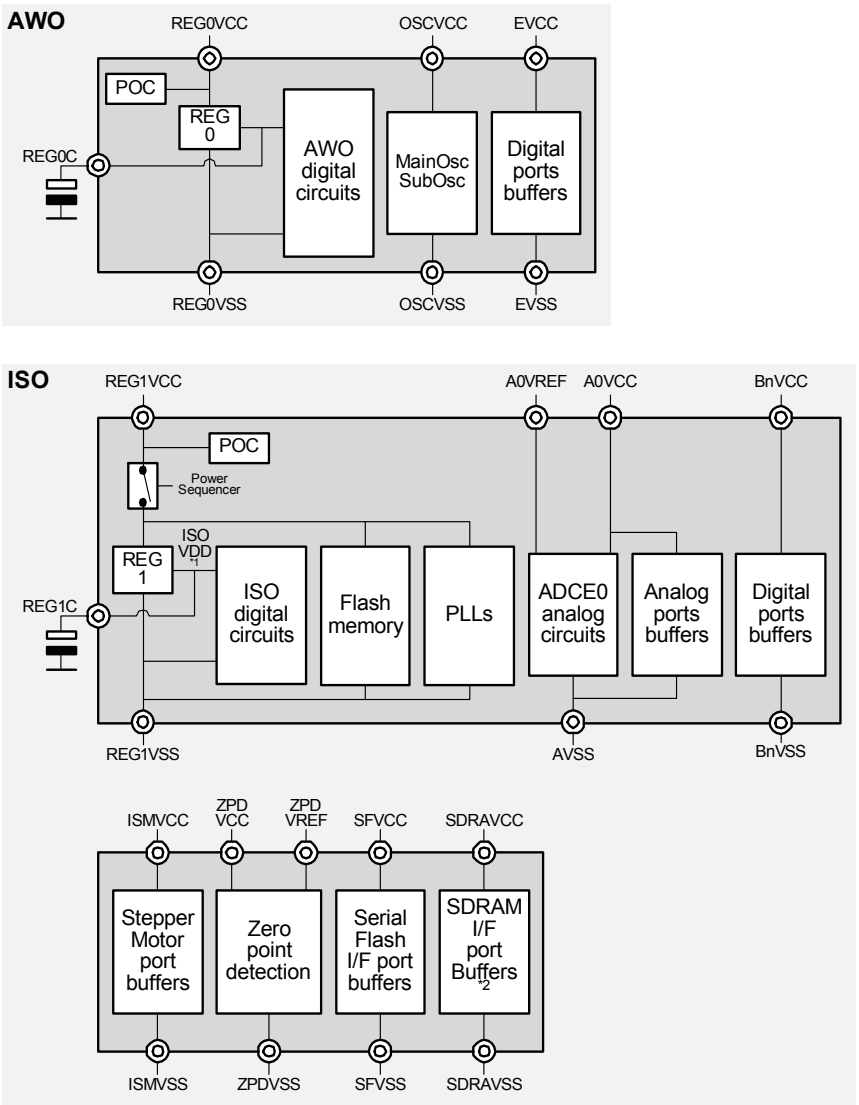
are monitored internally by Power-On-Clear circuits in order to secure the device operation. Refer to Section 9, Reset Controller for details.

10.1.3 Block Diagram of Power Domains



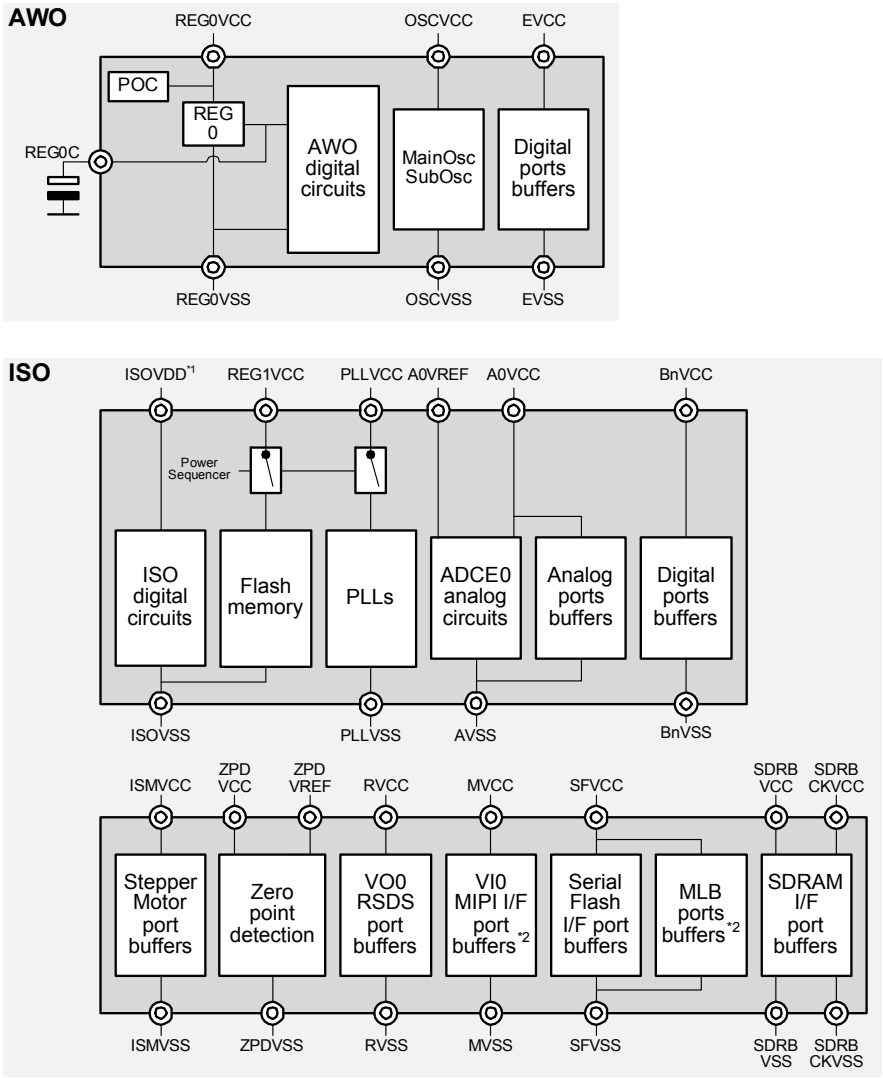
Note 1. ISOVDD is also input to the A/D Converter channel 20 for supply voltage monitoring purposes.

Figure 10.1 D1L1, D1L2(H) power supply overview



Note 1. ISOVDD is also input to the A/D Converter channel 20 for supply voltage monitoring purposes.
Note 2. Only for D1M1H and D1M1A

Figure 10.2 D1M1(H), D1M1-V2 and D1M1A power supply overview



Note 1. ISOVDD is also input to the A/D Converter channel 20 for supply voltage monitoring purposes.
Note 2. Only for D1M2H.

Figure 10.3 D1M2(H) power supply overview

Section 11 Temperature Measurement

The temperature measurement facility measures the on-chip temperature by use of an on-chip Temperature Sensor (TSN). The output signal TSNOUT of the sensor is converted by the A/D Converter into a temperature representative value, which is checked to be in a configurable value range. In case it crosses that range the error interrupt INTADCE0ERR is generated.
The temperature out-of-range detection interrupt INTADCE0ERR is forwarded to the Error Control Module and thus can initiate interrupts and a reset.

A selectable Timer Array Unit B (TAUB) channel is used to trigger the A/D conversion - i.e. control of the temperature - periodically.

The temperature measurement is conducted by the CPU, which

- enables the Temperature Sensor
- sets up the selected TAUB channel as interval timer with the desired cycle duration
- starts the TAUB interval timer, when the start-up time of the sensor has passed and the Temperature Sensor output signal TSNOUT is stable.

Afterwards continuous temperature control is running without any CPU involvement.

In order to ensure sufficient start-up time to stabilize the sensor output TSNOUT a programmable start-up counter is provided , that is operated with the C_ISO_PCLK clock. The start-up time is minimum 200 µs.
The counter is automatically started with enabling the Temperature Sensor and indicates a stable TSNOUT signal by setting the ready flag TSNSTAT.TSNRDY = 1.

The A/D Converter provides a dedicated conversion channel, assigned to a dedicated scan group, for the temperature signal TSNOUT conversion, that allows to chose long sampling times for the Sample & Hold circuit of the A/D Converter in order to cope with voltage drops due to the capacitive load during sampling.

Temperature Measurement functions summary

- Temperature sensor is available with the analog channel 37of A/D converter unit 0 (ADCE0).
- Tj = Between -40°C to 25°C: ±10°C
  Tj = Between 25°C to 150°C: ±5°C
- Minimum Temperature measurement time: 6.55 µs (with 40 MHz CLKAD)
- start-up time: 200 µs
- Temperature alarm error is generated when temperature reaches to the upper/lower limit.
- Temperature measurement mode
  Continuous temperature measurement mode
- Self-diagnosis function
  Supporting self-diagnosis function by inserting high temperature error.
- Error notification function
  ADCE0 interrupt and TSN error are generated when the result of temperature measurement exceeded the value set in ADCEmULLMTBR0 to ADCEmULLMTBR2 registers.
- Supporting error reduction by calibration between two measurement points of temperature (Tj = 25°C and Tj = 150°C).

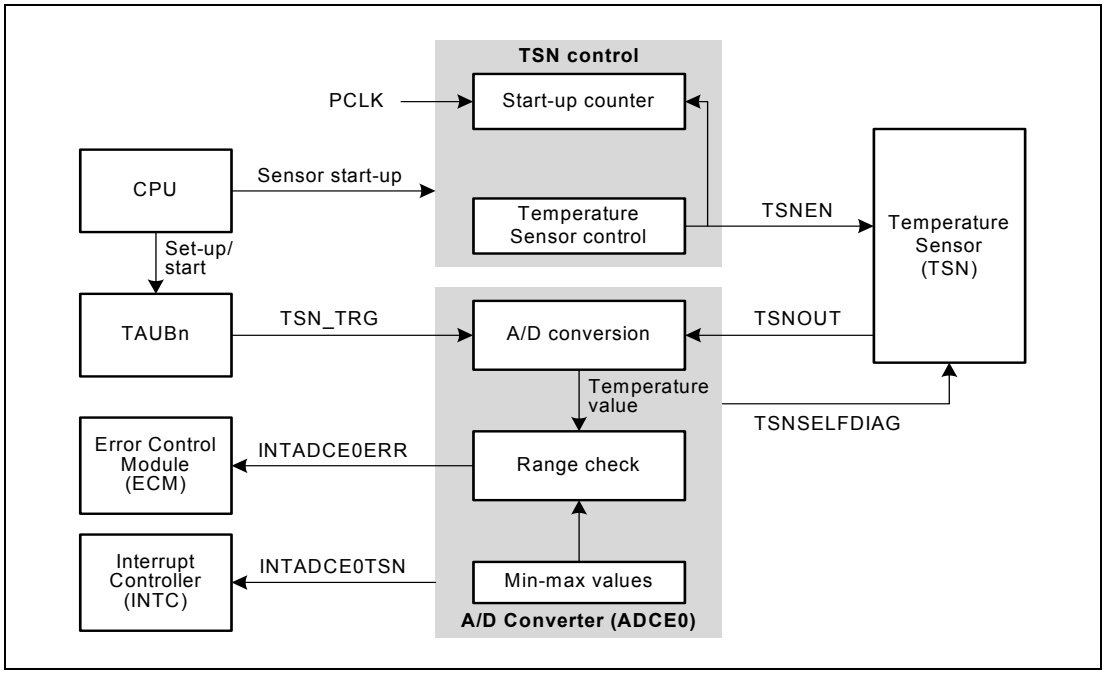


Figure 11.1 Temperature measurement set-up

11.1 Temperature calculation

The current temperature T_c can be calculated with the following formular:

$$T_c = \frac{150 - 25}{R_{150} - R_{25}} \times (R - R_{25}) + 25$$

T_c:	Current temperature in °C
R:	A/D conversion result of current measurement
R_{25}:	A/D conversion reference value for 25 °C for A0VREF = 5V
R_{150}:	A/D conversion reference value for 150 °C for A0VREF = 5V

The reference values read from the flash mask option bytes:

- OPBT10.TSNREFDH[11:0]: R_{150} value for A0VREF = 5V
- OPBT10.TSNREFDL[11:0]: R_{25} value for A0VREF = 5V

11.2 Temperature measurement initialization

The flow of AD conversion for temperature sensor is as follows.

(1) Select and set up the TAUBn channel to be used for triggering the temperature measurement A/D conversion. The TAUBn channel must not be started at this stage, but after the Temperature Sensor output signal is stable, i.e. after **(5)**.

(2) Set up all A/D Converter registers, related to temperature measurement:

- Temperature Sensor output limits by an upper/lower limit table register ADCE0ULLMTBR0 to 2

- Temperature measurement virtual channel register ADCE0TSNVCR:
  - upper/lower limit selection by TSNULS[1:0]
  - physical channel selection by TSNGCTRL[5:0] = 37
- Temperature measurement sampling control register ADCE0TSNSMPCR:
  - sampling time TSNSMPT[7:0]. The sampling time must be set to 6 µs. The correct TSNSMPT[7:0] value depends on the A/D Converter clock CLKAD.
- Temperature measurement trigger select control register ADCE0TSNSGTSEL
  - TTSNSEL[15:0] selects the A/D conversion trigger, i.e. the TAUBn channel, selected for triggering the A/D conversion in **(1)**.
- Temperature measurement scan group control register ADCE0TSNSGCR
  - TSNTRGMD = 1 enables the trigger to perform A/D conversion of the Temperature Sensor output signal

(3) Temperature Sensor start-up time by setting the TSNCNTCMP counter register.
The counter counts C_ISO_PCLK clocks. TSNCNTCMP has to be set to pass the minimum start-up time of 200 µs.

(4) Enable the Temperature Sensor by setting TSNCR.TSNEN = 1.
This starts also the Temperature Sensor start-up counter.

(5) Wait until the Temperature Sensor output TSNOUT is stable.
Stable output is indicated by TSNSTAT.TSNRDY = 1.

(6) Start the TAUBn channel, selected as the trigger in **(1)**.

(7) The TAUBn channel generates the conversion triggers TSN_TRG periodically and initiates A/D conversion of the Temperature Sensor output signal.

(8) The A/D Converter asserts the INTADCE0TSN interrupt, each time the conversion is completed.

(9) In case the A/D conversion result of the Temperature Sensor output exceeds the upper/lower limits, the A/D Converter asserts the error interrupt INTADCE0ERR and sets the upper limit/lower limit error register ADCE0ULER accordingly:
  - UE = 1 indicates Temperature Sensor output above upper limit
  - LE = 1 indicates Temperature Sensor output below lower limit
  - ULSG[1:0] = 0 indicates temperature measurement scan group number
  - ULECAP[5:0] = 37 indicated temperature measurement physical channel.

NOTES

1. All general A/D Converter settings, like e.g. resolution and alignment control via the A/D control register ADCE0ADCR and error control settings via the safety control register ADCEnSFTCR, are also valid for the temperature measurement.
2. For a detailed description of the A/D Converter registers refer to Section 45, A/D Converter (ADCE).

11.2.1 Stopping the temperature measurement

In order to stop the temperature measurement, disable temperature measurement scan group trigger via the control register ADCE0TSNSGCR:

– TSNTRGMD = 0 disables further A/D conversion triggers

11.2.2 Temperature measurement self-diagnostic

A temperature measurement self-diagnostic function allows to check proper operation of the Temperature Sensor.

The self-diagnostic mode is selected via the TSNSELFDIAG control bit of the A/D converter control register ADCEnADCR:

- TSNSELFDIAG = 0: self-diagnostic disabled
- TSNSELFDIAG = 1: self-diagnostic enabled

If self-diagnostic is enabled the Temperature Sensor outputs TSNOUT = 0 V.

During Temperature Sensor self-diagnostic the A/D Converter remains in normal operation mode.

In self-diagnostic mode it is expected that the maximum allowed temperature is exceeded.
Thus the lower limit ADCEnULLMTBRx.LLMTB[11:0], that represents the maximum allowed temperature T_{cmax}, has to be set for a temperature of

$T_{cmax} = T_{jmax} + 10\ °C$

By rearranging the formula for the temperature calculation, as given in Section 11.1, Temperature calculation, the lower limit value ADCEnULLMTBRx.LLMTB[11:0] = Rmin can be calculated by

$$R_{min} = \frac{(R_{150} - R_{25}) \times (T_{cmax} - 25)}{150 - 25} + R_{25}$$

T_{cmax}:	Maximum allowed temperature in °C
R_{min}:	A/D conversion result, representing the maximum allowed temperature T_
R_{25}:	A/D conversion reference value for 25 °C for A0VREF = 5V
R_{150}:	A/D conversion reference value for 150 °C for A0VREF = 5V

NOTE

ADCEnULLMTBRx with x = 0 to 2 in denotes the upper/lower limit pair, selected for the temperature measurement.

11.3 List of Registers

Following registers are used to operate the Temperature Sensor:

Table 11.1 Temperature Sensor registers

Module Name	Register Name	Symbol	Address
SELB	Temperature Sensor control register	TSNCR	FFC0 6034_H
SELB	Temperature Sensor status register	TSNSTAT	FFC0 6038_H
SELB	Compare match timer compare register	TSNCNTCMP	FFC0 $604C_H$

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

11.3.1 TSNCR — Temperature Sensor control register

This register enables or disables the Temperature Sensor.

Access: This register can be read or written in 32-bit units.

Address: FFC0 6034_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSNEN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 11.2 TSNCR register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	TSNEN	Temperature Sensor enable/disable 0: Temperature Sensor disabled 1: Temperature Sensor enabled

11.3.2 TSNSTAT — Temperature Sensor status register

This register indicates the status of the Temperature Sensor.

Access: This register can be read in 32-bit units.

Address: FFC0 6038$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSN RDY
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 11.3 TSNSTAT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	TSNRDY	Temperature Sensor status flag 0: Temperature Sensor is disabled or in start-up 1: Temperature Sensor is ready

11.3.3 TSNCNTCMP — Compare match timer compare register

This register defines the start-up time of the Temperature Sensor.

Access: This register can be read or written in 32-bit units.

Address: FFC0 604C$_H$

Initial value: FFFF FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSNCNTCMP[31:16]															
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSNCNTCMP[15:0]															
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11.4 TSNCNTCMP register contents

Bit Position	Bit Name	Function
31 to 0	TSNCNT CMP[31:0]	Compare value for start-up counter of the Temperature Sensor TSN

Section 12 Clock Controller

This section describes the Clock Controller functions of the RH850/D1L/D1M microcontrollers.

12.1 Clock Controller Overview

Features summary

The Clock Controller has the following functions:

- six clock sources:
  - Low Speed Internal Oscillator (LS IntOsc) with a nominal frequency of 240 kHz
  - High Speed Internal Oscillator (HS IntOsc) with a nominal frequency of 8 MHz
  - Sub Oscillator (SubOsc) with a nominal frequency of 32.768 kHz
  - Main Oscillator (MainOsc) 8 to 16 MHz
  - D1L1, D1L2(H), D1M1(H), D1M2(H):
    Spread Spectrum Phase Locked Loop (PLL0) with a nominal frequency of 320 MHz, 400 MHz, 480 MHz
  - D1M1-V2, D1M1A:
    Spread Spectrum Phase Locked Loop (PLL0) with a nominal frequency of 800 MHz, 960 MHz
  - D1L1, D1L2(H), D1M1(H), D1M1-V2, D1M1A:
    Fixed frequency Phase Locked Loop (PLL1) with a nominal frequency of up to 480 MHz
  - D1M2(H):
    Fixed frequency Phase Locked Loop (PLL1) with a nominal frequency of up to 960 MHz
  - D1M2(H) only: Phase Locked Loop (PLL2) for the video output channels pixel clock generation
    Refer to Section 37.5, Video channels clock generation for details.
- five Clock Monitors (CLMA0 to CLMA4) for
  - CLMA0 monitors the High Speed Internal Oscillator (HS IntOsc) f_{RH}
  - CLMA1 monitors the Low Speed Internal Oscillator (LS IntOsc) f_{RL}
  - CLMA2 monitors the Main Oscillator (MainOsc) f_X
  - CLMA3 monitors the PLL0 clock PLL0CLK
  - CLMA4 monitors the PLL1 clock PLL1CLK

  The additional two clock monitors CLMA5 and CLMA6 monitor the Video Input Interfaces pixel clocks at the ports VDCE0_VI_CLK and VDCE1_VI_CLK.
  Refer to Section 37.5.8, Video Input Pixel Clock Monitoring for details.

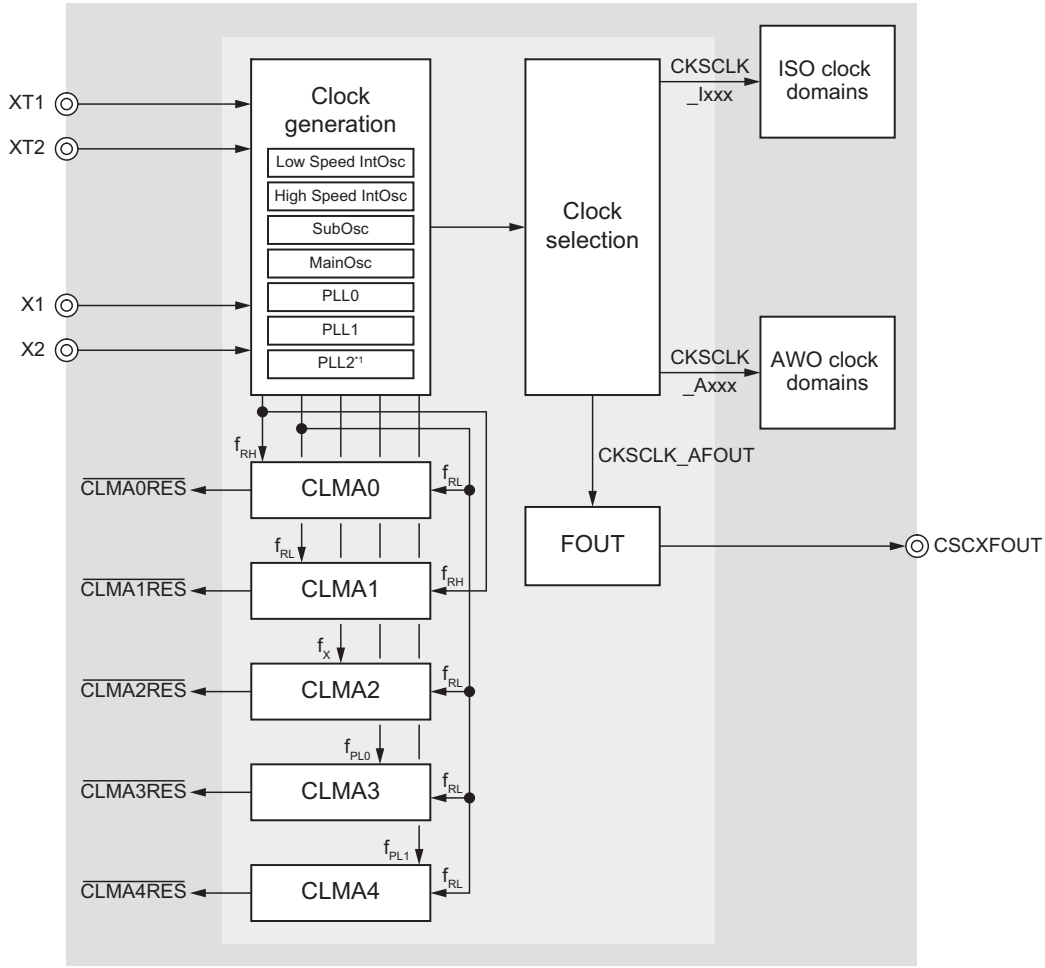
  If a Clock Monitor detects a failure of the monitored clock, it generates a signal towards the reset Controller or the Error Control Module (ECM).
- selectable and adjustable frequency output (CSCXFOUT)

For the specifications of the frequencies, acceptable variation, and other parameters of the clock generators, refer to the Data Sheet.

PLL index k

Throughout this section, the individual units of the PLL circuits are identified by the index "k" (k = 0 to 2). PLL2 is only available for D1M2(H) devices.

The following figure shows the main components of the Clock Controller.



Note 1. PLL2 is only available for D1M2(H) devices.

Figure 12.1 Clock Controller overview

12.1.1 Naming conventions

The clock signals, their control registers, etc. follow a defined naming convention, that reflects their membership to a certain power domain and clock domain.

Clock domain placeholder “xxx”

The placeholder “xxx” is used to identify the clock domain name, according to following naming convention:

- Clock domain names:
  - C_AWO_xxx: Always-On-Area clock domain
  - C_ISO_xxx: Isolated-Area clock domain
- Domain clock names:
  - CKSCLK_Axxx: Always-On-Area domain clock
  - CKSCLK_Ixxx: Isolated-Area domain clock
- Clock selector names:
  - CKSC_Axxx: clock selector for Always-On-Area clock selectors
  - CKSC_Ixxx: clock selector for Isolated-Area clock selectors
- Clock selector registers:
  - CKSC_AxxxS_CTL: Always-On-Area source clock selector registers
  - CKSC_AxxxD_CTL, CKDV_AxxxD_CTL: Always-On-Area source clock divider registers
  - CKSC_IxxxS_CTL: Isolated-Area source clock selector registers
  - CKSC_IxxxD_CTL, CKDV_IxxxD_CTL: Isolated-Area source clock divider registers

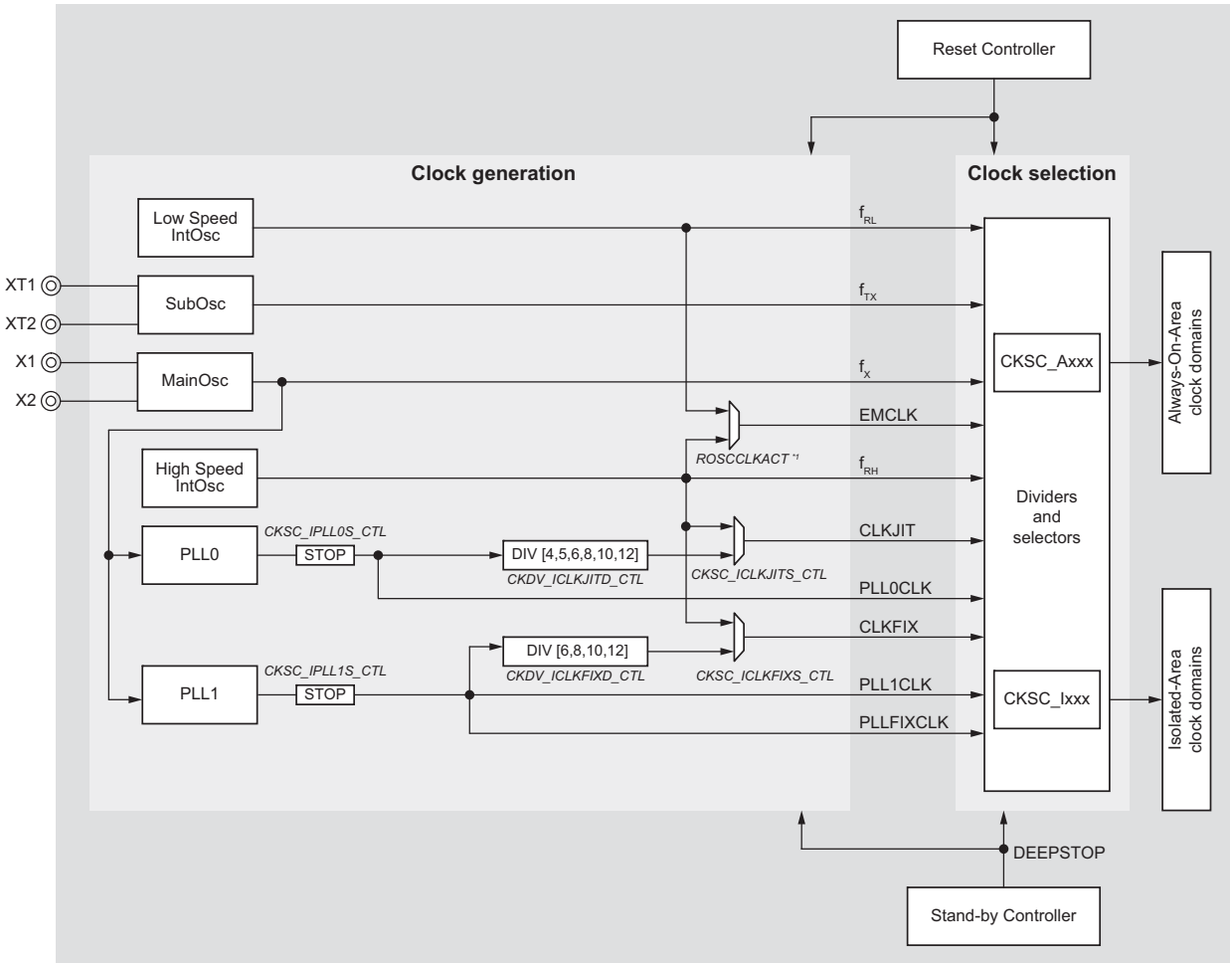
Example

The clock signal CKSCLK_ARTCA (placeholder xxx = RTCA) is the clock supplied to the clock domain C_AWO_RTCA in the Always-On-Area. This clock is selected by the clock selector register CKSC_ARTCAS_CTL.

12.1.2 Clock Controllers Block Diagrams

The following figure outlines the basic structure of the Clock Controller.

12.1.2.1 D1L1, D1L2(H) and D1M1 Clock Controller block diagram

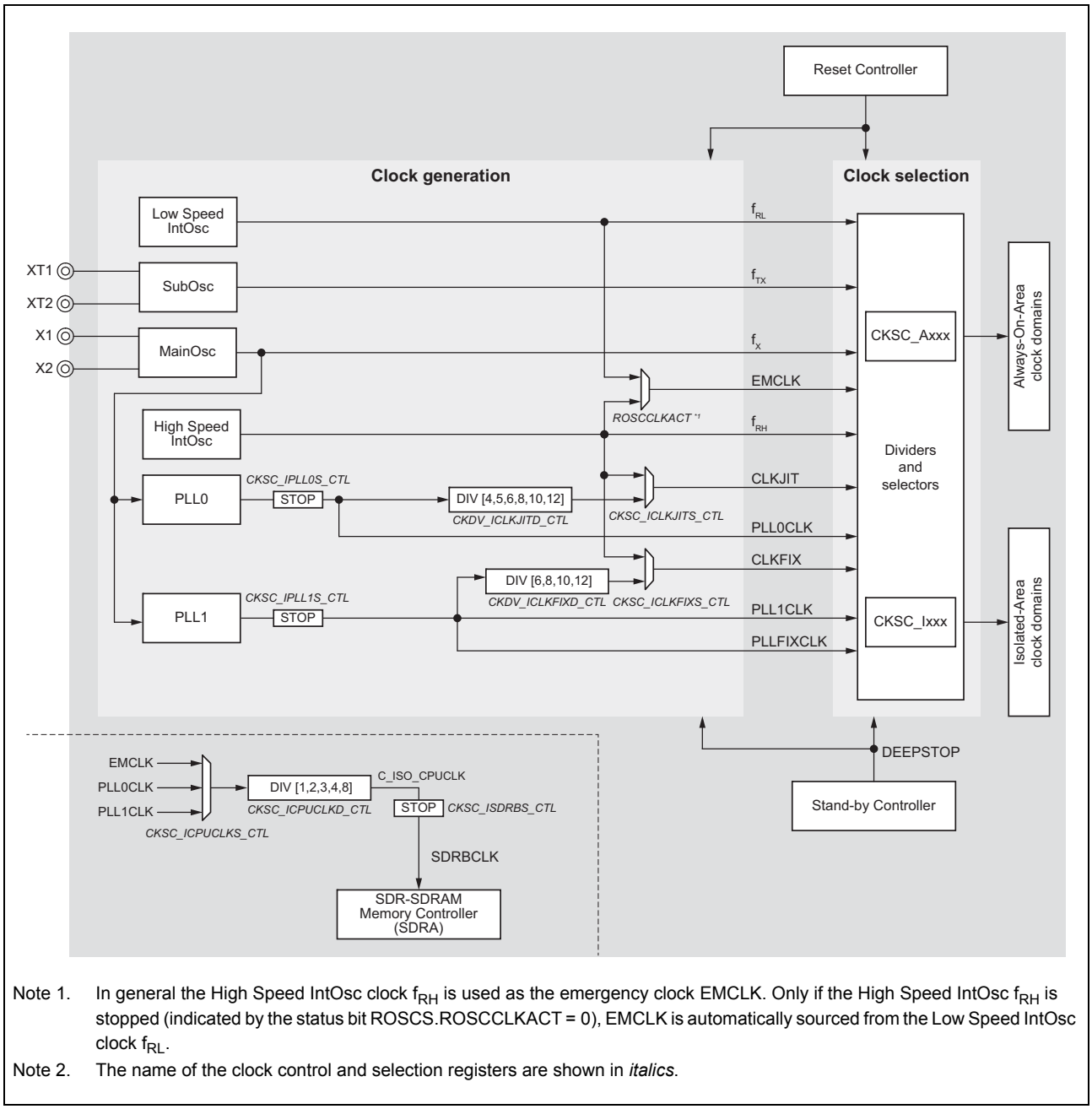


Note 1. In general the High Speed IntOsc clock f_{RH} is used as the emergency clock EMCLK. Only if the High Speed IntOsc clock f_{RH} is stopped (indicated by the status bit ROSCS.ROSCCLKACT = 0), EMCLK is automatically sourced from the Low Speed IntOsc clock f_{RL}.

Note 2. The name of the clock control and selection registers are shown in *italics*.

Figure 12.2 D1L1, D1L2(H) and D1M1 Clock Controller block diagram

12.1.2.2 D1M1H Clock Controller block diagram

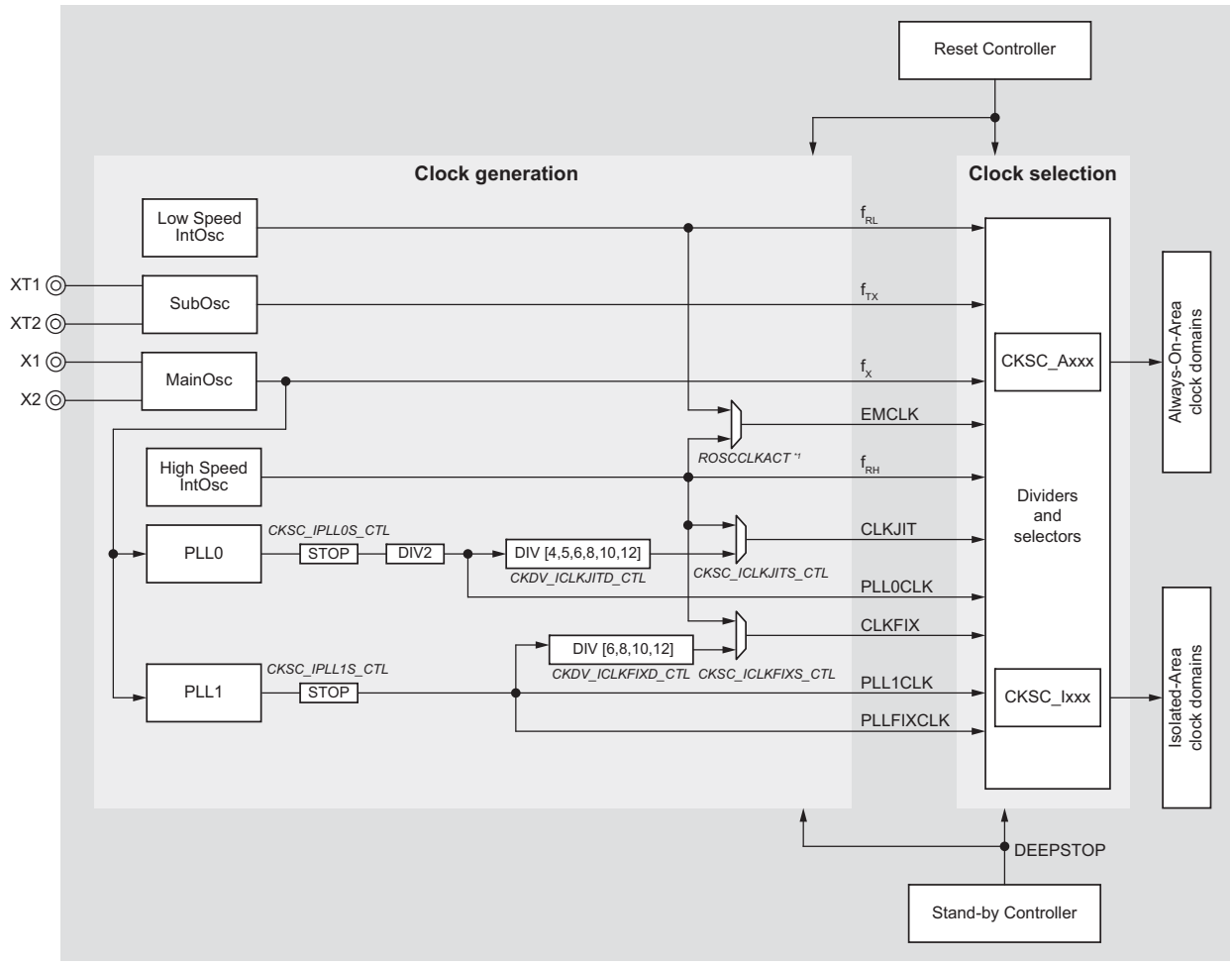


Note 1. In general the High Speed IntOsc clock f_{RH} is used as the emergency clock EMCLK. Only if the High Speed IntOsc f_{RH} is stopped (indicated by the status bit ROSCS.ROSCCLKACT = 0), EMCLK is automatically sourced from the Low Speed IntOsc clock f_{RL}.

Note 2. The name of the clock control and selection registers are shown in *italics*.

Figure 12.3 D1M1H Clock Controller block diagram

12.1.2.3 D1M1-V2 Clock Controller block diagram

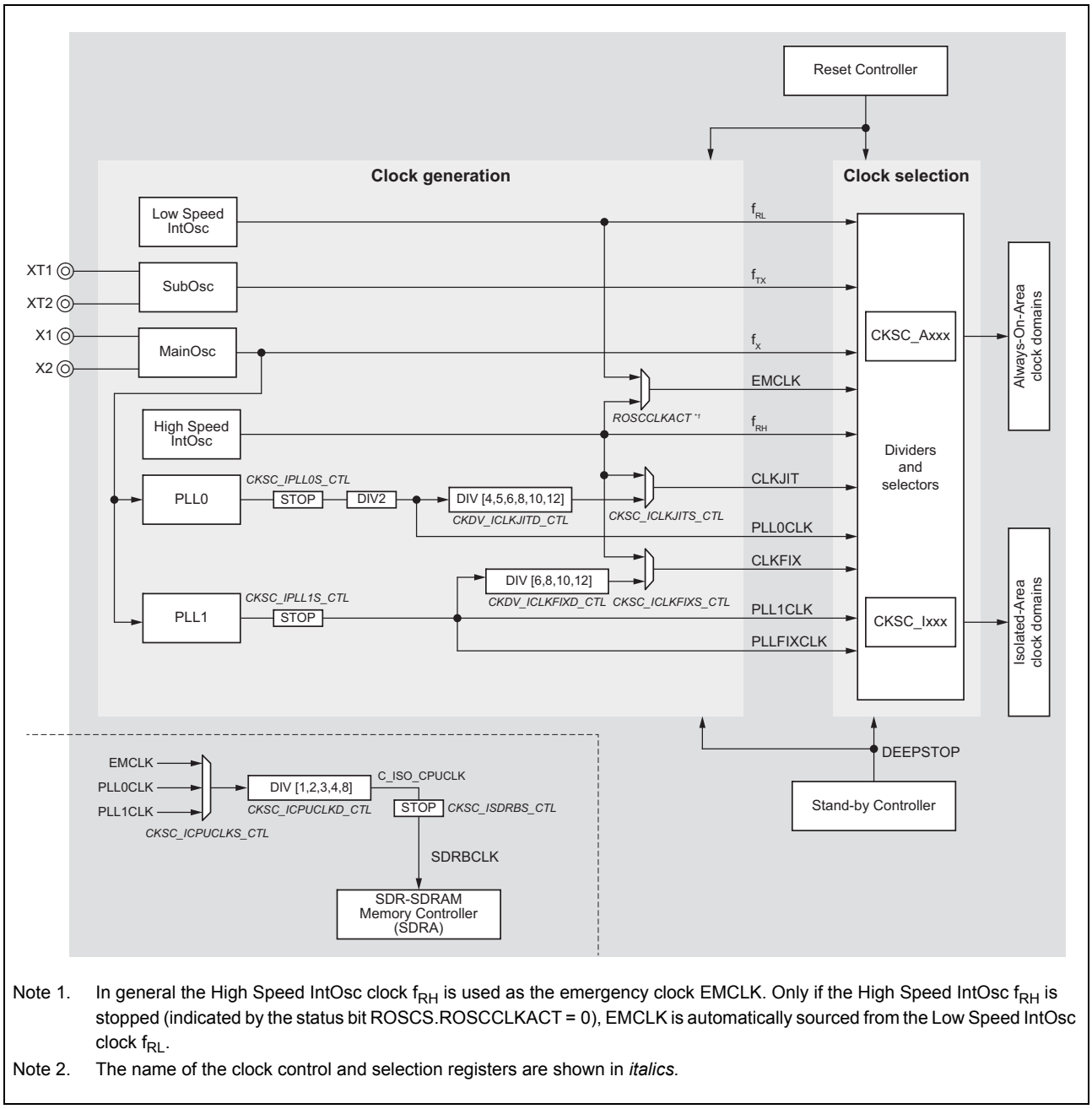


Note 1. In general the High Speed IntOsc clock f_{RH} is used as the emergency clock EMCLK. Only if the High Speed IntOsc clock f_{RH} is stopped (indicated by the status bit ROSCS.ROSCCLKACT = 0), EMCLK is automatically sourced from the Low Speed IntOsc clock f_{RL}.

Note 2. The name of the clock control and selection registers are shown in *italics*.

Figure 12.4 D1M1-V2 Clock Controller block diagram

12.1.2.4 D1M1A Clock Controller block diagram

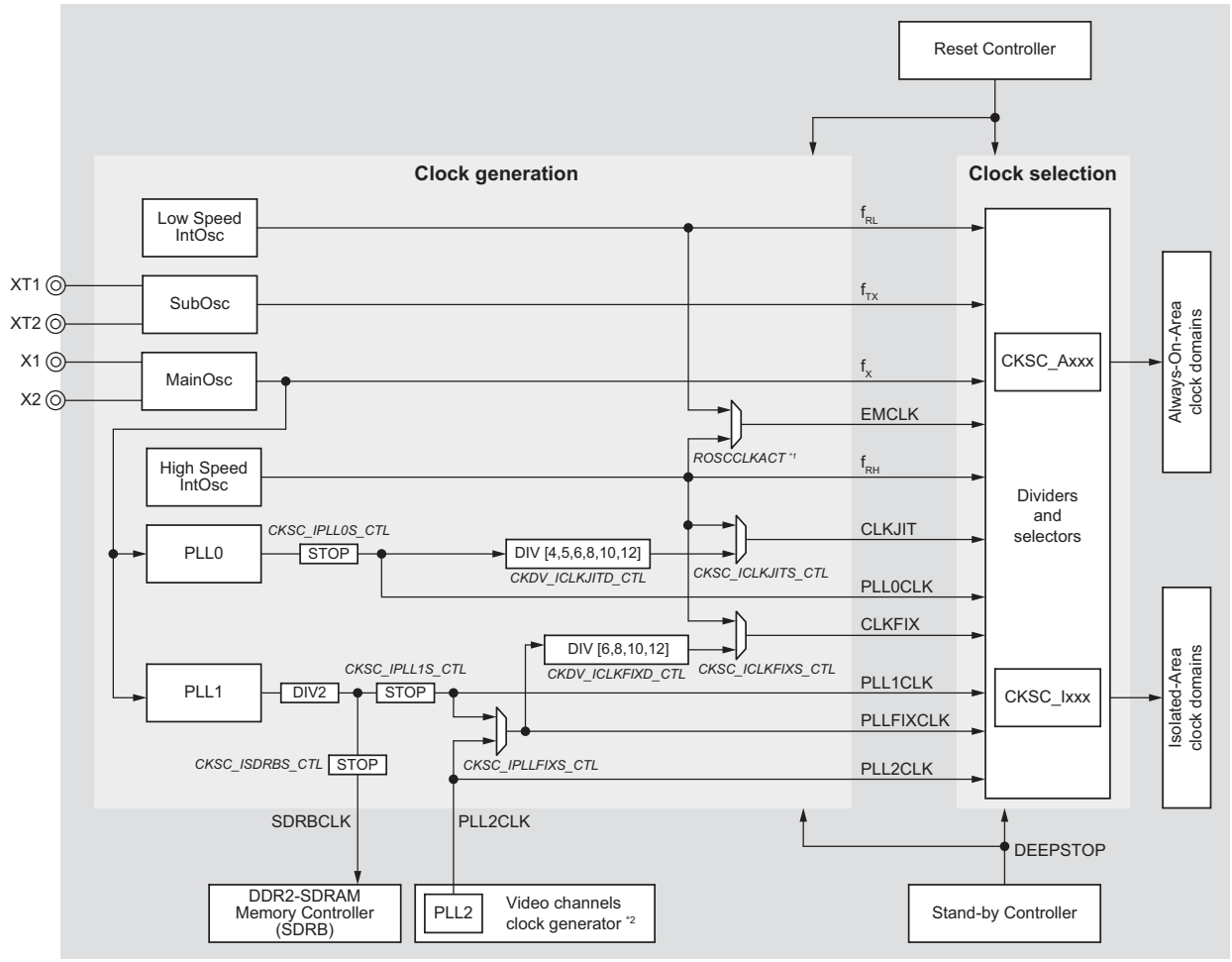


Note 1. In general the High Speed IntOsc clock f_{RH} is used as the emergency clock EMCLK. Only if the High Speed IntOsc f_{RH} is stopped (indicated by the status bit ROSCS.ROSCCLKACT = 0), EMCLK is automatically sourced from the Low Speed IntOsc clock f_{RL}.

Note 2. The name of the clock control and selection registers are shown in *italics*.

Figure 12.5 D1M1A Clock Controller block diagram

12.1.2.5 D1M2(H) Clock Controller block diagram



Note 1. In general the High Speed IntOsc clock f_{RH} is used as the emergency clock EMCLK. Only if the High Speed IntOsc f_{RH} is stopped (indicated by the status bit ROSCS.ROSCCLKACT = 0), EMCLK is automatically sourced from the Low Speed IntOsc clock f_{RL}.

Note 2. PLL2 is included in the video channels clock generator. Refer to Section 37.5, Video channels clock generation for details.

Note 3. The name of the clock control and selection registers are shown in *italics*.

Figure 12.6 D1M2(H) Clock Controller block diagram

NOTES

1. The control registers of the oscillators and PLL circuits in **Figure 12.2** to **Figure 12.6** are described in Section 12.3.2, Clock oscillators and PLL control registers.
2. The clock selection and control registers of the clock generation part in **Figure 12.2** to **Figure 12.6** are described in Section 12.3.3, Clock generator selection registers.
3. The clock selection and control registers of the clock selection part in **Figure 12.2** to **Figure 12.6** are described in Section 12.3.5, Always-On-Area clock domain selection registers and Section 12.3.6, Isolated-Area clock domain selection registers.

12.1.2.6 CPU Subsystem and bus clock domains

The base clock for the CPU Subsystem and all buses can be selected from various sources.

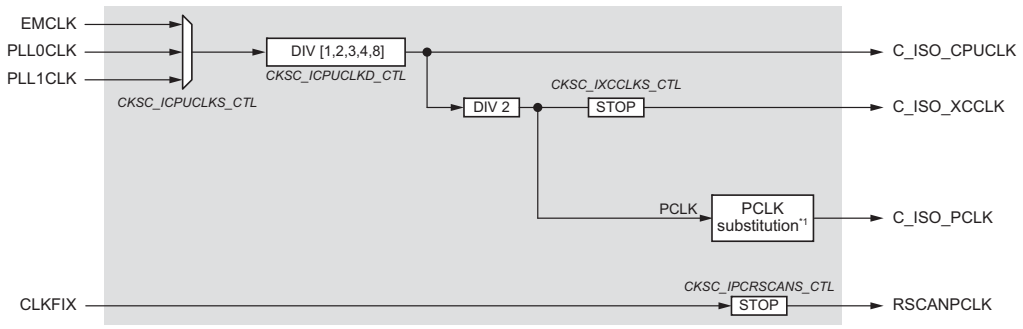
All bus clocks (C_ISO_XCCLK for Cross-connects, C_ISO_PCLK for PBUS) are derived from the CPU Subsystem clock domain C_ISO_CPUCLK.

Several bus clocks can be stopped individually, as shown in the figure below.

NOTE

The clock selection and control registers of the CPU and bus subsystems clock domains are described in Section 12.3.4, CPU Subsystems and bus clock domains selection registers.

(1) D1L1, D1L2(H) CPU and bus subsystems clock domains

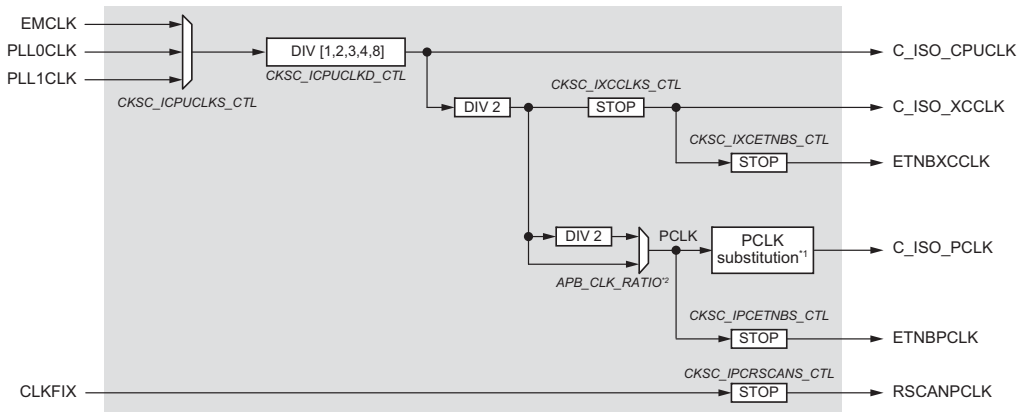


Note 1. Refer to Section 12.1.2.7, PCLK substitution for details.

Note: The name of the clock control and selection registers are shown in *italics*.

Figure 12.7 D1L1, D1L2(H) CPU and bus subsystems clock domains

(2) D1M1(H), D1M1-V2, D1M1A CPU and bus subsystems clock domains



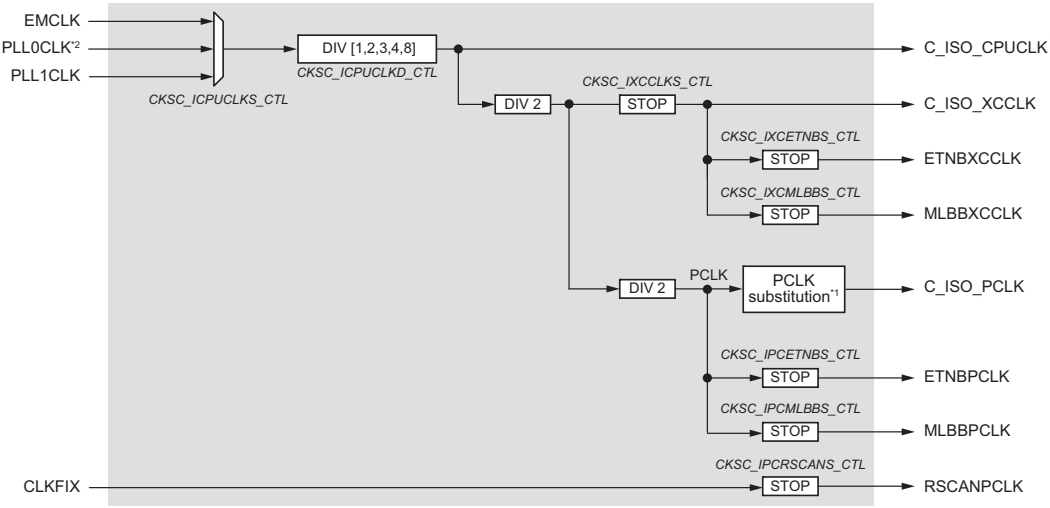
Note 1. Refer to Section 12.1.2.7, PCLK substitution for details.

Note 2. APB_CLK_RATIO selection register not available in D1M1-V2 and D1M1A.
D1M1-V2 implements fixed selection C_ISO_PCLK = C_ISO_XCCLK.
D1M1A implements fixed selection C_ISO_PCLK = C_ISO_XCCLK / 2.

Note: The name of the clock control and selection registers are shown in *italics*.

Figure 12.8 D1M1(H), D1M1-V2, D1M1A CPU and bus subsystems clock domains

(3) D1M2(H) CPU and bus subsystems clock domains



Note 1. Refer to Section 12.1.2.7, PCLK substitution for details.

Note 2. Selection of PLL0CLK for the C_ISO_CPUCLK clock domain is not allowed, when the DDR2-SDRAM Memory Controller (SDRB) is used.

Note: The name of the clock control and selection registers are shown in *italics*.

Figure 12.9 D1M2(H) CPU and bus subsystems clock domains

12.1.2.7 PCLK substitution

The Error Control Module is supplied by the C_ISO_PCLK clock. Since the Error Control Module's operation clock is also C_ISO_PCLK, the PCLK clock substitution ensures clock supply even if the selected source for C_ISO_PCLK fails to operate.

In the following situation the C_ISO_PCLK clock is switched to the High Speed IntOsc clock f_{RH}:

- The source of C_ISO_PCLK is the PLL0CLK or PLL1CLK:
  - CKSC_ICPUCLKS_CTL.CPUCLKSCSID[2:0] = 011_B for PLL0CLK
  - CKSC_ICPUCLKS_CTL.CPUCLKSCSID[2:0] = 100_B for PLL1CLK
- The selected PLL clock fails, which leads to an error detection of the respective Clock Monitor, thus
  - INTCLMA3TI is asserted, if PLL0CLK fails
  - INTCLMA4TI is asserted, if PLL1CLK fails
- CLMAOTCTL1.CLMA3CNT[3:0] cycles of the High Speed IntOsc clock f_{RH} have passed.

Since the Error Control Module is operating now with the f_{RH} clock, it handles the Clock Monitor error INTCLMA3TI or INTCLMA4TI.

The time between a Clock Monitor error detection and the switch of C_ISO_PCLK to f_{RH} is programmable via the CLMAOTCTL1 register.

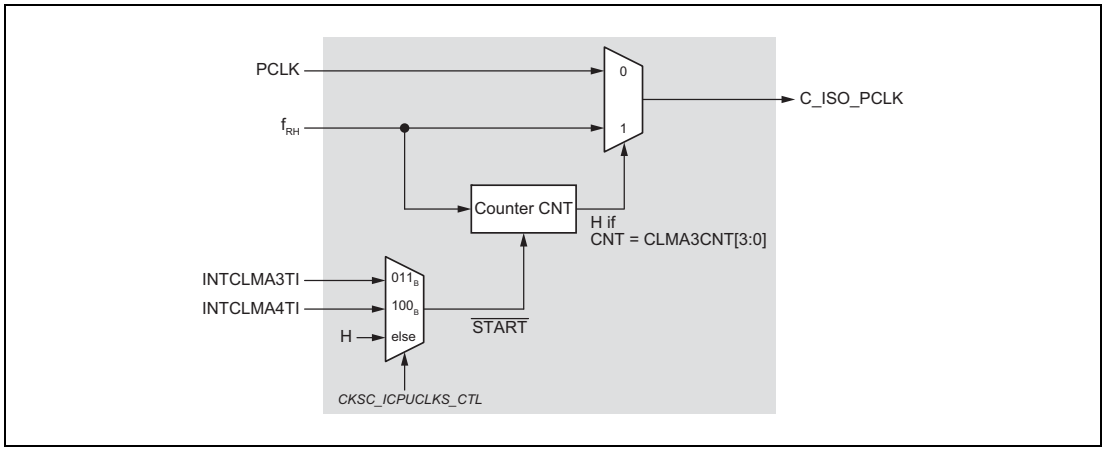


Figure 12.10 PCLK substitution

(1) CLMAOTCTL1 - PCLK substitution timing register

Access: This register is readable/writable in 8-bit units.

Address: FFF8 F2C4$_H$

Initial value: 0F$_H$

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	CLMA3CNT[3:0]			
Initial value	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 12.1 CLMAOTCTL1 Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When written, write the initial value.
3 to 0	CLMA3CNT[3:0]	Number of f_{RH} cycles between INTCLMA3TI or INTCLMA4TI Clock Monitor error detection and switch of C_ISO_PCLK to f_{RH}. **CAUTION** These bits must be written with more than 01$_H$.

NOTE

In the header files the module name of the above register is defined as:

CLMAC

12.1.3 Clock generators

Four clock generators are located on the Always-On-Area (AWO).
PLL0, PLL1 and PLL2 (for D1M2(H) only) are located on the Isolated-Area (ISO).

Low Speed Internal Oscillator (Low Speed IntOsc)

This oscillator generates the clock f_{RL}, which runs at a frequency of 240 kHz (Typ.). It starts operation after power up and can not be stopped, hence it is always operating.

High Speed Internal Oscillator (High Speed IntOsc)

The High Speed IntOsc generates a clock f_{RH}, which runs at a frequency of 8 MHz (Typ.).
It is automatically started after power up. It can be automatically stopped during DEEPSTOP mode and re-started after wake-up.

Sub Oscillator (SubOsc)

The SubOsc generates the sub-clock f_{TX}, which runs at a frequency of typical 32.768 kHz. Generation of the sub clock f_{TX} requires the connection of an external resonator to XT1 and XT2.
This clock is mainly used for real-time clock applications.

Main Oscillator (MainOsc)

The MainOsc generates the clock f_X.
Generation of the clock f_X requires the connection of an external resonator to X1 and X2.
The clock f_X is used as the reference clock for PLL0 and PLL1.

PLL0

The Spread Spectrum Phase Locked Loop (PLL0) generates the clock PLL0CLK. It is the main clock source for CPU and all busses. It is also the clock for jitter tolerant functional modules.

PLL1

PLL1 generates the PLL1CLK clock. It is the main clock for functional modules that need a fixed frequency.

PLL2 (D1M2(H) only)

PLL2 provides additional options for generating the video output pixel clocks of the D1M2(H) devices.

12.1.3.1 Clock generators reset

The clock generators on the Always-On-Area are reset by the PURES signal and the clock generator on the Isolated-Area is reset by the SYSRES signal or the ISORES signal.

NOTE

For the specifications of the frequencies, acceptable variation, and other parameters of the clock generators, refer to the Data Sheet.

12.1.4 Clock selectors

The clocks, generated by the clock generators, are input to the clock selectors CKSC_Axxx/CKSC_Ixxx.

Domain clocks CKSCLK_A<name>/CKSCLK_I<name> are selected by dedicated clock selectors from clocks directly input from the oscillators, or in some cases from clocks that have been divided by clock dividers.

- CKSC_IxxxS_CTL/CKSC_IxxxD_CTL registers: determine the clock for the Isolated-Area clock domains.
- CKSC_AxxxS_CTL/CKSC_AxxxD_CTL registers: determine the clock for the Always-On-Area clock domains.

Clock selectors reset

The clock selectors are reset by the SYSRES signal, which is asserted by all reset sources of the microcontroller. Hence after any reset the supply of all clock domains are set to their default configuration.

Clock stop request in DEEPSTOP mode

The Always-On-Area domain clocks can optionally be stopped in DEEPSTOP mode.

Continuation of stop of these clocks can be selected by use of individual stop mask registers.

CAUTION

Changing of one or all of this clock tree registers below must not be done again within a period of 20 µs after last change.

- **clock selector change of CKSC_IPLL0S_CTL**
- **clock selector change of CKSC_IPLL1S_CTL**
- **clock selector change of CKSC_ISDRBS_CTL**
- **clock selector change of CKSC_ICPUCLKS_CTL**
- **clock divider change of CKSC_ICPUCLKD_CTL**
- **PLL0 enable register PLL0E**
- **PLL1 enable register PLL1E**

CAUTION

You have to make sure that a clock source is active before you can change a clock selector already assigned to it.

Example:
CKSC_IDOTCLK0S_CTL defaults to PLL0PIXCLK which is stopped after reset. The PLL0PIXCLK must be started before the CKSC_IDOTCLK0S_CTL can be changed.

12.2 Clock Generators

12.2.1 Main Oscillator (MainOsc)

The Main Oscillator generates the clock f_X. f_X is also used as the PLL0 and PLL1 input clock PLL0CLKIN and PLL1CLKIN.

The following figure shows the basic structure and signals of the MainOsc.

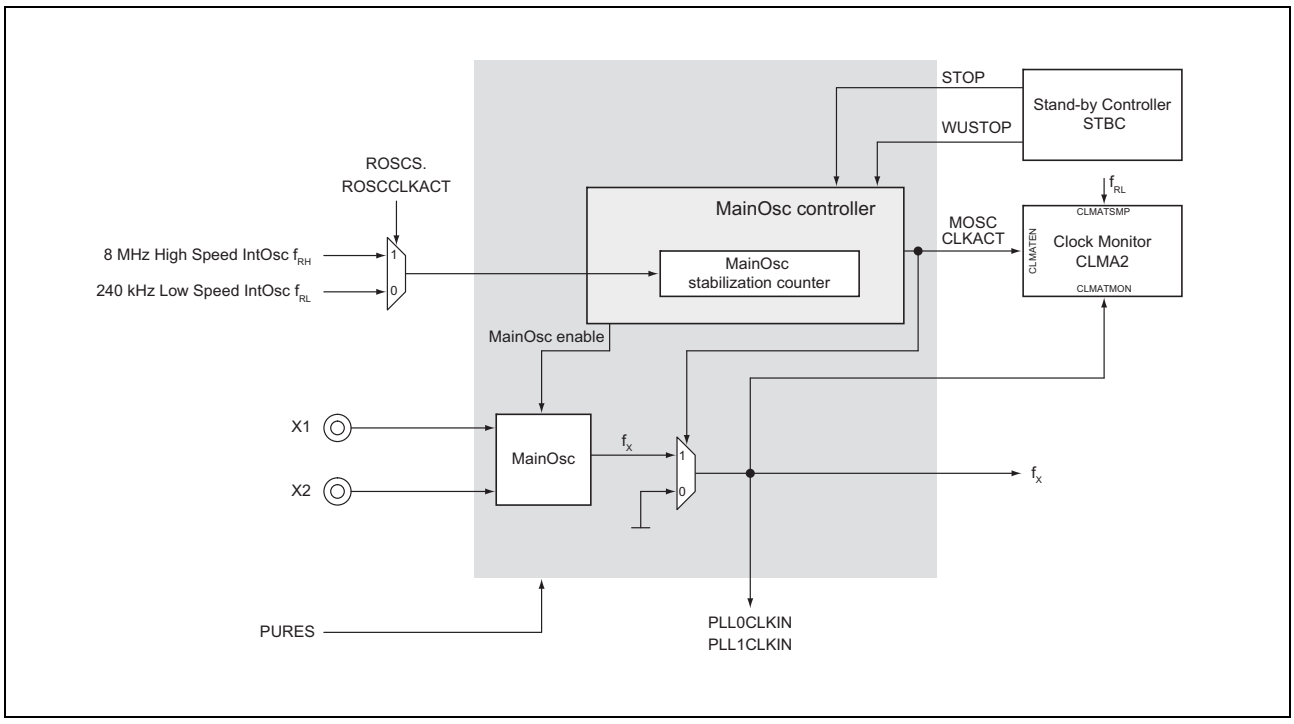


Figure 12.11 Main Oscillator (MainOsc)

The MainOsc is disabled after release from the power-up reset PURES state, and so must be enabled by setting the MainOsc enable trigger MOSCE.MOSCENTRG = 1.

MainOsc stabilization

The MOSCST.MOSCCLKST[16:0] bits set the MainOsc oscillation stabilization time.
The MainOsc stabilization counter counts the oscillation stabilization time with EMCLK as the clock source for counting. The range of settings is from 2^2 to (2^{17} –1), and is in cycles of EMCLK.
As long as the MainOsc is not stable, the MOSCCLKACT signal disables the f_X output.
If the MainOsc stabilization counter has reached the value, defined by MOSCST.MOSCCLKST[16:0], f_X is judged as stable and the change of MOSCCLKACT from 0 to 1 makes f_X available when a waveform is output from the MainOsc.

Stable and active f_X clock is indicated by MOSCS.MOSCCLKACT = 1.

MainOsc amplification gain

By using MOSCC.MOSCAMPSEL[1:0], the MainOsc's input frequency, determined by the external resonator, can be selected in the range from 8 MHz to 16 MHz.

MainOsc STOP requests in DEEPSTOP mode

The STOP signal from the Stand-by Controller requests the MainOsc Controller to switch off the f_X clock in DEEPSTOP mode.

The stop request mask bit MOSCSTPM.MOSCSTPMSK controls whether the MainOsc is stopped during DEEPSTOP or continues operation:

- MOSCSTPM.MOSCSTPMSK = 0:
  The STOP request signal is not masked, so the MainOsc is stopped in DEEPSTOP.
  If the MainOsc was in operation before DEEPSTOP, it is automatically re-started after wake-up from DEEPSTOP and the oscillation stabilization time is counted.
- MOSCSTPM.MOSCSTPMSK = 1:
  The STOP request signal is masked, so the MainOsc continues to operate in DEEPSTOP.

Clock Monitor control

The MainOsc activity signal MOSCCLKACT enables or disables supervision by the Clock Monitor CLMA2. In case the MainOsc is inactive (MOSCCLKACT = 0), supervision of its output clock f_X by CLMA2 is also deactivated.

MainOsc enable/disable trigger

The MainOsc can be enabled and disabled by the enable and disable trigger control bits:

- enable trigger MOSCE.MOSCENTRG = 1 starts the MainOsc
  Note that setting the enable trigger is only effective if the MainOsc is inactive, i.e. if MOSCS.MOSCCLKACT = 0.
- disable trigger MOSCE.MOSCDISTRG = 1 stops the MainOsc
  Note that setting the disable trigger is only effective if the MainOsc is active, i.e. if MOSCCLKACT = 1.

12.2.2 Sub Oscillator (SubOsc)

The Sub Oscillator generates the sub clock f_{TX}. f_{TX} has usually a frequency of 32.768 kHz and is used for the Real-time Clock.

The following figure shows the basic structure and signals of the SubOsc.

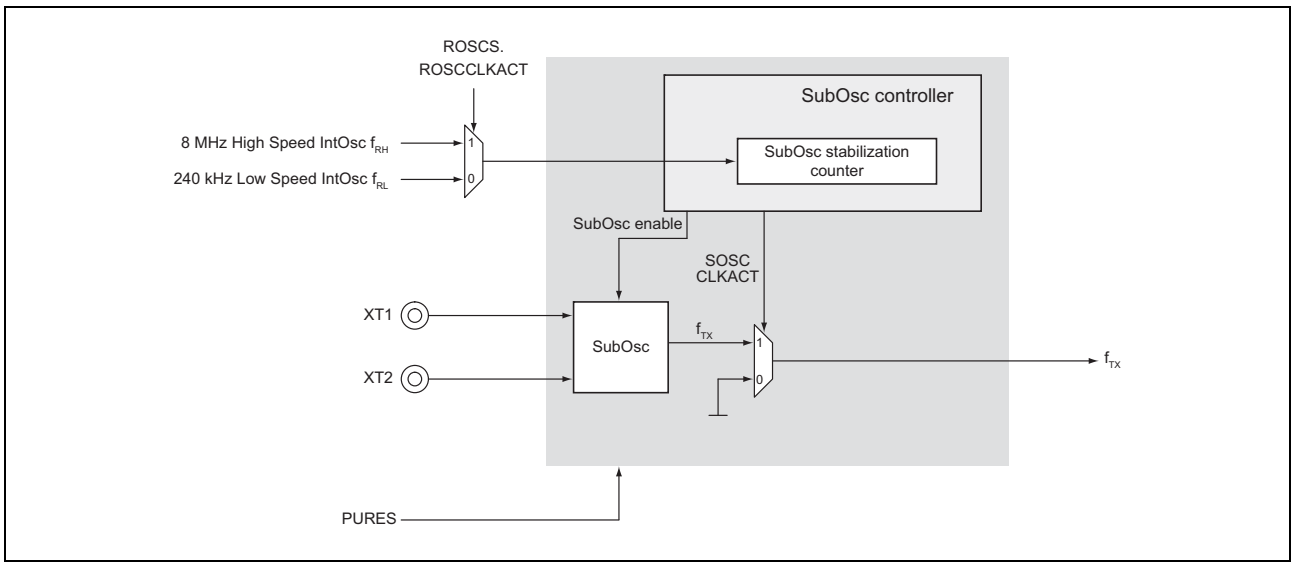


Figure 12.12 Sub Oscillator (SubOsc)

The SubOsc is disabled after release from the power-up reset PURES state, and so must be enabled by setting the SubOsc enable trigger SOSCE.SOSCENTRG = 1.

SubOsc stabilization

The SOSCST.SOSCCLKST[29:0] bits set the SubOsc oscillation stabilization time.

The SubOsc stabilization counter counts the oscillation stabilization time with EMCLK as the clock source for counting.

As long as the SubOsc is not stable, the SOSCCLKACT signal disables the f_{TX} output.

If the SubOsc stabilization counter has reached the value, defined by SOSCST.SOSCCLKST[29:0] bits, f_{TX} is judged as stable and the change of SOSCCLKACT from 0 to 1 makes f_{TX} available.

Secure the stabilization time longer than 2 seconds.

Stable and active f_{TX} clock is indicated by SOSCS.SOSCCLKACT = 1.

NOTE

The stabilization time of the SubOsc is specified in the Data Sheet.

SubOsc input frequencies

The SubOsc input frequency is typically 32.768 kHz.

SubOsc enable trigger/disable trigger

The SubOsc can be enabled or disabled by using the enable/disable trigger control bit.

- Enable trigger SOSCE.SOSCENTRG = 1 starts the SubOsc.
  Note that setting the enable trigger is only effective if the SubOsc is inactive, i.e. if SOSCS.SOSCCLKACT = 0.
- Disable trigger SOSCE.SOSCDISTRG = 1 stops the SubOSC.
  Note that setting the disable trigger is only effective if the SubOSC is active; that is, if SOSCS.SOSCCLKACT = 1.

12.2.3 High Speed Internal Oscillator (High Speed IntOsc)

The High Speed Internal Oscillator generates the clock f_{RH}. f_{RH} has a nominal frequency of 8 MHz.

The following figure shows the basic structure and signals of the High Speed IntOsc.

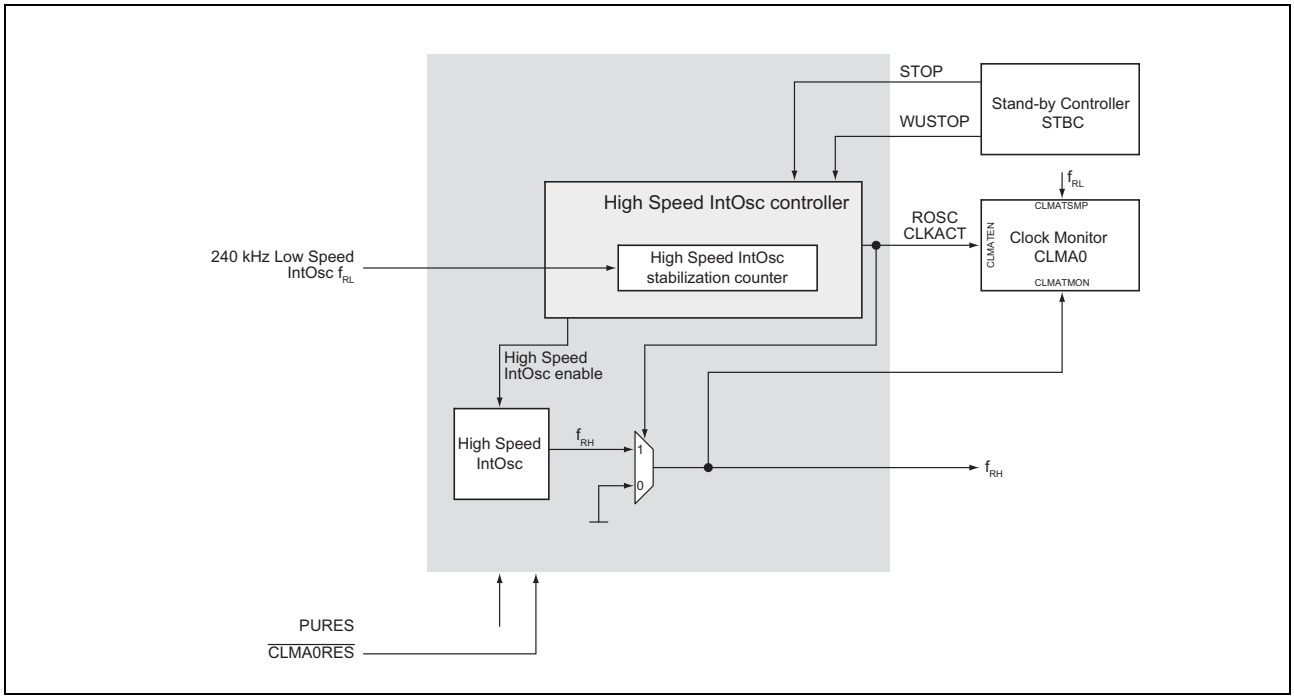


Figure 12.13 High Speed Internal Oscillator (High Speed IntOsc)

After reset release from the Power-up reset (PURES) or Clock Monitor 0 reset ($\overline{\text{CLMA0RES}}$) state the High Speed IntOsc starts operation.

NOTE

The High Speed IntOsc may only be stopped by software (using ROSCE.ROSCDISTRG = 1) after a CLMA0 reset. Thus CLMA0 reset indicates a High Speed IntOsc malfunction. In order to remain operable and to perform some emergency tasks, the software may stop the High Speed IntOsc and continue operation based on the Low Speed IntOsc.

High Speed IntOsc stabilization

Stable and active f_{RH} clock is indicated by ROSCS.ROSCCLKACT = 1.

NOTE

The stabilization time of the High Speed IntOsc is specified in the Data Sheet.

High Speed IntOsc STOP requests in DEEPSTOP mode

The STOP signal from the Stand-by Controller requests the High Speed IntOsc Controller to switch off the f_{RH} clock in DEEPSTOP mode.
The stop request mask bit ROSCSTPM.ROSCSTPMSK controls whether the High Speed IntOsc is stopped during DEEPSTOP or continues operation:

- ROSCSTPM.ROSCSTPMSK = 0:
  The STOP request signal is not masked, so the High Speed IntOsc is stopped in DEEPSTOP and automatically restarted after wake-up from DEEPSTOP.
- ROSCSTPM.ROSCSTPMSK = 1:
  The STOP request signal is masked, so the High Speed IntOsc continues to operate in DEEPSTOP.

Clock Monitor control

The High Speed IntOsc activity signal ROSCCLKACT enables or disables supervision by the Clock Monitor CLMA0. In case the High Speed IntOsc is inactive (ROSCCLKACT = 0), supervision of its output clock f_{RH} by CLMA0 is also deactivated.

The High Speed IntOsc clock f_{RH} is used as the sampling clock for Clock Monitor CLMA1.

12.2.4 Low Speed Internal Oscillator (Low Speed IntOsc)

The Low Speed Internal Oscillator generates the clock f_{RL}. f_{RL} has a nominal frequency of 240 kHz.

The following figure shows the basic structure and signals of the Low Speed IntOsc.

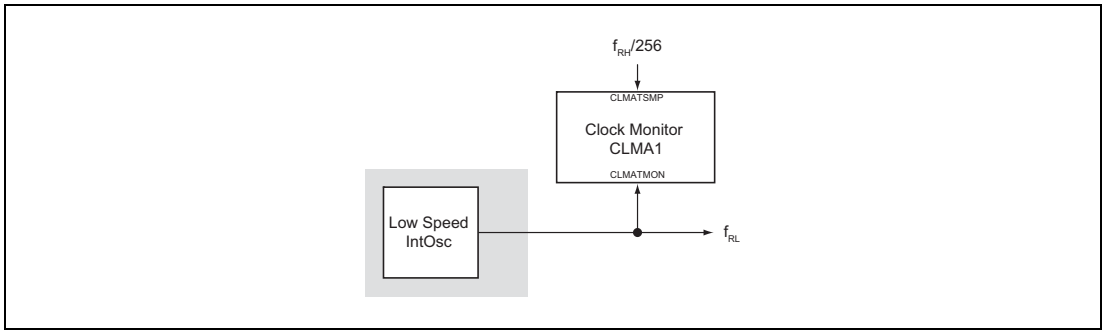


Figure 12.14 Low Speed Internal Oscillator (Low Speed IntOsc)

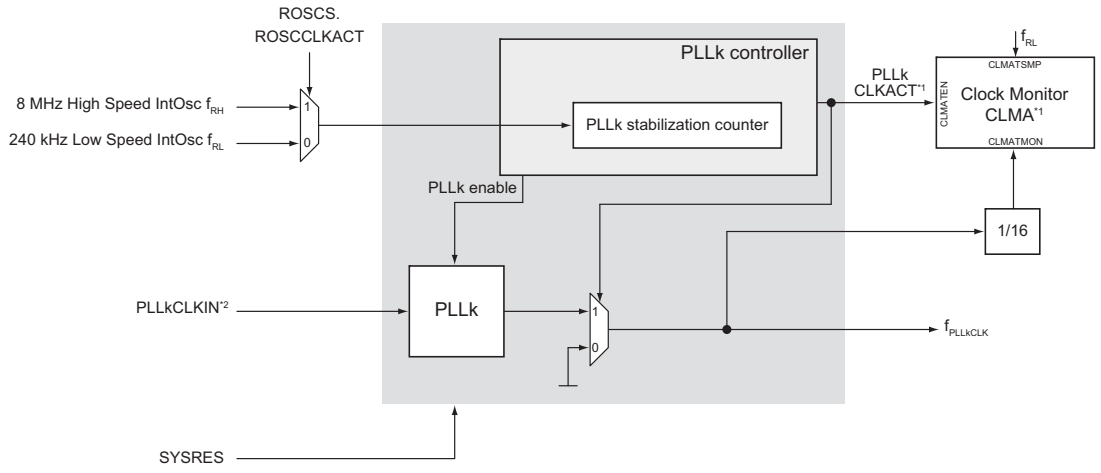
After reset release the Low Speed IntOsc starts operation. It can not be stopped.

The Low Speed IntOsc clock f_{RL} is used as the sampling clock for all Clock Monitors, except CLMA1.

12.2.5 PLLs

The RH850/D1L/D1M microcontrollers incorporate up to three PLL circuits.
PLLk with k = 0 to 2 identifies an individual unit. PLL2 is available only for D1M2(H) devices.

The following figure shows the basic structure and signals of the PLLk circuits.



Note 1. PLL0 output clock $f_{PLL0CLK}$/16 is monitored by Clock Monitor CLMA3.
PLL1 output clock $f_{PLL1CLK}$/16 is monitored by Clock Monitor CLMA4.
PLL2 output clock $f_{PLL2CLK}$ is not monitored.

Note 2. PLL0CLKIN = MainOsc clock f_X
PLL1CLKIN = MainOsc clock f_X
As for PLL2CLKIN refer Section 37.5, Video channels clock generation.

Figure 12.15 PLLk

All PLLs are disabled after release from the reset state, and so must be enabled by setting PLLkE.PLLkENTRG to 1.

PLL stabilization

The PLLk stabilization counter starts counting the stabilization time.
As long as the PLLk is not stable, the PLLkCLKACT signal disables the $f_{PLLkCLK}$ output.
If the PLLk stabilization counter has reached a predefined value, $f_{PLLkCLK}$ is judged as stable and the change of PLLkCLKACT from 0 to 1 makes $f_{PLLkCLK}$ available.

The stable and active state of the $f_{PLLkCLK}$ clock is indicated by PLLkS.PLLkCLKACT = 1.

NOTE

The stabilization time of the PLLs is specified in the Data Sheet.

PLLk in DEEPSTOP mode

Since all PLLk reside on the Isolated-Area, their power supply is switch off in DEEPSTOP mode.

Clock Monitor control

The PLL0 and PLL1 output clocks $f_{PLL0CLK}$ and $f_{PLL1CLK}$ are supervised by Clock Monitors.
The PLL0 and PLL1 activity signals PLL0CLKACT and PLL1CLKACT enable or disable supervision by the Clock Monitor CLMA3 and CLMA4. In case the PLL0 or PLL1 is inactive (PLL0CLKACT = 0 or PLL1CLKACT = 0), supervision of the output clocks $f_{PLL0CLK}$ or $f_{PLL1CLK}$ by CLMA3 or CLMA4 is also deactivated.

PLLk enable/disable trigger

The PLLk can be enabled and disabled by the enable and disable trigger control bits:

- enable trigger PLLkE.PLLkENTRG = 1 starts the PLLk
  Note that setting the enable trigger is only effective if the PLLk is inactive, i.e. if PLLkS.PLLkCLKACT = 0.
- disable trigger PLLkE.PLLkDISTRG = 0 stops the PLLk
  Note that setting the disable trigger is only effective if the PLLk is active, i.e. if PLLkS.PLLkCLKACT = 1.

12.2.6 PLLk parameters

The PLLk (k = 0 to 2) are configured by a set of parameters, derived from the control register PLLkC.

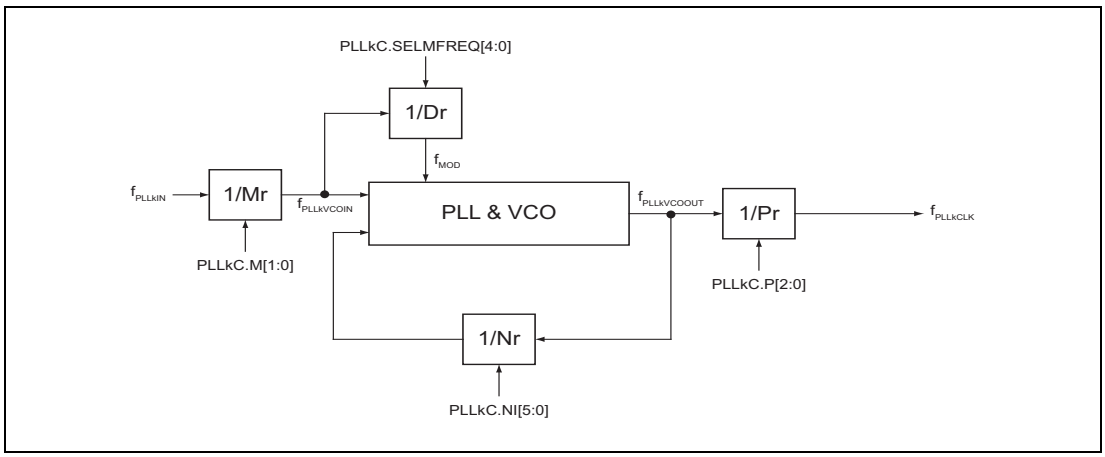


Figure 12.16 PLLk circuit

12.2.6.1 PLLk input clocks

- PLL0: f_{PLL0IN} = MainOsc f_X
- PLL1: f_{PLL1IN} = MainOsc f_X
- PLL2: f_{PLL2IN} is the frequency of the PLL2INCLK and depends on the setting of the CKSC_IPLL2INS_CTL clock selector and CKDV_IPLL2IND_CTL clock divider. Refer to Section 37.5, Video channels clock generation for details.

12.2.6.2 PLL0 modes

The PLL0 can used as a Spread Spectrum Clock Generator (SSCG) with dithering output clock:

• PLLkC.SSMODE[1:0] = $0x_H$: PLL mode with fixed output frequency, i.e. dithering disabled

• PLLkC.SSMODE[1:0] = 10_H: Down spread dithering mode

• PLLkC.SSMODE[1:0] = 11_H: Center spread dithering mode

NOTES

1. PLL1 and PLL2 can not be used in any dithering mode.
2. The output clock of PLL2 can be dithered, if PLL0 is operated in dithering mode and its output clock is selected as the PLL2 input clock via the CKSC_IPLL2INS_CTL clock selector.

12.2.6.3 PLLk output frequency

The PLLk output frequency $f_{PLLkCLK}$ is calculated as follows:

$$f_{PLLkCLK} = f_{PLLkIN} \bullet \frac{Nr}{Mr \bullet Pr}$$

The values Nr, Mr and Pr are derived from PLLkC register bits:

(1) Nr value

The Nr value is determined by

Nr = PLLkC.NI[5:0] + 1.

- PLL0:
  - D1L1, D1L2(H), D1M1(H), D1M2(H):
    Nr = PLL0C.NI[6:0] + 1
    Allowed range:
    $12 \le Nr \le 80$ thus PLL0C.NI[6:0] = $0B_H$ to $4F_H$
  - D1M1A, D1M1-V2:
    Nr = PLL0C.NI[6:0] + 1
    Allowed range:
    $24 \le Nr \le 160$ thus PLL0C.NI[6:0] = $0B_H$ to $4F_H$
- PLL1:
  - D1L1, D1L2(H), D1M1(H), D1M1A, D1M1-V2:
    Nr = PLL1C.NI[5:0] + 1
    Allowed range:
    $20 \le Nr \le 60$ thus PLL1C.NI[5:0] = 13_H to $3B_H$
  - D1M2(H):
    Nr = (PLL1C.NI[6:0] + 1) × 2
    Allowed range:
    $24 \le Nr \le 160$ thus PLL1C.NI[6:0] = $0B_H$ to $4F_H$
- PLL2: Nr = PLL2C.NI[5:0] + 1
  Allowed range:
  $20 \le Nr \le 60$ thus PLL2C.NI[5:0] = 13_H to $3B_H$

(2) Mr value

Mr is determined by PLLkC.M[1:0]:

Mr = PLLkC.M[1:0] + 1

Allowed range: $1 \le Mr \le 3$, thus PLLkC.M[1:0] = 0, 1, 2

(3) Pr value

Pr is determined by PLLkC.P[2:0] according to the following table:

	PLL0 output frequency			
PLL0C.P[2:0]	**Pr**	**D1L1, D1L2(H), D1M1(H), D1M2(H)**	**Pr**	**D1M1A, D1M1-V2**
000_B	1	320 MHz ≤ $f_{PLL0CLK}$ ≤ 480 MHz	1	533 MHz ≤ $f_{PLL0CLK}$ ≤ 960 MHz
001_B	2	160 MHz ≤ $f_{PLL0CLK}$ ≤ 320 MHz	2	266 MHz ≤ $f_{PLL0CLK}$ ≤ 533 MHz
010_B	4	80 MHz ≤ $f_{PLL0CLK}$ ≤ 160 MHz	4	160 MHz ≤ $f_{PLL0CLK}$ ≤ 266 MHz
011_B	8	40 MHz ≤ $f_{PLL0CLK}$ ≤ 80 MHz	8	66.6 MHz ≤ $f_{PLL0CLK}$ ≤ 160 MHz
100	16	20 MHz ≤ $f_{PLL0CLK}$ ≤ 40 MHz	16	40 MHz ≤ $f_{PLL0CLK}$ ≤ 66.6 MHz
101_B	–	Invalid	–	Invalid
110_B				
111_B				

	PLL1 output frequency			
PLL1C.P[2:0]	**Pr**	**D1L1, D1L2(H), D1M1(H), D1M1-V2, D1M1A**	**Pr**	**D1M2(H)**
000_B	1	240 MHz ≤ $f_{PLL1CLK}$ ≤ 480 MHz	1	533 MHz ≤ $f_{PLL1CLK}$ ≤ 960 MHz
001_B	2	120 MHz ≤ $f_{PLL1CLK}$ ≤ 240 MHz	2	266 MHz ≤ $f_{PLL1CLK}$ ≤ 533 MHz
010_B	4	60 MHz ≤ $f_{PLL1CLK}$ ≤ 120 MHz	4	160 MHz ≤ $f_{PLL1CLK}$ ≤ 266 MHz
011_B	6	40 MHz ≤ $f_{PLL1CLK}$ ≤ 80 MHz	8	66.6 MHz ≤ $f_{PLL1CLK}$ ≤ 133 MHz
100	8	30 MHz ≤ $f_{PLL1CLK}$ ≤ 40 MHz	16	40 MHz ≤ $f_{PLL1CLK}$ ≤ 66.6 MHz
101_B	16	20 MHz ≤ $f_{PLL1CLK}$ ≤ 30 MHz	–	Invalid
110_B	–	Invalid		
111_B				

PLL2C.P[2:0]	**Pr**	**PLL2 output frequency (D1M2(H))**
000_B	1	240 MHz ≤ $f_{PLL2CLK}$ ≤ 480 MHz
001_B	2	120 MHz ≤ $f_{PLL2CLK}$ ≤ 240 MHz
010_B	4	60 MHz ≤ $f_{PLL2CLK}$ ≤ 120 MHz
011_B	6	40 MHz ≤ $f_{PLL2CLK}$ ≤ 80 MHz
100	8	30 MHz ≤ $f_{PLL2CLK}$ ≤ 60 MHz
101_B	16	20 MHz ≤ $f_{PLL2CLK}$ ≤ 30 MHz
110_B	–	Invalid
111_B		

12.2.6.4 PLL0 frequency dithering parameters

If frequency dithering of the PLL0 is enabled (PLL0C.SSMODE[1:0] = $1x_H$), additional parameters must be set in PLL0C in order to specify the modulation frequency and range:

(1) PLL0C.SELMFREQ[4:0]: modulation frequency selection

The modulation frequency f_{MOD} is calculated as follows:

$$f_{MOD} = f_{PLL0VCOIN} / (4 \times Dr)$$

The factor Dr is determined by PLL0C.SELMFREQ[4:0].
Refer to Section 12.3.2.14, PLL0C — PLL0 control register for the relation between Dr and SELMFREQ[4:0].

The allowed range of f_{MOD} is 20 kHz ≤ f_{MOD} ≤ 100 kHz.

(2) PLLkC.SELMPERCENT[2:0]: dithering frequency range

The dithering frequency range is given in % of the PLL0 output clock frequency $f_{PLL0CLK}$.

- Down spread mode: - 1 % to -10 % of $f_{PLL0CLK}$
- Center spread mode: ± 1 % to ± 5 % of $f_{PLL0CLK}$

12.3 Registers

This section contains a description of all registers of the Clock Controller.

12.3.1 Clock Controller registers overview

The Clock Controller is controlled and operated by means of the following registers:

Table 12.2 List of Clock Controller registers (1/3)

Module Name	Register Name	Symbol	Address
Clock oscillators and PLL control registers:			
SYS	MainOsc enable register	MOSCE	FFF8 1100$_H$
SYS	MainOsc status register	MOSCS	FFF8 1104$_H$
SYS	MainOsc control register	MOSCC	FFF8 1108$_H$
SYS	MainOsc stabilization time register	MOSCST	FFF8 110C$_H$
SYS	MainOsc stop mask register	MOSCSTPM	FFF8 1118$_H$
SYS	SubOsc enable register	SOSCE	FFF8 1200$_H$
SYS	SubOsc status register	SOSCS	FFF8 1204$_H$
SYS	SubOsc stabilization time register	SOSCST	FFF8 120C$_H$
SYS	High-speed IntOsc enable register	ROSCE	FFF8 1000$_H$
SYS	High Speed IntOsc status register	ROSCS	FFF8 1004$_H$
SYS	High-speed IntOsc stop mask register	ROSCSTPM	FFF8 1018$_H$
SYS	PLL0 enable register	PLL0E	FFF8 9000$_H$
SYS	PLL0 status register	PLL0S	FFF8 9004$_H$
SYS	PLL0 control register	PLL0C	FFF8 9008$_H$
SYS	PLL1 enable register	PLL1E	FFF8 9100$_H$
SYS	PLL1 status register	PLL1S	FFF8 9104$_H$
SYS	PLL1 control register	PLL1C	FFF8 9108$_H$
SYS	PLL2 enable register	PLL2E	FFF8 50C0$_H$
SYS	PLL2 status register	PLL2S	FFF8 50C4$_H$
SYS	PLL2 control register	PLL2C	FFF8 50C8$_H$
Clock generator selection registers:			
SYS	PLL0CLK clock control register	CKSC_IPLL0S_CTL	FFF8 9200$_H$
SYS	PLL0CLK clock active register	CKSC_IPLL0S_ACT	FFF8 9208$_H$
SYS	PLL1CLK clock control register	CKSC_IPLL1S_CTL	FFF8 9300$_H$
SYS	PLL1CLK clock active register	CKSC_IPLL1S_ACT	FFF8 9308$_H$
SYS	CLKJIT clock divider register	CKDV_ICLKJITD_CTL	FFF8 53C0$_H$
SYS	CLKJIT clock divider status register	CKDV_ICLKJITD_STAT	FFF8 53C4$_H$
SYS	CLKJIT source clock selection register	CKSC_ICLKJITS_CTL	FFF8 5400$_H$
SYS	CLKJIT source clock active register	CKSC_ICLKJITS_ACT	FFF8 5408$_H$
SYS	CLKFIX clock divider register	CKDV_ICLKFIXD_CTL	FFF8 5700$_H$
SYS	CLKFIX clock divider status register	CKDV_ICLKFIXD_STAT	FFF8 5704$_H$
SYS	CLKFIX source clock selection register	CKSC_ICLKFIXS_CTL	FFF8 5740$_H$
SYS	CLKFIX source clock active register	CKSC_ICLKFIXS_ACT	FFF8 5748$_H$
SYS	PLLFIXCLK source clock selection register	CKSC_IPLLFIXS_CTL	FFF8 5000$_H$
SYS	PLLFIXCLK source clock active register	CKSC_IPLLFIXS_ACT	FFF8 5008$_H$
SYS	SDRBCLK clock control register	CKSC_ISDRBS_CTL	FFF8 5140$_H$

Table 12.2 List of Clock Controller registers (2/3)

Module Name	Register Name	Symbol	Address
SYS	SDRBCLK clock active register	CKSC_ISDRBS_ACT	FFF8 5148$_H$
CPU Subsystems and bus clock domains selection registers:			
SYS	C_ISO_CPUCLK source clock selection register	CKSC_ICPUCLKS_CTL	FFF8 A000$_H$
SYS	C_ISO_CPUCLK source clock active register	CKSC_ICPUCLKS_ACT	FFF8 A008$_H$
SYS	C_ISO_CPUCLK clock divider register	CKSC_ICPUCLKD_CTL	FFF8 A100$_H$
SYS	C_ISO_CPUCLK clock divider active register	CKSC_ICPUCLKD_ACT	FFF8 A108$_H$
SYS	C_ISO_XCCLK clock control register	CKSC_IXCCLKS_CTL	FFF8 5180$_H$
SYS	C_ISO_XCCLK clock active register	CKSC_IXCCLKS_ACT	FFF8 5188$_H$
SYS	ETNBPCLK clock control register	CKSC_IPCETNBS_CTL	FFF8 5280$_H$
SYS	ETNBPCLK clock active register	CKSC_IPCETNBS_ACT	FFF8 5288$_H$
SYS	ETNBXCCLK clock control register	CKSC_IXCETNBS_CTL	FFF8 51C0$_H$
SYS	ETNBXCCLK clock active register	CKSC_IXCETNBS_ACT	FFF8 51C8$_H$
SYS	MLBBPCLK clock control register	CKSC_IPCMLBBS_CTL	FFF8 52C0$_H$
SYS	MLBBPCLK clock active register	CKSC_IPCMLBBS_ACT	FFF8 52C8$_H$
SYS	MLBBXCCLK clock control register	CKSC_IXCMLBBS_CTL	FFF8 5200$_H$
SYS	MLBBXCCLK clock active register	CKSC_IXCMLBBS_ACT	FFF8 5208$_H$
SYS	RSCANPCLK clock control register	CKSC_IPCRSCANS_CTL	FFF8 5240$_H$
SYS	RSCANPCLK clock active register	CKSC_IPCRSCANS_ACT	FFF8 5248$_H$
PWRG	PBUS clock ratio selection register	APB_CLK_RATIO	FFF8 F510$_H$
Always-On-Area clock domain selection registers:			
SYS	C_AWO_AWOT source clock selection register	CKSC_AAWOTS_CTL	FFF8 2100$_H$
SYS	C_AWO_AWOT source clock active register	CKSC_AAWOTS_ACT	FFF8 2108$_H$
SYS	C_AWO_AWOT clock divider register	CKSC_AAWOTD_CTL	FFF8 2200$_H$
SYS	C_AWO_AWOT clock divider active register	CKSC_AAWOTD_ACT	FFF8 2208$_H$
SYS	C_AWO_AWOT stop mask register	CKSC_AAWOTD_STPM	FFF8 2218$_H$
SYS	C_AWO_WDTA0 clock divider register	CKSC_AWDTA0D_CTL	FFF8 2000$_H$
SYS	C_AWO_WDTA0 clock divider active register	CKSC_AWDTA0D_ACT	FFF8 2008$_H$
SYS	C_AWO_RTCA source clock selection register	CKSC_ARTCAS_CTL	FFF8 2300$_H$
SYS	C_AWO_RTCA source clock active register	CKSC_ARTCAS_ACT	FFF8 2308$_H$
SYS	C_AWO_RTCA clock divider register	CKSC_ARTCAD_CTL	FFF8 2400$_H$
SYS	C_AWO_RTCA clock divider active register	CKSC_ARTCAD_ACT	FFF8 2408$_H$
SYS	C_AWO_RTCA stop mask register	CKSC_ARTCAD_STPM	FFF8 2418$_H$
SYS	C_AWO_FOUT source clock selection register	CKSC_AFOUTS_CTL	FFF8 2700$_H$
SYS	C_AWO_FOUT source clock active register	CKSC_AFOUTS_ACT	FFF8 2708$_H$
SYS	C_AWO_FOUT stop mask register	CKSC_AFOUTS_STPM	FFF8 2718$_H$
Isolated-Area clock domain selection registers:			
SYS	C_ISO_MLBB source clock selection register	CKSC_IMLBBS_CTL	FFF8 5300$_H$
SYS	C_ISO_MLBB source clock active register	CKSC_IMLBBS_ACT	FFF8 5308$_H$
SYS	C_ISO_SFMA source clock selection register	CKSC_ISFMAS_CTL	FFF8 5340$_H$
SYS	C_ISO_SFMA source clock active register	CKSC_ISFMAS_ACT	FFF8 5348$_H$
SYS	C_ISO_SFMA clock divider register	CKSC_ISFMAD_CTL	FFF8 5380$_H$
SYS	C_ISO_SFMA clock divider active register	CKSC_ISFMAD_ACT	FFF8 5388$_H$
SYS	C_ISO_RSCAN source clock divider register	CKSC_IRSCAND_CTL	FFF8 5780$_H$
SYS	C_ISO_RSCAN source clock active register	CKSC_IRSCAND_ACT	FFF8 5788$_H$

Table 12.2 List of Clock Controller registers (3/3)

Module Name	Register Name	Symbol	Address
SYS	C_ISO_RSCANXIN clock selection register	CKSC_IRSCANXINS_CTL	FFF8 5800$_H$
SYS	C_ISO_RSCANXIN source clock active register	CKSC_IRSCANXINS_ACT	FFF8 5808$_H$
SYS	C_ISO_SSIF clock divider register	CKDV_ISSIFD_CTL	FFF8 57C0$_H$
SYS	C_ISO_SSIF clock divider active register	CKDV_ISSIFD_STAT	FFF8 57C4$_H$
SYS	C_ISO_TAUB01 source clock selection register	CKSC_ITAUB01S_CTL	FFF8 5540$_H$
SYS	C_ISO_TAUB01 source clock active register	CKSC_ITAUB01S_ACT	FFF8 5548$_H$
SYS	C_ISO_TAUB2 source clock selection register	CKSC_ITAUB2S_CTL	FFF8 5580$_H$
SYS	C_ISO_TAUB2 source clock active register	CKSC_ITAUB2S_ACT	FFF8 5588$_H$
SYS	C_ISO_TAUJ source clock selection register	CKSC_ITAUJS_CTL	FFF8 55C0$_H$
SYS	C_ISO_TAUJ source clock active register	CKSC_ITAUJS_ACT	FFF8 55C8$_H$
SYS	C_ISO_OSTM source clock selection register	CKSC_IOSTMS_CTL	FFF8 5600$_H$
SYS	C_ISO_OSTM source clock active register	CKSC_IOSTMS_ACT	FFF8 5608$_H$
SYS	C_ISO_LCBI source clock selection register	CKSC_ILCBIS_CTL	FFF8 54C0$_H$
SYS	C_ISO_LCBI clock active register	CKSC_ILCBIS_ACT	FFF8 54C8$_H$
SYS	C_ISO_ADCE clock divider register	CKSC_IADCED_CTL	FFF8 5480$_H$
SYS	C_ISO_ADCE clock divider active register	CKSC_IADCED_ACT	FFF8 5488$_H$
SYS	C_ISO_ISM source clock selection register	CKSC_IISMS_CTL	FFF8 5440$_H$
SYS	C_ISO_ISM source clock active register	CKSC_IISMS_ACT	FFF8 5448$_H$
SYS	C_ISO_RLIN source clock selection register	CKSC_IRLINS_CTL	FFF8 5500$_H$
SYS	C_ISO_RLIN source clock active register	CKSC_IRLINS_ACT	FFF8 5508$_H$
Write protection registers:			
SYS	Control protection command register	PROTCMD0	FFF8 0000$_H$
SYS	Control protection status register	PROTS0	FFF8 0004$_H$
SYS	Isolated-Area clock control protection command register 1	PROTCMD1	FFF8 8000$_H$
SYS	Isolated-Area clock control protection status register 1	PROTS1	FFF8 8004$_H$
SYS	Isolated-Area clock control protection command register 2	PROTCMDD1	FFF8 7000$_H$
SYS	Isolated-Area clock control protection status register 2	PROTSD1	FFF8 7004$_H$
PWRG	Protection command register	PROTCMDPWRGD	FFF8 F580$_H$
PWRG	Protection status register	PROTSPWRGD	FFF8 F584$_H$

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> is defined in the above table.

12.3.2 Clock oscillators and PLL control registers

12.3.2.1 MOSCE — MainOsc enable register

This register is used to start and stop the MainOsc.

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 1100$_H$

Initial value: 0000 0000$_H$ This register is initialized by the power-up reset PURES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSC DISTRG	MOSC ENTRG
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12.3 MOSCE register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	MOSCDISTRG	MainOsc disable trigger*1 0: no function 1: stops the MainOsc
0	MOSCENTRG	MainOsc enable trigger 0: no function 1: starts the MainOsc

Note 1. <Recommended procedure for stopping operations when MainOsc is stopped by MOSCDISTRG>
1. Check that there is no clock domain for which MainOsc is selected.
   If MainOsc is selected for a clock domain, disable the setting or select a clock source other than MainOsc.
2. Stop the MainOsc (MOSCE.MOSCDISTRG = 1)
3. Confirm that the MainOsc has been stopped (MOSCS.MOSCCLKACT = 0)

12.3.2.2 MOSCS — MainOsc status register

This register provides activity status information about the MainOsc.

Access: This register can be read in 32-bit units.

Address: FFF8 1104$_H$

Initial value: 0000 0000$_H$ This register is initialized by the power-up reset PURES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCCLKACT	—*1	—*1
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12.4 MOSCS register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2	MOSCCLKACT	MainOsc activation status 0: MainOsc is inactive 1: MainOsc is active
1, 0	Reserved	When written, write the initial value.

12.3.2.3 MOSCC — MainOsc control register

This register is used to specify amplification gain of the MainOsc.

This register can only be written, if the MainOsc is disabled.

Access: This register can be read/written in 32-bit units.

Address: FFF8 1108_H

Initial value: 0000 0004_H This register is initialized by the power-up reset PURES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—*1	MOSCAMPSEL [1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Note 1. Do not change the default value “1” of this bit.

Table 12.5 MOSCC register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
2	–	**CAUTION** The default value “1” of this bit must not be changed.
1, 0	MOSC AMPSEL[1:0]	MainOsc frequency selection

AMPSEL[1:0]	MainOsc frequency
00_B	8 MHz < $f_X \leq$ 16 MHz
01_B	8 MHz
10_B	8 MHz Low Power Mode
11_B	8 MHz Very Low Power Mode

NOTE

Standard oscillator circuits usually do not support Low Power Modes or Very Low Power Modes. Please ask your oscillator circuit supplier for special oscillators to support these modes.

12.3.2.4 MOSCST — MainOsc stabilization time register

This register determines the MainOsc stabilization time.

This register can only be written, if the MainOsc is disabled.

Access: This register can be read/written in 32-bit units.

Address: FFF8 110C$_H$

Initial value: 0000 44C0$_H$ This register is initialized by the power-up reset PURES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCCLKST16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MOSCCLKST[15:0]															
Initial value	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.6 MOSCST register contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When written, write the initial value.
16 to 0	MOSC CLKST[16:0]	MOSCCLKST[16:0] defines the number EMCLK cycles, used as the MainOsc stabilization time. • If High Speed IntOsc active (ROSCS.ROSCCLKACT = 1): Stabilization time = MOSCCLKST[16:0] / f_{RH} • If High Speed IntOsc inactive (ROSCS.ROSCCLKACT = 0): Stabilization time = MOSCCLKST[16:0] / f_{RL}

NOTE

Refer to the Data Sheet for information about the MainOsc stabilization time.

12.3.2.5 MOSCSTPM — MainOsc Stop Mask Register

Access: This register can be read/written in 32-bit units.

Address: FFF8 1118$_H$

Initial value: 0000 0002$_H$. This register is initialized by the power-up reset PURES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOSCSTPMSK
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.7 MOSCSTPM register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	MOSCSTPMSK	MainOsc stop request mask 0: MainOsc stops operation in DEEPSTOP mode. 1: MainOsc continues operation in DEEPSTOP mode

12.3.2.6 SOSCE — SubOsc enable register

This register is used to start and stop the SubOsc.

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 1200_H

Initial value: $0000\ 0000_H$. This register is initialized by the power-up reset PURES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOSC DISTRG	SOSC ENTRG
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12.8 SOSCE register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	SOSCDISTRG	SubOsc disable trigger*1 0: No function 1: Stop SubOsc
0	SOSCENTRG	SubOsc enable trigger 0: no function 1: starts SubOsc

Note 1. <Recommended procedure for stopping operations when SubOsc is stopped by SOSCDISTRG>
1. Check that there is no clock domain for which SubOsc is selected.
   If SubOsc is selected for a clock domain, disable the setting or select a clock source other than SubOsc.
2. Stop the SubOsc (SOSCE.SOSCDISTRG = 1)
3. Confirm that the SubOsc has been stopped (SOSCS.SOSCCLKACT = 0)

12.3.2.7 SOSCS — SubOsc status register

This register provides activity status information about the SubOsc.

Access: This register can be read in 32-bit units.

Address: FFF8 1204_H

Initial value: 0000 0000_H This register is initialized by the power-up reset PURES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SOSCCLKACT	—*1	—*1
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12.9 SOSCS register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2	SOSCCLKACT	SubOsc activation status 0: SubOsc is inactive 1: SubOsc is active
1, 0	Reserved	When written, write the initial value.

12.3.2.8 SOSCST — SubOsc stabilization time register

This register determines the SubOsc stabilization time.

This register can only be written, if the SubOsc is disabled.

Access: This register can be read/written in 32-bit units.

Address: FFF8 120C$_H$

Initial value: 010C 8E00$_H$. This register is initialized by the power-up reset PURES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SOSCCLKST[29:16]													
Initial value	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SOSCCLKST[15:0]															
Initial value	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.10 SOSCST register contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
29 to 0	SOSC CLKST[29:0]	The SOSCCLKST[29:0] bits specify the count for the SubOsc stabilization time counter. • If HS IntOsc is active (ROSCS.ROSCCLKACT = 1): stabilization time = SOSCCLKST[29:0] / f_{RH} • If HS IntOsc is inactive (ROSCS.ROSCCLKACT = 0): stabilization time = SOSCCLKST[29:0] / f_{RL}

NOTE

Refer to the Data Sheet for information about the SubOsc stabilization time.

CAUTION

Set this register when SubOsc has been stopped.

12.3.2.9 ROSCE — High Speed IntOsc enable register

This register is used to stop the High Speed IntOsc operation.

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 1000_H

Initial value: $0000\ 0000_H$. This register is initialized by the power-up reset PURES and the Clock Monitor 0 reset CLMA0RES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSC DISTRG	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 12.11 ROSCE register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	ROSCDISTRG	High Speed IntOsc disable trigger*1 0: no function 1: High Speed IntOsc stopped
0	Reserved	When written, write the initial value.

Note 1. <Recommended procedure for stopping operations when High-Speed IntOsc is stopped by ROSCDISTRG>
1. Check that there is no clock domain for which High-Speed IntOsc is selected.
   If High-Speed IntOsc is selected for a clock domain, disable the setting or select a clock source other than High-Speed IntOsc.
2. Stop the High-Speed IntOsc (ROSCE.ROSCDISTRG = 1)
3. Confirm that the High-Speed IntOsc has been stopped (ROSCS.ROSCCLKACT = 0).

12.3.2.10 ROSCS — High Speed IntOsc status register

This register provides activity status information about the High Speed IntOsc.

Access: This register can be read in 32-bit units.

Address: FFF8 1004_H

Initial value: $\overline{0000\ 0007_H}$. This register is initialized by the power-up reset PURES and the Clock Monitor 0 reset $\overline{CLMA0RES}$.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCCLKACT	—*1	—*1
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12.12 ROSCS register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2	ROSCCLKACT	High Speed IntOsc activation status 0: High Speed IntOsc is inactive 1: High Speed IntOsc is active
1, 0	Reserved	When written, write the initial value.

12.3.2.11 ROSCSTPM — High Speed IntOsc Stop Mask Register

Access: This register can be read/written in 32-bit units.

Address: FFF8 1018_H

Initial value: $0000\ 0002_H$. This register is initialized by the power-up reset PURES and the Clock Monitor 0 reset $\overline{CLMA0RES}$.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ROSCSTPMSK
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.13 ROSCSTPM register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	ROSCSTPMSK	High Speed IntOsc stop request mask in DEEPSTOP mode 0: High Speed IntOsc stops operation in DEEPSTOP mode. 1: High Speed IntOsc continues operation in DEEPSTOP mode Note that the High Speed IntOSC can be stopped in the case the High Speed IntOSC disable trigger ROSCE.ROSCDISTRG is set to 1 regardless of the setting of this bit.

NOTE

Except in D1M1A and D1M1-V2, there are possibility that MCU does not wake-up. When using the interrupt signals INTP0 to INTP10 for wake-up factor, set the input pulse width long enough or configure that high speed internal oscillator does not stop in DEEPSTOP (ROSCSTPM.ROSCSTPMSK = 1). To refer the sufficient value, see RH850/D1L/D1M Group DATASHEET.

12.3.2.12 PLL0E — PLL0 enable register

This register is used to start and stop the PLL0.

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 9000_H

Initial value: $0000\ 0000_H$. This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0 DISTRG	PLL0 ENTRG
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12.14 PLL0E register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	PLL0DISTRG	PLL0 disable trigger*1 0: no function 1: stops PLL0
0	PLL0ENTRG	PLL0 enable trigger*2 0: no function 1: starts PLL0

Note 1. <Recommended procedure for stopping operations when PLL0 is stopped by PLL0EDISTRG>
1. Check that there is no clock domain for which PLL0 is selected.
If PLL0 is selected for a clock domain, disable the setting or select a clock source other than PLL0.
2. Stop the PLL0 (PLL0E.PLL0DISTRG = 1)
3. Confirm that the PLL0 has been stopped (PLL0S.PLL0CLKACT = 0).

Note 2. Before enabling the PLL0 by PLL0ENTRG, confirm that the MainOsc is active.

12.3.2.13 PLL0S — PLL0 status register

This register provides activity status information about the PLL0.

Access: This register can be read in 32-bit units.

Address: FFF8 9004$_H$

Initial value: 0000 0000$_H$. This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0CLKACT	—*1	—*1
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12.15 PLL0S register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2	PLL0CLKACT	PLL0 activation status 0: PLL0 is inactive 1: PLL0 is active
1, 0	Reserved	When written, write the initial value.

12.3.2.14 PLL0C — PLL0 control register

This register is used to specify the PLL0 output clock frequencies $f_{PLL0CLK}$.

This register can only be written, if the PLL0 is disabled.

Access: This register can be read/written in 32-bit units.

Address: FFF8 9008_H

Initial value: 6000 $003B_H$. This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FVV[2:0]			SELMFREQ[4:0]					–	SELMPERCENT[2:0]			–	–	–	–
Initial value	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	SSMODE[1:0]		M[1:0]		P[2:0]			–	NI[6:0]						
Initial value	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.16 PLL0C register contents (1/5)

Bit Position	Bit Name	Function
31 to 29	FVV[2:0]	PLL0 VCO output frequency $f_{PLL0VCOOUT}$ range

- D1L1, D1L2(H), D1M1(H), D1M2(H):

FVV[2:0]	VCO frequency range
000_B	320 MHz < $f_{PLL0VCOOUT}$ ≤ 360 MHz
001_B	360 MHz < $f_{PLL0VCOOUT}$ ≤ 400 MHz
010_B	400 MHz < $f_{PLL0VCOOUT}$ ≤ 440 MHz
011_B	440 MHz < $f_{PLL0VCOOUT}$ ≤ 480 MHz
100_B	480 MHz < $f_{PLL0VCOOUT}$ ≤ 520 MHz
101_B	520 MHz < $f_{PLL0VCOOUT}$ ≤ 560 MHz
110_B	560 MHz < $f_{PLL0VCOOUT}$ ≤ 600 MHz
111_B	600 MHz < $f_{PLL0VCOOUT}$ ≤ 640 MHz

- D1M1A, D1M1-V2:

FVV[2:0]	VCO frequency range
000_B	640 MHz < $f_{PLL0VCOOUT}$ ≤ 720 MHz
001_B	720 MHz < $f_{PLL0VCOOUT}$ ≤ 800 MHz
010_B	800 MHz < $f_{PLL0VCOOUT}$ ≤ 880 MHz
011_B	880 MHz < $f_{PLL0VCOOUT}$ ≤ 960 MHz
100_B	Invalid
101_B	
110_B	
111_B	

Table 12.16 PLL0C register contents (2/5)

Bit Position	Bit Name	Function
28 to 24	SELMFREQ[4:0]	Modulation frequency selection The modulation frequency is calculated as follows: $f_{MOD} = f_{PLL0VCOIN} / (4 \times Dr)$ The allowed range of f_{MOD} is 20 kHz ≤ f_{MOD} ≤ 100 kHz
23	Reserved	When written, write the initial value.

SELMFREQ[4:0]	Dr
00000_B	10
00001_B	12
00010_B	18
00011_B	20
00100_B	22
00101_B	26
00110_B	28
00111_B	30
01000_B	34
01001_B	38
01010_B	40
01011_B	44
01100_B	50
01101_B	56
01110_B	58
01111_B	60
$1\ 0000_B$	62
$1\ 0001_B$	66
$1\ 0010_B$	72
$1\ 0011_B$	76
$1\ 0100_B$	80
$1\ 0101_B$	84
$1\ 0110_B$	86
$1\ 0111_B$	100
$1\ 1000_B$	120
$1\ 1001_B$	126
$1\ 1010_B$	134
$1\ 1011_B$	150
$1\ 1100_B$	166
$1\ 1101_B$	200
$1\ 1110_B$	250
$1\ 1111_B$	300

Table 12.16 PLL0C register contents (3/5)

Bit Position	Bit Name	Function
22 to 20	SELMPERCENT [2:0]	Dithering frequency range The range is given in % of the VCO output frequency $f_{PLL0VCOOUT}$.

SELMPERCENT[2:0]	Down spread	Center spread
000_B	- 1 %	Invalid
001_B	- 2 %	± 1 %
010_B	- 3 %	Invalid
011_B	- 4 %	± 2 %
100_B	- 5 %	Invalid
101_B	- 6 %	± 3 %
110_B	- 8 %	± 4 %
111_B	- 10 %	± 5 %

19 to 15	Reserved	When written, write the initial value.
14 to 13	SSMODE[1:0]	Frequency dithering mode selection

SSMODE[1:0]	Frequency dithering mode
00_B	Fixed frequency (dithering disabled)
01_B	
10_B	Down spread dithering
11_B	Center spread dithering

12 to 11	M[1:0]	Mr divider selection

M[1:0]	Mr-Value
00_B	1
01_B	2
10_B	3
11_B	Invalid

Table 12.16 PLL0C register contents (4/5)

Bit Position	Bit Name	Function
10 to 8	P[2:0]	Pr divider selection

- D1L1, D1L2(H), D1M1(H), D1M2(H)

P[2:0]	Pr-Value	PLL output frequency range
000_B	1	320 MHz ≤ $f_{PLL0CLK}$ ≤ 480 MHz
001_B	2	160 MHz ≤ $f_{PLL0CLK}$ ≤ 320 MHz
010_B	4	80 MHz ≤ $f_{PLL0CLK}$ ≤ 160 MHz
011_B	8	40 MHz ≤ $f_{PLL0CLK}$ ≤ 80 MHz
100_B	16	20 MHz ≤ $f_{PLL0CLK}$ ≤ 40 MHz
101_B		
110_B		
111_B		

- D1M1A, D1M1-V2

P[2:0]	Pr-Value	PLL output frequency range
000_B	1	533 MHz ≤ $f_{PLL0CLK}$ ≤ 960 MHz
001_B	2	266 MHz ≤ $f_{PLL0CLK}$ ≤ 533 MHz
010_B	4	160 MHz ≤ $f_{PLL0CLK}$ ≤ 266 MHz
011_B	8	66.6 MHz ≤ $f_{PLL0CLK}$ ≤ 160 MHz
100_B	16	40 MHz ≤ $f_{PLL0CLK}$ ≤ 66.6 MHz
101_B	–	Invalid
110_B		
111_B		

7	Reserved	When written, write the initial value.

Table 12.16 PLL0C register contents (5/5)

Bit Position	Bit Name	Function
6 to 0	NI[6:0]	Nr divider selection: integer part

- D1L1, D1L2(H), D1M1(H), D1M2(H)

NI[6:0]	Integer part of Nr
000 0000_B	Invalid
...	
000 1010_B	
000 1011_B	12
000 1100_B	Invalid
000 1101_B	14
...	(Only even values are available.)
100 1110_B	Invalid
100 1111_B	80
101 0000_B	Invalid
...	
111 1111_B	

- D1M1A, D1M1-V2

NI[6:0]	Integer part of Nr
000 0000_B	Invalid
...	
000 1010_B	
000 1011_B	24
000 1100_B	26
...	(Only even values are available.)
100 1110_B	158
100 1111_B	160
101 0000_B	Invalid
...	
111 1111_B	

12.3.2.15 PLL1E — PLL1 enable register

This register is used to start and stop the PLL1.

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 9100_H

Initial value: $0000\ 0000_H$. This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1 DISTRG	PLL1 ENTRG
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12.17 PLL1E register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	PLL1DISTRG	PLL1 disable trigger*1 0: no function 1: stops PLL1
0	PLL1ENTRG	PLL1 enable trigger*2 0: no function 1: starts PLL1

Note 1. Before stopping the PLL1 by PLL1DISTRG, confirm that the PLL1 output clock is not used as the source clock for any clock domain.

Note 2. Before enabling the PLL1 by PLL1ENTRG, confirm that the MainOsc is active.

12.3.2.16 PLL1S — PLL1 status register

This register provides activity status information about the PLL1.

Access: This register can be read in 32-bit units.

Address: FFF8 9104$_H$

Initial value: 0000 0000$_H$. This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1CLKACT	—*1	—*1
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12.18 PLL1S register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2	PLL1CLKACT	PLL1 activation status 0: PLL1 is inactive 1: PLL1 is active
1, 0	Reserved	When written, write the initial value.

12.3.2.17 PLL1C — PLL1 control register

This register is used to specify the PLL1 output clock frequencies $f_{PLL1CLK}$.

This register can only be written, if the PLL1 is disabled.

Access: This register can be read/written in 32-bit units.

Address: FFF8 9108_H

Initial value:
- D1L1, D1L2(H), D1M1(H), D1M1-V2, D1M1A: 0000 $003B_H$.
- D1M2(H): 6000 $001D_H$.

This register is initialized by any reset.

D1L1, D1L2(H), D1M1(H), D1M1-V2, D1M1A:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	M[1:0]		P[2:0]			—	—	NI[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

D1M2(H):

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FVV[2:0]			—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	M[1:0]		P[2:0]			—	NI[6:0]						
Initial value	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.19 PLL1C register contents (1/3)

Bit Position	Bit Name	Function
31 to 29	Reserved	D1L1, D1L2(H), D1M1(H), D1M1-V2, D1M1A: When written, write the initial value.
	FVV[2:0]	D1M2(H) PLL1 VCO output frequency $f_{PLL1VCOOUT}$ range

FVV[2:0]	PLL1 VCO frequency range
000_B	640 MHz < $f_{PLL1VCOOUT}$ ≤ 720 MHz
001_B	720 MHz < $f_{PLL1VCOOUT}$ ≤ 800 MHz
010_B	800 MHz < $f_{PLL1VCOOUT}$ ≤ 880 MHz
011_B	880 MHz < $f_{PLL1VCOOUT}$ ≤ 960 MHz
100_B	Invalid
101_B	Invalid
110_B	Invalid
111_B	Invalid

28 to 13	Reserved	When written, write the initial value.
12 to 11	M[1:0]	Mr divider selection

M[1:0]	Mr-Value
00_B	1
01_B	2
10_B	3
11_B	Invalid

Table 12.19 PLL1C register contents (2/3)

Bit Position	Bit Name	Function
10 to 8	P[2:0]	D1L1, D1L2(H), D1M1(H), D1M1-V2, D1M1A: Pr divider selection

P[2:0]	Pr-Value	PLL1 output frequency range
000_B	1	240 MHz ≤ $f_{PLL1CLK}$ ≤ 480 MHz
001_B	2	120 MHz ≤ $f_{PLL1CLK}$ ≤ 240 MHz
010_B	4	60 MHz ≤ $f_{PLL1CLK}$ ≤ 120 MHz
011_B	6	40 MHz ≤ $f_{PLL1CLK}$ ≤ 80 MHz
100_B	8	30 MHz ≤ $f_{PLL1CLK}$ ≤ 40 MHz
101_B	16	20 MHz ≤ $f_{PLL1CLK}$ ≤ 30 MHz
110_B	–	Invalid
111_B		

D1M2(H):
Pr divider selection

P[2:0]	Pr-Value	PLL1 output frequency range
000_B	1	533 MHz ≤ $f_{PLL1CLK}$ ≤ 960 MHz
001_B	2	266 MHz ≤ $f_{PLL1CLK}$ ≤ 533 MHz
010_B	4	160 MHz ≤ $f_{PLL1CLK}$ ≤ 266 MHz
011_B	8	66.6 MHz ≤ $f_{PLL1CLK}$ ≤ 160 MHz
100_B	16	40 MHz ≤ $f_{PLL1CLK}$ ≤ 66.6 MHz
101_B	–	Invalid
110_B		
111_B		

7	Reserved	When written, write the initial value.
6	Reserved	D1L1, D1L2(H), D1M1(H), D1M1-V2, D1M1A: When written, write the initial value.
5 to 0	NI[5:0]	D1L1, D1L2(H), D1M1(H), D1M1-V2, D1M1A: Nr divider selection: integer part

NI[5:0]	Integer part of Nr
$00\ 0000_B$	Invalid
...	
$01\ 0010_B$	
$01\ 0011_B$	20
$01\ 0100_B$	Invalid
$01\ 0101_B$	22
...	(Only even values are available)
$11\ 1011_B$	60
$11\ 1100_B$	Invalid
...	
$11\ 1111_B$	

Table 12.19 PLL1C register contents (3/3)

Bit Position	Bit Name	Function
6 to 0	NI[6:0]	D1M2(H): Nr divider selection: integer part

NI[6:0]	Integer part of Nr
$000\ 0000_B$	Invalid
...	
$000\ 1010_B$	
$000\ 1011_B$	24
$000\ 1100_B$	26
...	(Only even values are available.)
$100\ 1110_B$	158
$100\ 1111_B$	160
$111\ 1100_B$	Invalid
...	
$111\ 1111_B$	

12.3.2.18 PLL2E — PLL2 enable register (D1M2(H) only)

This register is used to start and stop the PLL2.

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 50C0$_H$

Initial value: 0000 0000$_H$. This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL2 DISTRG	PLL2 ENTRG
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12.20 PLL2E register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	PLL2DISTRG	PLL2 disable trigger*1 0: no function 1: stops PLL2
0	PLL2ENTRG	PLL2 enable trigger*2 0: no function 1: starts PLL2

Note 1. Before stopping the PLL2 by PLL2DISTRG, confirm that the PLL2 output clock is not used as the source clock for any clock domain.

Note 2. Before enabling the PLL2 by PLL2ENTRG, confirm that the selected PLL2 input clock is active. Refer to Section 37.5, Video channels clock generation for details about the PLL2 clock input selections.

12.3.2.19 PLL2S — PLL2 status register (D1M2(H) only)

This register provides activity status information about the PLL2.

Access: This register can be read in 32-bit units.

Address: FFF8 50C4$_H$

Initial value: 0000 0000$_H$. This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL2CLKACT	—*1	—*1
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. The values of bit 1 and 0 are undefined.
After masking bit 1 and 0, check only bit 2 to verify the status.

Table 12.21 PLL2S register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2	PLL2CLKACT	PLL2 activation status 0: PLL2 is inactive 1: PLL2 is active
1, 0	Reserved	When written, write the initial value.

12.3.2.20 PLL2C — PLL2 control register (D1M2(H) only)

This register is used to specify the PLL2 output clock frequencies $f_{PLL2CLK}$.

This register can only be written, if the PLL2 is disabled.

Access: This register can be read/written in 32-bit units.

Address: FFF8 50C8$_H$

Initial value: 0001 002F$_H$. This register is initialized by any reset.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	M[1:0]		P[2:0]			—	—	NI[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.22 PLL2C register contents (1/2)

Bit Position	Bit Name	Function
31 to 13	Reserved	When written, write the initial value.
12 to 11	M[1:0]	Mr divider selection

M[1:0]	Mr-Value	PLL2 input frequency f_{PLL2IN}
00_B	1	8 MHz ≤ f_{PLL2IN} < 16 MHz
01_B	2	16 MHz ≤ f_{PLL2IN} < 24 MHz
10_B	3	f_{PLL2IN} = 24 MHz
11_B	Setting prohibited	

Bit Position	Bit Name	Function
10 to 8	P[2:0]	Pr divider selection

P[2:0]	Pr-Value	PLL output frequency range
000_B	1	240 MHz ≤ $f_{PLL2CLK}$ ≤ 480 MHz
001_B	2	120 MHz ≤ $f_{PLL2CLK}$ ≤ 240 MHz
010_B	4	60 MHz ≤ $f_{PLL2CLK}$ ≤ 120 MHz
011_B	6	40 MHz ≤ $f_{PLL2CLK}$ ≤ 80 MHz
100_B	8	30 MHz ≤ $f_{PLL2CLK}$ ≤ 60 MHz
101_B	16	20 MHz ≤ $f_{PLL2CLK}$ ≤ 30MHz
110_B	Setting prohibited	
111_B		

Bit Position	Bit Name	Function
7 to 6	Reserved	When written, write the initial value.

Table 12.22 PLL2C register contents (2/2)

Bit Position	Bit Name	Function
5 to 0	NI[5:0]	Nr divider selection: integer part

NI[5:0]	Integer part of Nr
$00\ 0000_B$	Invalid
...	
$01\ 0010_B$	
$01\ 0011_B$	20
$01\ 0100_B$	Invalid
$01\ 0101_B$	22
...	(Only even values are available.)
$11\ 1010_B$	Invalid
$11\ 1011_B$	60
$11\ 1100_B$	Invalid
...	
$11\ 1111_B$	

12.3.3 Clock generator selection registers

12.3.3.1 CKSC_IPLL0S_CTL — PLL0CLK clock control register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 9200$_H$
Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0S STP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.23 CKSC_IPLL0S_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PLL0S STP	PLL0CLK stop control 0: Stop PLL0CLK 1: Activate PLL0CLK (default)

NOTE

PLL0SSTP = 0 does not stop the operation of the PLL0. It just switches off the forwarding of the PLL0 output clock.

12.3.3.2 CKSC_IPLL0S_ACT — PLL0CLK clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 9208$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0S ACT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.24 CKSC_IPLL0S_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PLL0S ACT	PLL0CLK status 0: PLL0CLK is stopped 1: PLL0CLK is active

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.

Its shows the correct value, when the selected clock becomes active.

12.3.3.3 CKSC_IPLL1S_CTL — PLL1CLK clock control register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 9300$_H$
Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1S STP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.25 CKSC_IPLL1S_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PLL1S STP	PLL1CLK stop control 0: Stop PLL1CLK 1: Activate PLL1CLK (default)

NOTES

1. PLL1SSTP = 0 does not stop the operation of the PLL1 clock generator. It just switches off the forwarding of the PLL1 output clock.
2. For D1M2(H) devices:
   The direct PLL1 output clock is used as the clock supply for the DDR2-SDRAM Memory Controller (SDRB). Thus the SDRB clock supply is not stopped by PLL1SSTP = 0.

12.3.3.4 CKSC_IPLL1S_ACT — PLL1CLK clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 9308$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1S ACT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.26 CKSC_IPLL1S_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PLL1S ACT	PLL1CLK status 0: PLL1CLK is stopped 1: PLL1CLK is active

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.

Its shows the correct value, when the selected clock becomes active.

12.3.3.5 CKDV_ICLKJITD_CTL — CLKJIT clock divider register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 53C0$_H$

Initial value: 0000 0006$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CLKJITDIV[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 12.27 CKDV_ICLKJITD_CTL register contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When written, write the initial value.
3 to 0	CLKJIT DIV[3:0]	Clock divider setting for CLKJIT 0000_B: Disabled 0100_B: PLL0CLK /4 0101_B: PLL0CLK /5 0110_B: PLL0CLK /6 (default) 1000_B: PLL0CLK /8 1010_B: PLL0CLK /10 1100_B: PLL0CLK /12 All others: Setting prohibited

12.3.3.6 CKDV_ICLKJITD_STAT — CLKJIT clock divider status register

Access: This register can be read in 32-bit units.

Address: FFF8 53C4$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKJIT ACT	CLKJIT SYNC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.28 CKDV_ICLKJITD_STAT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	CLKJIT ACT	CLKJIT clock output active state 0: clock is inactive 1: clock is active
0	CLKJIT SYNC	CLKJIT clock divider state 0: output clock does not correspond to the current CLKJIT clock divider 1: output clock corresponds to the current CLKJIT clock divider

12.3.3.7 CKSC_ICLKJITS_CTL — CLKJIT source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5400$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKJITS CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12.29 CKSC_ICLKJITS_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	CLKJITS CSID[1:0]	Source clock selection for CLKJIT 01_B: CKDV_ICLKJITD_CTL selection 10_B: High Speed IntOsc f_{RH} (default) All others: Setting prohibited

12.3.3.8 CKSC_ICLKJITS_ACT — CLKJIT source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5408$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKJITS ACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.30 CKSC_ICLKJITS_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	CLKJITS ACT[1:0]	Current active CLKJITS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.3.9 CKDV_ICLKFIXD_CTL — CLKFIX clock divider register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5700$_H$

Initial value: 0000 0006$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CLKFIXDIV[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 12.31 CKSC_ICLKFIXD_CTL register contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When written, write the initial value.
3 to 0	CLKFIX DIV[3:0]	Clock divider setting for CLKFIX 0000_B: Disabled 0110_B: PLLFIXCLK /6 (default) 1000_B: PLLFIXCLK /8 1010_B: PLLFIXCLK /10 1100_B: PLLFIXCLK /12 All others: Setting prohibited

12.3.3.10 CKDV_ICLKFIXD_STAT — CLKFIX clock divider status register

Access: This register can be read in 32-bit units.

Address: FFF8 5704$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKFIX ACT	CLKFIX SYNC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.32 CKDV_ICLKFIXD_STAT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	CLKFIX ACT	CLKFIX clock output active state 0: clock is inactive 1: clock is active
0	CLKFIX SYNC	CLKFIX clock divider state 0: output clock does not correspond to the current CLKFIX clock divider 1: output clock corresponds to the current CLKFIX clock divider

12.3.3.11 CKSC_ICLKFIXS_CTL — CLKFIX source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5740$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKFIXS CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12.33 CKSC_ICLKFIXS_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	CLKFIXS CSID[1:0]	Source clock selection for CLKFIX 01_B: CKDV_ICLKFIXD_CTL selection 10_B: High Speed IntOsc f_{RH} (default) All others: Setting prohibited

12.3.3.12 CKSC_ICLKFIXS_ACT — CLKFIX source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5748$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKFIXS ACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.34 CKSC_ICLKFIXS_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	CLKFIXS ACT[1:0]	Current active CLKFIXS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.3.13 CKSC_IPLLFIXS_CTL — PLLFIXCLK source clock selection register (D1M2(H) only)

The selection of the PLLFIXCLK clock is only available for the D1M2(H) devices.
For all other devices the PLLFIXCLK clock is always the PLL1CLK.

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5000$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLFIXS CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12.35 CKSC_IPLLFIXS_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	PLLFIXS CSID[1:0]	Source clock selection for PLLFIXCLK 01_B: PLL1CLK (default) 10_B: PLL2CLK (D1M2(H) only) All others: Setting prohibited

12.3.3.14 CKSC_IPLLFIXS_ACT — PLLFIXCLK source clock active register (D1M2(H) only)

Access: This register can be read in 32-bit units.

Address: FFF8 5008$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLLFIXS ACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.36 CKSC_IPLLFIXS_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	PLLFIXS ACT[1:0]	Current active PLLFIXS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.3.15 CKSC_ISDRBS_CTL — SDRBCLK clock control register (D1M2(H), D1M1H, D1M1Aonly)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5140$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDRBS STP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.37 CKSC_ISDRBS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	SDRBS STP	SDRBCLK stop control 0: Stop SDRBCLK clock (default) 1: Activate SDRBCLK clock

12.3.3.16 CKSC_ISDRBS_ACT — SDRBCLK clock active register (D1M2(H), D1M1H, D1M1Aonly)

Access: This register can be read in 32-bit units.

Address: FFF8 5148$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDRBD ACT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.38 CKSC_ISDRBS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	SDRBD ACT	SDRBCLK clock status 0: SDRBCLK clock is stopped 1: SDRBCLK clock is operating

12.3.4 CPU Subsystems and bus clock domains selection registers

12.3.4.1 CKSC_ICPUCLKS_CTL — C_ISO_CPUCLK source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 A000$_H$
Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKSCSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.39 CKSC_ICPUCLKS_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	CPUCLKS CSID[2:0]*1	Source clock selection for C_ISO_CPUCLK 001$_B$: EMCLK (default) 011$_B$: PLL0CLK 100$_B$: PLL1CLK All others: Setting prohibited

Note 1. For D1L1 products only:
After changing the value of CPUCLKSCSID[2:0] a waiting time of 1 µsec must pass by inserting an appropriate number of NOP operations before continuing with the application program.

CAUTION

The clock source selected for the C_ISO_CPUCLK clock domain should not be stopped by software.

12.3.4.2 CKSC_ICPUCLKS_ACT — C_ISO_CPUCLK source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 A008$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKSACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.40 CKSC_ICPUCLKS_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	CPUCLKS ACT[2:0]	Current active CPUCLKS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.4.3 CKSC_ICPUCLKD_CTL — C_ISO_CPUCLK clock divider register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 A100$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKDCSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.41 CKSC_ICPUCLKD_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	CPUCLKD CSID[2:0]	Clock divider setting for C_ISO_CPUCLK 001$_B$: CKSC_ICPUCLKS_CTL selection /1 (default) 010$_B$: CKSC_ICPUCLKS_CTL selection /2 011$_B$: CKSC_ICPUCLKS_CTL selection /4 100$_B$: CKSC_ICPUCLKS_CTL selection /8 101$_B$: CKSC_ICPUCLKS_CTL selection /3 All others: Setting prohibited

12.3.4.4 CKSC_ICPUCLKD_ACT — C_ISO_CPUCLK clock divider active register

Access: This register can be read in 32-bit units.

Address: FFF8 A108$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUCLKDACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.42 CKSC_ICPUCLKD_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	CPUCLKD ACT[2:0]	Current active CPUCLKD clock divider

12.3.4.5 CKSC_IXCCLKS_CTL — C_ISO_XCCLK clock control register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5180$_H$
Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	XCCLKSSTP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.43 CKSC_IXCCLKS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	XCCLKSSTP	C_ISO_XCCLK stop control 0: Stop C_ISO_XCCLK (default) 1: Activate C_ISO_XCCLK

CAUTION

Before activating C_ISO_XCCLK by XCCLKSSTP = 1,

- **the XC0 cross-connect software reset must be released (MRSTC.XC0RES = 1) and**
- **the Ethernet AVB MAC (ETNB0) software reset must be released (MRSTC.ETNB0RES = 1) and**
- **the Ethernet AVB MAC (ETNB0) PBUS clock ETNBPCLK must be activated by CKSC_IPCETNBS_CTL.IPCETNBSSTP = 1.**

12.3.4.6 CKSC_IXCCLKS_ACT — C_ISO_XCCLK clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5188$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	XCCLKSACT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.44 CKSC_IXCCLKS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	XCCLKS ACT	C_ISO_XCCLK status 0: C_ISO_XCCLK is stopped (default) 1: C_ISO_XCCLK is active

12.3.4.7 CKSC_IPCETNBS_CTL — ETNBPCLK clock control register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5280$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPC ETNBS STP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.45 CKSC_IPCETNBS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	IPCETNBS STP	ETNBPCLK stop control 0: Stop ETNBPCLK clock (default) 1: Activate ETNBPCLK clock

12.3.4.8 CKSC_IPCETNBS_ACT — ETNBPCLK clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5288$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCETNBSACT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.46 CKSC_IPCETNBS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PCETNBSACT	ETNBPCLK status 0: ETNBPCLK is stopped 1: ETNBPCLK is active

12.3.4.9 CKSC_IXCETNBS_CTL — ETNBXCCLK clock control register (D1M2(H), D1M1(H), D1M1-V2, D1M1A only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 51C0$_H$
Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IXCETNBSSTP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.47 CKSC_IXCETNBS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	IXCETNBS STP	ETNBXCCLK stop control 0: Stop ETNBXCCLK clock (default) 1: Activate ETNBXCCLK clock

12.3.4.10 CKSC_IXCETNBS_ACT — ETNBXCCLK clock active register (D1M2(H), D1M1(H), D1M1-V2, D1M1A only)

Access: This register can be read in 32-bit units.

Address: FFF8 51C8$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	XC ETNBS ACT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.48 CKSC_IXCETNBS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	XCETNBS ACT	ETNBXCCLK status 0: ETNBXCCLK is stopped 1: ETNBXCCLK is active

12.3.4.11 CKSC_IPCMLBBS_CTL — MLBBPCLK clock control register (D1M2H only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 52C0$_H$
Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPC MLBBS STP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.49 CKSC_IPCMLBBS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	IPCMLBBS STP	MLBBPCLK stop control 0: Stop MLBBPCLK clock (default) 1: Activate MLBBPCLK clock

12.3.4.12 CKSC_IPCMLBBS_ACT — MLBBPCLK clock active register (D1M2H only)

Access: This register can be read in 32-bit units.

Address: FFF8 52C8$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCMLBBSACT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.50 CKSC_IPCMLBBS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PCMLBBS ACT	MLBBPCLK status 0: MLBBPCLK is stopped 1: MLBBPCLK is active

12.3.4.13 CKSC_IXCMLBBS_CTL — MLBBXCCLK clock control register (D1M2H only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5200$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IXC MLBBS STP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.51 CKSC_IXCMLBBS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	IXCMLBBS STP	MLBBXCCLK stop control 0: Stop MLBBXCCLK clock (default) 1: Activate MLBBXCCLK

12.3.4.14 CKSC_IXCMLBBS_ACT — MLBBXCCLK clock active register (D1M2H only))

Access: This register can be read in 32-bit units.

Address: FFF8 5208$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	XC MLBBS ACT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.52 CKSC_IXCMLBBS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	XCMLBBS ACT	MLBBXCCLK status 0: MLBBXCCLK is stopped 1: MLBBXCCLK is active

12.3.4.15 CKSC_IPCRSCANS_CTL — RSCANPCLK clock control register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5240$_H$
Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCRSCANSSTP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.53 CKSC_IPCRSCANS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PCRSCANSSTP	RSCANPCLK stop control 0: Stop RSCANPCLK clock 1: Activate RSCANPCLK clock (default)

12.3.4.16 CKSC_IPCRSCANS_ACT — RSCANPCLK clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5248$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PCRSCANSACT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.54 CKSC_IPCRSCANS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PCRSCANSACT	PCRSCANCLK status 0: RSCANPCLK is stopped 1: RSCANPCLK is active

12.3.4.17 APB_CLK_RATIO — PBUS clock ration selection register (D1M1(H) only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDPWRGD.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 F510$_H$
Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	APB_CLK_RATIO
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.55 APB_CLK_RATIO register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	APB_CLK_RATIO	PBUS clock ratio selection 0: C_ISO_PCLK = C_ISO_CPUCLK / 2 1: C_ISO_PCLK = C_ISO_CPUCLK / 4 (default)

NOTE

- APB_CLK_RATIO only can change on High Speed IntOsc operation of CPU (except PLL operation of CPU).
- If software change the APB_CLK_RATIO settings, software should confirm the reflected correct setting by read access. After then, software can go forward next steps.
- APB_CLK_RATIO only can change from 1 to 0. The change from 0 to 1 is not permitted.

12.3.5 Always-On-Area clock domain selection registers

12.3.5.1 CKSC_AAWOTS_CTL — C_AWO_AWOT source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 2100_H
Initial value: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AWOTSCSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.56 CKSC_AAWOTS_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	AWOTS CSID[2:0]	Source clock selection for C_AWO_AWOT 000_B: Disabled 001_B: High Speed IntOsc f_{RH} (default) 010_B: MainOsc f_X 011_B: Low Speed IntOsc f_{RL} All others: Setting prohibited

12.3.5.2 CKSC_AAWOTS_ACT — C_AWO_AWOT source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 2108$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AWOTSACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.57 CKSC_AAWOTS_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	AWOTS ACT[2:0]	Current active AWOTS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.5.3 CKSC_AAWOTD_CTL — C_AWO_AWOT clock divider register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 2200_H

Initial value: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AWOTDCSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.58 CKSC_AAWOTD_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	AWOTD CSID[2:0]	Clock divider setting for C_AWO_AWOT 001_B: CKSC_AAWOTS_CTL selection /1 (default) 010_B: CKSC_AAWOTS_CTL selection /2 011_B: CKSC_AAWOTS_CTL selection /4 100_B: CKSC_AAWOTS_CTL selection /8 101_B: CKSC_AAWOTS_CTL selection /16 All others: Setting prohibited

12.3.5.4 CKSC_AAWOTD_ACT — C_AWO_AWOT clock divider active register

Access: This register can be read in 32-bit units.

Address: FFF8 2208$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	AWOTD ACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.59 CKSC_AAWOTD_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	AWOTD ACT[2:0]	Current active AWOTD clock divider

12.3.5.5 CKSC_AAWOTD_STPM — C_AWO_AWOT stop mask register

Access: This register can be read/written in 32-bit units.

Address: FFF8 2218$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AWOTD STP MSK
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

The initial value “1” of bit 1 must not be changed.

Table 12.60 CKSC_AAWOTD_STPM register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	AWOTD STPMSK	0: Clock domain C_AWO_AWOT is stopped in DEEPSTOP mode Clock domain C_AWO_AWOT is stopped by disabling the C_AWO_AWOT source clock selection register (CKSC_AAWOTS_CTL = 0000 0000$_H$). 1: Clock domain C_AWO_AWOT is not stopped in DEEPSTOP mode Clock domain C_AWO_AWOT is not stopped by disabling the C_AWO_AWOT source clock selection register (CKSC_AAWOTS_CTL = 0000 0000$_H$).

12.3.5.6 CKSC_AWDTA0D_CTL — C_AWO_WDTA0 clock divider register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 2000_H

Initial value: 0000 0001_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTA0DCSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.61 CKSC_AWDTA0D_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	WDTA0D CSID[2:0]	Clock divider setting for C_AWO_WDTA0 001_B: Low Speed IntOsc f_{RL} / 128 (default) 010_B: Low Speed IntOsc f_{RL} / 1 011_B: Low Speed IntOsc f_{RL} / 256 100_B: Low Speed IntOsc f_{RL} / 512 101_B: Low Speed IntOsc f_{RL} / 1024 110_B: Low Speed IntOsc f_{RL} / 2048 All others: Setting prohibited

CAUTION

Confirm that CKSC_AWDTA0D_CTL is CKSC_AWDTA0D_ACT before setting the CKSC_AWDTA0D_CTL register.

12.3.5.7 CKSC_AWDTA0D_ACT — C_AWO_WDTA0 clock divider active register

Access: This register can be read in 32-bit units.

Address: FFF8 2008$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	WDTA0DACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.62 CKSC_AWDTA0D_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	WDTA0D ACT[2:0]	Current active WDTA0D clock divider

12.3.5.8 CKSC_ARTCAS_CTL — C_AWO_RTCA source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 2300_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCASCSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.63 CKSC_ARTCAS_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	RTCAS CSID[2:0]	Source clock selection for C_AWO_RTCA 000_B: Disable (default) 001_B: SubOsc f_{TX} 010_B: MainOsc f_X 011_B: Low Speed IntOsc f_{RL} 100_B: High Speed IntOsc f_{RH} All others: Setting prohibited

NOTE

To avoid supplying a clock signal at a frequency of higher than 4 MHz to the C_AWO_RTCA clock domain due to the CKSC_ARTCAS_CTL register setting, check that CKSC_ARTCAD_ACT = 0000 0000_H (disabled) when the setting of CKSC_ARTCAS_CTL is 010_B (MainOsc f_X) or 100_B (High Speed IntOsc f_{RH}).

CAUTION

Always make sure that the source clock is operating before assigning it (writing to this register). The source clock assignment cannot be changed afterwards if the assigned source clock is not running.

This register is only reset by POC0RES, ISOPWRES and $\overline{\text{DBRES}}$.

12.3.5.9 CKSC_ARTCAS_ACT — C_AWO_RTCA source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 2308$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCASACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.64 CKSC_ARTCAS_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	RTCAS ACT[2:0]	Current active RTCAS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.5.10 CKSC_ARTCAD_CTL — C_AWO_RTCA clock divider register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 2400$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCADCSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.65 CKSC_ARTCAD_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	RTCAD CSID[2:0]	Clock divider setting for C_AWO_RTCA 000_B: Disabled (default) 001_B: CKSC_ARTCAS_CTL selection /1 010_B: CKSC_ARTCAS_CTL selection /2 011_B: CKSC_ARTCAS_CTL selection /4 100_B: CKSC_ARTCAS_CTL selection /8 101_B: CKSC_ARTCAS_CTL selection /16 All others: Setting prohibited

12.3.5.11 CKSC_ARTCAD_ACT — C_AWO_RTCA clock divider active register

Access: This register can be read in 32-bit units.

Address: FFF8 2408$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCADACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.66 CKSC_ARTCAD_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	RTCAD ACT[2:0]	Current active RTCAD clock divider

12.3.5.12 CKSC_ARTCAD_STPM — C_AWO_RTCA stop mask register

Access: This register can be read/written in 32-bit units.

Address: FFF8 2418$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCAD STP MSK
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

The initial value “1” of bit 1 must not be changed.

Table 12.67 CKSC_ARTCAD_STPM register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	RTCAD STPMSK	0: Clock domain C_AWO_RTCA is stopped in DEEPSTOP mode Clock domain C_AWO_RTCA is stopped by disabling the C_AWO_RTCA source clock selection register (CKSC_ARTCAS_CTL = 0000 0000$_H$). 1: Clock domain C_AWO_RTCA is not stopped in DEEPSTOP mode Clock domain C_AWO_RTCA is not stopped by disabling the C_AWO_RTCA source clock selection register (CKSC_ARTCAS_CTL = 0000 0000$_H$).

12.3.5.13 CKSC_AFOUTS_CTL — C_AWO_FOUT source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 2700_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUTSCSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.68 CKSC_AFOUTS_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	FOUTS CSID[2:0]	Source clock selection for C_AWO_FOUT 000_B: Disabled (default) 001_B: MainOsc f_X 010_B: High Speed IntOsc f_{RH} 011_B: Low Speed IntOsc f_{RL} 100_B: SubOsc f_{TX} 101_B: PLL0CLK/4 110_B: PLL1CLK/4 111_B: PLL2CLK/4 (D1M2(H) only)

CAUTION

Before setting FOUTSCSID[2:0] to 101_B or 110_B make sure that PLL0 and PLL1 are operating. Otherwise FOUT will not work.

12.3.5.14 CKSC_AFOUTS_ACT — C_AWO_FOUT source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 2708$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUTSACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.69 CKSC_AFOUTS_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	FOUTS ACT[2:0]	Current active FOUTS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.5.15 CKSC_AFOUTS_STPM — C_AWO_FOUT stop mask register

Access: This register can be read/written in 32-bit units.

Address: FFF8 2718$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUTSSTPMSK
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

The initial value “1” of bit 1 must not be changed.

Table 12.70 CKSC_AFOUTS_STPM register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	FOUTS STPMSK	0: Clock domain C_AWO_FOUT is stopped in DEEPSTOP mode 1: Clock domain C_AWO_FOUT is not stopped in DEEPSTOP mode

12.3.6 Isolated-Area clock domain selection registers

12.3.6.1 CKSC_IMLBBS_CTL — C_ISO_MLBB source clock selection register (D1M2H only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5300$_H$
Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MLBBS CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12.71 CKSC_IMLBBS_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	MLBBS CSID[1:0]	Source clock selection for C_ISO_MLBB 00_B: Disabled (default) 01_B: PLL0CLK /2 10_B: PLL0CLK /4 11_B: Setting prohibited

12.3.6.2 CKSC_IMLBBS_ACT — C_ISO_MLBB source clock active register (D1M2H only)

Access: This register can be read in 32-bit units.

Address: FFF8 5308$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MLBBSACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.72 CKSC_IMLBBS_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	MLBBS ACT[1:0]	Current active MLBBS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.6.3 CKSC_ISFMAS_CTL — C_ISO_SFMA source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5340$_H$
Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SFMAS CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12.73 CKSC_ISFMAS_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	SFMAS CSID[1:0]	Source clock selection for C_ISO_SFMA 01_B: PLL0CLK 10_B: PLL1CLK (default) All others: Setting prohibited

12.3.6.4 CKSC_ISFMAS_ACT — C_ISO_SFMA source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5348$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SFMAS ACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.74 CKSC_ISFMAS_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	SFMAS ACT[1:0]	Current active SFMAS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.6.5 CKSC_ISFMAD_CTL — C_ISO_SFMA clock divider register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5380$_H$
Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SFMAD CSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.75 CKSC_ISFMAD_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	SFMAD CSID[2:0]	D1L2(H), D1M1(H), D1M1-V2, D1M1A, D1M2(H) Clock divider setting for C_ISO_SFMA 000_B: Disabled (default) 001_B: CKSC_ISFMAS_CTL selection /2 010_B: CKSC_ISFMAS_CTL selection /3 011_B: CKSC_ISFMAS_CTL selection /4 100_B: CKSC_ISFMAS_CTL selection /6 All others: Setting prohibited
		D1L1 Clock divider setting for C_ISO_SFMA 000_B: Disabled (default) 100_B: CKSC_ISFMAS_CTL selection /6 All others: Setting prohibited

12.3.6.6 CKSC_ISFMAD_ACT — C_ISO_SFMA clock divider active register

Access: This register can be read in 32-bit units.

Address: FFF8 5388$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	SFMADACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.76 CKSC_ISFMAD_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	SFMAD ACT[2:0]	Current active SFMAD clock divider

12.3.6.7 CKSC_IRSCAND_CTL — C_ISO_RSCAN source clock divider register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5780$_H$

Initial value: 0000 0004$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RSCANDCSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.77 CKSC_IRSCAND_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	RSCAND CSID[2:0]	Source clock divider for C_ISO_RSCAN 000_B: Disabled 001_B: PLLFIXCLK /30 010_B: PLLFIXCLK /40 011_B: PLLFIXCLK /60 100_B: PLLFIXCLK /12 (default) 101_B: PLLFIXCLK /24 110_B: PLLFIXCLK /10 (D1M1A only) All others: Setting prohibited

12.3.6.8 CKSC_IRSCAND_ACT — C_ISO_RSCAN source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5788$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RSCANDACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.78 CKSC_IRSCAND_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	RSCAND ACT[2:0]	Current active RSCAND source clock selection

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.

Its shows the correct value, when the selected clock becomes active.

12.3.6.9 CKSC_IRSCANXINS_CTL — C_ISO_RSCANXIN clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5800$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSCANXINSCSID0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.79 CKSC_IRSCANXINS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	RSCANXINSCSID0	Source clock selection for C_ISO_RSCANXIN 0: Disabled 1: MainOsc f_X (default).

12.3.6.10 CKSC_IRSCANXINS_ACT — C_ISO_RSCANXIN source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5808$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IRSCANXINSACT0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.80 CKSC_IRSCANXINS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	IRSCANXINSACT0	Current active RSCANXINS source clock selection 0: selected clock inactive 1: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.6.11 CKDV_ISSIFD_CTL — C_ISO_SSIF clock divider register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 57C0$_H$

Initial value: 0000 000A$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SSIFD CSID[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.81 CKSC_ISSIFD_CTL register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When written, write the initial value.
7 to 0	SSIFD CSID[7:0]	Clock divider setting for C_ISO_SSIF 0: Disabled 10: PLLFIXCLK /10 (default) ... 255: PLLFIXCLK /255 All others: Setting prohibited

12.3.6.12 CKDV_ISSIFD_STAT — C_ISO_SSIF clock divider active register

Access: This register can be read in 32-bit units.

Address: FFF8 57C4$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIF ACT	SSIF SYNC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.82 CKDV_ISSIFD_STAT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	SSIF ACT	SSIF clock output active state 0: clock is inactive 1: clock is active
0	SSIF SYNC	SSIF clock divider state 0: output clock does not correspond to the current SSIFD clock divider 1: output clock corresponds to the current SSIFD clock divider

12.3.6.13 CKSC_ITAUB01S_CTL — C_ISO_TAUB01 source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5540$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TAUB01S CSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.83 CKSC_ITAUB01S_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	TAUB01S CSID[2:0]	Source clock selection for C_ISO_TAUB01 000_B: Disabled 001_B: CLKFIX /1 010_B: CLKFIX /8 (default) 011_B: CLKJIT /1 100_B: CLKJIT /8 All others: Setting prohibited

12.3.6.14 CKSC_ITAUB01S_ACT — C_ISO_TAUB01 source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5548$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TAUB01S ACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.84 CKSC_ITAUB01S_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	TAUB01S ACT[2:0]	Current active TAUB01S source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.6.15 CKSC_ITAUB2S_CTL — C_ISO_TAUB2 source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5580$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TAUB2SCSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.85 CKSC_ITAUB2S_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	TAUB2S CSID[2:0]	Source clock selection for C_ISO_TAUB2 000_B: Disabled 001_B: CLKFIX /1 010_B: CLKFIX /8 (default) 011_B: CLKJIT /1 100_B: CLKJIT /8 101_B: MainOsc f_X All others: Setting prohibited

12.3.6.16 CKSC_ITAUB2S_ACT — C_ISO_TAUB2 source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5588$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TAUB2SACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.86 CKSC_ITAUB2S_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	TAUB2S ACT[2:0]	Current active TAUB2S source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.6.17 CKSC_ITAUJS_CTL — C_ISO_TAUJ source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 55C0$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TAUJS CSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 12.87 CKSC_ITAUJS_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	TAUJS CSID[2:0]	Source clock selection for C_ISO_TAUJ 000_B: Disabled 001_B: CLKFIX /1 010_B: CLKFIX /8 (default) 011_B: CLKJIT /1 100_B: CLKJIT /8 101_B: MainOsc f_X 110_B: C_ISO_CPUCLK /4 All others: Setting prohibited

12.3.6.18 CKSC_ITAUJS_ACT — C_ISO_TAUJ source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 55C8$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TAUJSACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.88 CKSC_ITAUJS_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	TAUJS ACT[2:0]	Current active ITAUJS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.6.19 CKSC_IOSTMS_CTL — C_ISO_OSTM source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5600$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OSTMS CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12.89 CKSC_IOSTMS_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	OSTMS CSID[1:0]	Source clock selection for C_ISO_OSTM 00_B: Disabled 01_B: CLKJIT 10_B: C_ISO_CPUCLK /4 (default) 11_B: MainOsc f_X

12.3.6.20 CKSC_IOSTMS_ACT — C_ISO_OSTM source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5608$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OSTMSACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.90 CKSC_IOSTMS_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	OSTMS ACT[1:0]	Current active IOSTMS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.6.21 CKSC_ILCBIS_CTL — C_ISO_LCBI source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 54C0$_H$
Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LCBIS CSID0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.91 CKSC_ILCBIS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	LCBIS CSID0	Source clock selection for C_ISO_LCBI 0: Disabled (default) 1: CLKJIT

12.3.6.22 CKSC_ILCBIS_ACT — C_ISO_LCBI source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 54C8$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LCBIS ACT0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.92 CKSC_ILCBIS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	LCBIS ACT0	Current active LCBIS source clock selection 0: selected clock inactive 1: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.6.23 CKSC_IADCED_CTL — C_ISO_ADCE clock divider register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5480$_H$
Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCED CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 12.93 CKSC_IADCED_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1, 0	ADCED CSID[1:0]	Clock divider setting for C_ISO_ADCE 01_B: CLKJIT /2 (default) 10_B: CLKJIT /4 All others: Setting prohibited

CAUTIONS

1. **The A/D Converter (ADCE) requires a minimum operating frequency of 8 MHz. Make sure that C_ISO_ADCE is not below 8 MHz. Refer to the Data Sheet for details.**
2. **The CKSC_IADCED_CTL register must be set so that the relationship between frequency of the C_ISO_PCLK and C_ISO_ADCE clocks is retained within the range of "C_ISO_PCLK/C_ISO_ADCE = 1 to 2.4".**

12.3.6.24 CKSC_IADCED_ACT — C_ISO_ADCE clock divider active register

Access: This register can be read in 32-bit units.

Address: FFF8 5488$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCEDACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.94 CKSC_IADCED_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1, 0	ADCED ACT[1:0]	Current active ADCED clock divider

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.

Its shows the correct value, when the selected clock becomes active.

12.3.6.25 CKSC_IISMS_CTL — C_ISO_ISM source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5440$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ISMS CSID0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 12.95 CKSC_IISMS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	ISMS CSID0	Source clock selection for C_ISO_ISM 0: Disabled (default) 1: CLKJIT

12.3.6.26 CKSC_IISMS_ACT — C_ISO_ISM source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5448$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ISMS ACT0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.96 CKSC_IISMS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	ISMS ACT0	Current active ISMS source clock selection 0: selected clock inactive 1: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.3.6.27 CKSC_IRLINS_CTL — C_ISO_RLIN source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5500$_H$

Initial value:
- D1M1(H), D1M1-V2, D1M1A: 0000 0003$_H$
- All other devices: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RLINS CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Note 1. In D1M1(H), D1M1-V2, D1M1A, initial value of this bit is 1.

Table 12.97 CKSC_IRLINS_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1, 0	RLINS CSID[1:0]	Source clock selection for C_ISO_RLIN • D1M1(H), D1M1-V2, D1M1A: 01_B: PLLFIXCLK /10 10_B: CLKJIT /1 11_B: PLLFIXCLK /20 (default) All others: Setting prohibited • All other devices: 01_B: PLLFIXCLK /10 10_B: CLKJIT /1 (default) All others: Setting prohibited

12.3.6.28 CKSC_IRLINS_ACT — C_ISO_RLIN source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5508$_H$

Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RLINS ACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1*1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. In D1M1(H), D1M1-V2, D1M1A, initial value of this bit is 0.

Table 12.98 CKSC_IRLINS_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1, 0	RLINS ACT[1:0]	Current active RLINS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

12.4 Clock Controller set-up

The following procedure shows an initial set-up of the Clock Controller to activate the CPU Subsystem and the cross-connects.

(1) Configure the MainOsc
- Select the MainOsc frequency via the MOSCC register.
- Select the correct MainOsc stabilization time via the MOSCST register.
- Start the MainOsc by MOSCE.MOSCENTRG = 1.
- Wait until MOSCS.MOSCCLKACT = 1 indicates MainOsc active.

(2) Configure the PLLk clock generators
- Define the PLLk parameters via the PLLkC register.
- Start the PLLs by PLLkE.PLLkENTRG = 1.
- Wait until PLLkS.PLLkCLKACT = 1 indicates PLLk active.
- Enable the PLLkCLK clock by CKSC_IPLLkS_CTL.PLLkSSTP = 1.
- Wait until CKSC_IPLLkS_ACT = 1 indicates PLLkCLK active.

(3) Configure the SubOsc
- Select correct SubOsc stabilization time via SOSCST register.
- Start the SubOsc by SOSCE.SOSCENTRG = 1.
- Wait until SOSCS.SOSCCLKACT = 1 indicates SubOsc active.

(4) Activate the C_ISO_CPUCLK clock domain to supply the CPU Subsystem with its operation clock
- Select the C_ISO_CPUCLK clock divider via the CKSC_ICPUCLKD_CTL register.
- Wait until CKSC_ICPUCLKD_ACT = 1 indicates the selected divider.
- Select the C_ISO_CPUCLK clock source via the CKSC_ICPUCLKS_CTL register.
- Wait until CKSC_ICPUCLKS_ACT = 1 indicates C_ISO_CPUCLK active.

NOTE

In order to prevent supplying unintentional clocks to the CPU Subsystem, it is necessary to set CKSC_CPUCLKS_CTL after selecting the divider by CKSC_ICPUCLKD_CTL.

(5) Activate C_ISO_XCCLK clock domain to supply the cross-connects with its operation clocks
- Release the Ethernet AVB Mac (ETNB0) reset by MRSTC.ETNB0RES = 1.
- Enable the ETNBPCLK clock by CKSC_IPCETNBS_CTL.IPCETNBSSTP = 1.
- Wait until CKSC_IPCETNBS_ACT = 1 indicates ETNBPCLK active.
- Release the XC0 cross-connect reset by MRSTC.XC0RES = 1.
- Enable the C_ISO_XCCLK clock by CKSC_IXCCLKS_CTL.XCCLKSSTP = 1.
- Wait until CKSC_IXCCLKS_ACT = 1 indicates C_ISO_XCCLK active.

NOTE

For stopping the C_ISO_XCCLK clock, follow the procedure shown below.
- Make sure that ETNBPCLK is active, i.e. CKSC_IPCETNBS_ACT = 1.

- Make sure that XC0 cross-connect reset is inactive, i.e. MRSTC.XC0RES = 1.
- Make sure that the Ethernet AVB MAC (ETNB0) reset is inactive, i.e. MRSTC.ETNB0RES = 1.
- Stop the C_ISO_XCCLK clock by CKSC_IXCCLKS_CTL.XCCLKSSTP = 0.
- Wait until CKSC_IXCCLKS_ACT = 0 indicates C_ISO_XCCLK inactive.

12.5 Clock Selection

The base clocks, generated by the clock generators, are distributed to the functional modules by several clock domains. The tables below show the various clock domains, their possible clock sources, clock dividers and control options.
Note that various functional modules provide additional internal prescalers and dividers.

NOTE

The default value of the clock selector and divider registers are denoted in *italics*.

CAUTIONS

1. **When changing the source clock by a clock selector, the selected clock source before changing the selector must not be stopped until the new source was selected. Otherwise the clock selector will not be able to perform the new assignment.**
2. **The register access clock for each module must be supplied when the registers of the module are accessed.**
   The module can't receive request from CPU if register access clock for modules is not supplied.

12.5.1 Base clocks

The following table shows the selection and control options for the generation of the clock controller base clocks.

Refer also to Section 12.1.2, Clock Controllers Block Diagrams for figures of the clock controller.

Table 12.99 Clock controller base clocks selection and control (1/2)

	Clock selection		Clock divider	
Base clock	**Register**	**Selections**	**Register**	**Divisors**
PLL0CLK	CKSC_IPLL0S_CTL	Stop	–	–
		PLL0CLK		
PLL1CLK	CKSC_IPLL1S_CTL	Stop	–	–
		PLL1CLK		
CLKJIT	–	PLL0CLK	CKDV_ICLKJITD_CTL	Disable, 4, 5, *6*, 8, 10, 12
	CKSC_ICLKJITS_CTL	CKDV_ICLKJITD_CTL selection	–	–
		HS IntOsc f_{RH}		
CLKFIX	–	PLLFIXCLK	CKDV_ICLKFIXD_CTL	Disable, *6*, 8, 10, 12
	CKSC_ICLKFIXS_CTL	CKDV_ICLKFIXD_CTL selection	–	–
		HS IntOsc f_{RH}		

Table 12.99 Clock controller base clocks selection and control (2/2)

Base clock	Clock selection		Clock divider	
	Register	Selections	Register	Divisors
PLLFIXCLK	CKSC_IPLLFIXS_CTL (D1M2(H) only)*2	*PLL1CLK*	–	–
		PLL2CLK		
SDRBCLK (D1M2(H), D1M1H and D1M1A only)	CKSC_ISDRBS_CTL	*Stop*	–	–
		SDRBCLK		

Note 1. The default value of the clock selector and divider registers are denoted in *italics*.
Note 2. For all other devices: PLLFIXCLK = PLL1CLK.

Table 12.100 Base clocks maximum frequencies

Base clock	Maximum frequency								
	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M1-V2	D1M1A	D1M2	D1M2H
PLL0CLK	480 MHz					960 MHz		480 MHz	
PLL1CLK	480 MHz							960 MHz	
CLKJIT	80 MHz								
CLKFIX	80 MHz								
PLLFIXCLK	480 MHz								
SDRBCLK	–				200 MHz	–	240 MHz	480 MHz	

12.5.2 CPU and buses subsystems clocks

The following table shows the selection and control options for the generation of the CPU and buses subsystem clocks.

Refer also to Section 12.1.2.6, CPU Subsystem and bus clock domains for figures of the clock controller.

Table 12.101 CPU Subsystems and bus clock domains selection (1/2)

Clock domain	Clock selection		Clock divider		Module
	Register	Selections	Register	Divisors	
C_ISO_CPUCLK	CKSC_ICPUCLKS_CTL	*EMCLK*	CKSC_ICPUCLKD_CTL	*1*, 2, 4, 8, 3	CPU Subsystem
		PLL0CLK			
		PLL1CLK			
C_ISO_XCCLK	CKSC_IXCCLKS_CTL	*Stop*	–	–	Cross-connect systems
		C_ISO_XCCLK			
ETNBPCLK	CKSC_IPCETNBS_CTL	*Stop*	–	–	ETNB0
		ETNBPCLK			
ETNBXCCLK (D1M2(H), D1M1(H), D1M1-V2 and D1M1A)	CKSC_IXCETNBS_CTL	*Stop*	–	–	
		ETNBXCCLK			

Table 12.101 CPU Subsystems and bus clock domains selection (2/2)

Clock domain	Clock selection		Clock divider		Module
	Register	Selections	Register	Divisors	
MLBBPCLK (D1M2H)	CKSC_IPCMLBBS_CTL	*Stop*	–	–	MLBB0
		MLBBPCLK			
MLBBXCCLK (D1M2H)	CKSC_IXCMLBBS_CTL	*Stop*	–	–	
		MLBBXCCLK			
RSCANPCLK	CKSC_IPCRSCANS_CTL	Stop	–	–	RSCAN0
		RSCANPCLK			

Note 1. The default value of the clock selector and divider registers are denoted in *italics*.

Table 12.102 CPU and buses subsystems clocks maximum frequencies

Base clock	Maximum frequency								
	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M1-V2	D1M1A	D1M2	D1M2H
C_ISO_CPUCLK	120 MHz			160 MHz	200 MHz	160 MHz	240 MHz		
C_ISO_XCCLK	60 MHz			80 MHz	100 MHz	80 MHz	120 MHz		
C_ISO_PCLK	60 MHz			80 MHz	50 MHz	80 MHz	60 MHz		
ETNBPCLK	–			80 MHz	50 MHz	80 MHz	60 MHz		
ETNBXCCLK	–			80 MHz	100 MHz	80 MHz	120 MHz		
MLBBPCLK	–								60 MHz
MLBBXCCLK	–								120 MHz
RSCANPCLK	80 MHz								

12.5.3 Always-On-Area clock domains

The following table shows the selection and control options for the generation of the Always-On-Area clock domains.

AWO clock domains stop

Each AWO clock domain except for C_AWO_WDTA0 is equipped with a stop mask register, that allows to select either stop or continuation of the clock domain in DEEPSTOP mode.

Table 12.103 Always-On-Area clock domains selections (1/2)

Clock domain	Clock selection		Clock divider		Module
	Register	Selections	Register	Divisors	
C_AWO_AWOT	CKSC_AAWOTS_CTL	Disabled	CKSC_AAWOTD_CTL	*1*, 2, 4, 8, 16	AWOT0
		HS IntOsc f_{RH}			
		MainOsc f_X			
		LS IntOsc f_{RL}			
C_AWO_WDTA0	–	LS IntOsc f_{RL}	CKSC_AWDTA0D_CTL	*128*, 1, 256, 512, 1024, 2048	WDTA0

Table 12.103 Always-On-Area clock domains selections (2/2)

Clock domain	Clock selection		Clock divider		Module
	Register	Selections	Register	Divisors	
C_AWO_RTCA	CKSC_ARTCAS_CTL	*Disabled*	CKSC_ARTCAD_CTL	*Disabled*, 1, 2, 4, 8, 16	RTCA0
		LS IntOsc f_{RL}			
		HS IntOsc f_{RH}			
		MainOsc f_X			
		SubOsc f_{TX}			
C_AWO_FOUT	CKSC_AFOUTS_CTL	*Disabled*	–	–	FOUT
		MainOsc f_X			
		HS IntOsc f_{RH}			
		LS IntOsc f_{RL}			
		SubOsc f_{TX}			
		PLL0CLK/4			
		PLL1CLK/4			
		PLL2CLK/4 (D1M2(H) only)			

Note 1. The default value of the clock selector and divider registers are denoted in *italics*.

Table 12.104 Always-On-Area clock domain clocks maximum frequencies

Base clock	Maximum frequency								
	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M1-V2	D1M1A	D1M2	D1M2H
C_AWO_AWOT	8 MHz								
C_AWO_WDTA0	240 kHz								
C_AWO_RTCA	4 MHz								
C_AWO_FOUT	20 MHz								

12.5.4 Isolated-Area clock domains

The following table shows the selection and control options for the generation of the Isolated-Area clock domains.

Table 12.105 Isolated-Area clock domains selections (1/2)

Clock domain	Clock selection		Clock divider		Module
	Register	Selections	Register	Divisors	
C_ISO_MLBB	CKSC_IMLBBS_CTL	*Disabled*	–	–	MLBB0
		PLL0CLK/2			
		PLL0CLK/4			
C_ISO_SFMA	CKSC_ISFMAS_CTL	PLL0CLK	CKSC_ISFMAD_CTL	*Disabled*, 2, 3, 4, 6*3	SFMA0 SFMA1 SFMA2
		PLL1CLK			

Table 12.105 Isolated-Area clock domains selections (2/2)

Clock domain	Clock selection		Clock divider		Module
	Register	Selections	Register	Divisors	
C_ISO_RSCAN	–	PLLFIXCLK	CKSC_IRSCAND_CTL	Disabled 30, 40, 60, *12*, 24	RSCAN0
C_ISO_RSCANXIN	CKSC_IRSCANXINS_CTL	Disabled	–	–	
		MainOsc f_X			
C_ISO_SSIF	–	PLLFIXCLK	CKDV_ISSIFD_CTL	Disabled *10*, 11, ..., 255	SSIF0 SSIF1
C_ISO_TAUB01	CKSC_ITAUB01S_CTL	Disabled	–	–	TAUB0 TAUB1
		CLKFIX/1			
		CLKFIX/8			
		CLKJIT/1			
		CLKJIT/8			
C_ISO_TAUB2	CKSC_ITAUB2S_CTL	Disabled	–	–	TAUB2
		CLKFIX/1			
		CLKFIX/8			
		CLKJIT/1			
		CLKJIT/8			
		MainOsc f_X			
C_ISO_TAUJ	CKSC_ITAUJS_CTL	Disabled	–	–	TAUJ0
		CLKFIX/1			
		CLKFIX/8			
		CLKJIT/1			
		CLKJIT/8			
		MainOsc f_X			
		C_ISO_CPUCLK /4			
C_ISO_OSTM	CKSC_IOSTMS_CTL	Disabled	–	–	OSTM0 OSTM1
		CLKJIT			
		C_ISO_CPUCLK /4			
		MainOsc f_X			
C_ISO_LCBI	CKSC_ILCBIS_CTL	*Disabled*	–	–	LCBI0
		CLKJIT			
C_ISO_ADCE	–	CLKJIT	CKSC_IADCED_CTL	*2*, 4	ADCE0
C_ISO_ISM	CKSC_IISMS_CTL	*Disabled*	–	–	ISM0
		CLKJIT			
C_ISO_RLIN	CKSC_IRLINS_CTL	PLLFIXCLK /10	–	–	RLIN0 RLIN1 RLIN2 RLIN3
		CLKJIT			
		PLLFIXCLK /20[*2]			

Note 1. The default value of the clock selector and divider registers are denoted in *italics*.

Note 2. This selection option is only available for D1M1(H), D1M1-V2, D1M1A devices.

Note 3. In D1L1, other divisors are not available.

Table 12.106 Isolated-Area clock domain clocks maximum frequencies

Base clock	Maximum frequency								
	D1L1	D1L2	D1L2H	D1M1	D1M1H	D1M1-V2	D1M1A	D1M2	D1M2H
C_ISO_MLBB	–								120 MHz
C_ISO_SFMA	80 MHz	240 MHz							
C_ISO_RSCAN	40 MHz								
C_ISO_RSCANXIN	16 MHz								
C_ISO_SSIF	50 MHz								
C_ISO_TAUB01	80 MHz								
C_ISO_TAUB2	80 MHz								
C_ISO_TAUJ	80 MHz								
C_ISO_OSTM	80 MHz								
C_ISO_LCBI	80 MHz				–	80 MHz	–		
C_ISO_ADCE	40 MHz								
C_ISO_ISM	80 MHz								
C_ISO_RLIN	48 MHz			24 MHz*1		48 MHz			

Note 1. When used 50 MHz for PCLK

12.5.5 Video output channels clock domains

The video output channels have additional pixel clock generators.
Refer to Section 37.5, Video channels clock generation for details.

12.6 Frequency Output Function (FOUT)

The Frequency Output function allows to output the CKSCLK_AFOUT as the external signal CSCXFOUT.

Furthermore, the frequency of CKSCLK_AFOUT can be divided by the FOUT clock divider before it is output.

The figure below outlines the Frequency Output function.

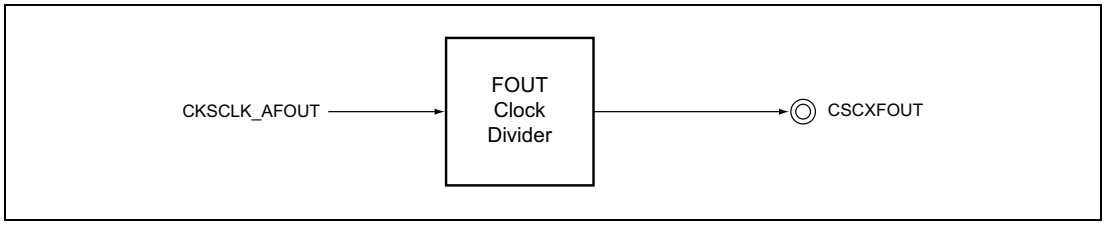


Figure 12.17 Frequency Output function

12.6.1 CSCXFOUT Clock Divider

The CSCXFOUT Clock Divider divides its input clock CKSCLK_AFOUT by 1 to 512 and thus provides a variable CSCXFOUT clock.

The divisor value N is defined by FOUTDIV.FOUTDIV[9:0], where

N = FOUTDIV.FOUTDIV[9:0]

Note that the clock output is stopped if FOUTDIV.FOUTDIV[9:0] = 000_H.

The frequency $f_{CSCXFOUT}$ of the output clock is calculated by

$f_{CSCXFOUT}$ = CKSCLK_AFOUT / N

When a new clock divisor is written to the FOUTDIV.FOUTDIV[9:0] bits, it becomes effective in synchronization with the CSCXFOUT output clock. Accordingly, the clock divisor can be changed even while the CSCXFOUT clock is operating.

12.6.2 Clock supply

FOUT provides one clock input:

Table 12.107 FOUT clock supply

Module	Clock	Connect to
FOUT	CKSCLK_AFOUT	Clock domain C_AWO_FOUT

12.6.3 Register of the CSCXFOUT Clock Divider

The CSCXFOUT Clock Divider is controlled and operated by the following register.

Table 12.108 Register of the CSCXFOUT Clock Divider

Module Name	Register Name	Symbol	Address
SYS	Clock divider register	FOUTDIV	FFF8 2800_H
SYS	Clock divider status register	FOUTSTAT	FFF8 2804_H

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

12.6.4 CSCXFOUT Clock Divider Control Register Details

12.6.4.1 FOUTDIV — Clock divider register

This register defines the clock divisor.

Access: This register can be read/written in 32-bit units.

Address: FFF8 2800_H

Initial value: 0000 0000_H This register is initialized by SYSRES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	FOUTDIV[9:0]									
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.109 FOUTDIV register contents

Bit Position	Bit Name	Function
31 to 10	Reserved	When written, write the initial value.
9 to 0	FOUTDIV[9:0]	Clock divisor N 000_H: Clock output is stopped (default) 001_H: N = 1 002_H: N = 2 ... $1FF_H$: N = 511 200_H: N = 512 All others: Setting prohibited

12.6.4.2 FOUTSTAT — Clock divider status register

This register defines the clock divider status.

Access: This register can be read in 32-bit units.

Address: FFF8 2804_H

Initial value: 0000 0001_H This register is initialized by SYSRES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FOUTCLKACT	FOUTSYNC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.110 FOUTSTAT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	FOUTCLKACT	Divider clock active bit 0: CLKOUT inactive 1: CLKOUT active
0	FOUTSYNC	Divider clock synchronized 0: CLKOUT does not correspond to the actual divisor setting in FOUTDIV 1: CLKOUT corresponds to the actual divisor setting in FOUTDIV

12.7 Clock Monitor A (CLMA)

This section contains a generic description of the Clock Monitor A (CLMA).

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.

The subsequent sections describe the features that apply to all implementations.

12.7.1 Overview of RH850/D1L/D1M CLMA

12.7.1.1 Units

This microcontroller has the following numbers of Units of the Clock Monitor A.

Table 12.111 Units

Clock Monitor A	D1L1	D1L2(H)	D1M1(H)	D1M1-V2	D1M1A	D1M2	D1M2H
Units	5	5	6	6	6	6	7
Names	CLMAn (n = 0 to 4)	CLMAn (n = 0 to 4)	CLMAn (n = 0 to 5)	CLMAn (n = 0 to 5)	CLMAn (n = 0 to 5)	CLMAn (n = 0 to 4, 6)	CLMAn (n = 0 to 6)

Index n

Throughout this section, the individual units of a Clock Monitor A are identified by the index “n” (n = 0 to 6), for example, CLMAnCTL0 for the control register 0 of CLMAn.

12.7.1.2 Register addresses

All CLMAn register addresses are given as address offsets from the individual base address <CLMAn_base>.

The base address <CLMAn_base> of each CLMAn is listed in the following table.

Table 12.112 Register base addresses <CLMAn_base>

CLMAn	<CLMAn_base> address
CLMA0	FFF8 C000$_H$
CLMA1	FFF8 F000$_H$
CLMA2	FFF8 F080$_H$
CLMA3	FFF8 F100$_H$
CLMA4	FFF8 F180$_H$
CLMA5	FFF8 F200$_H$
CLMA6	FFF8 F280$_H$

12.7.1.3 Clock supply

The monitored and the sampling clocks of all Clock Monitors A are listed in the following table.

Table 12.113 CLMAn clock supply (1/2)

Module	Clock	Connected to
CLMA0	Register access clock	Clock Controller • C_ISO_PCLK in RUN mode • EMCLK in DEEPSTOP mode
	CLMATSMP	Low Speed IntOsc f_{RL} (240 kHz)
	CLMATMON	High Speed IntOsc f_{RH} (8 MHz)

Table 12.113 CLMAn clock supply (2/2)

Module	Clock	Connected to
CLMA1	Register access clock	Clock Controller • C_ISO_PCLK in RUN mode • EMCLK in DEEPSTOP mode
	CLMATSMP	High Speed IntOsc f_{RH}/256 (31.25 kHz)
	CLMATMON	Low Speed IntOsc f_{RL} (240 kHz)
CLMA2	Register access clock	Clock Controller • C_ISO_PCLK in RUN mode • EMCLK in DEEPSTOP mode
	CLMATSMP	Low Speed IntOsc f_{RL} (240 kHz)
	CLMATMON	MainOsc
CLMA3	Register access clock	Clock Controller C_ISO_PCLK
	CLMATSMP	Low Speed IntOsc f_{RL} (240 kHz)
	CLMATMON	$f_{PLL0CLK}$ / 16
CLMA4	Register access clock	Clock Controller C_ISO_PCLK
	CLMATSMP	Low Speed IntOsc f_{RL} (240 kHz)
	CLMATMON	$f_{PLL1CLK}$ / 16
CLMA5	Register access clock	Clock Controller C_ISO_PCLK
	CLMATSMP	Low Speed IntOsc f_{RL} (240 kHz)
	CLMATMON	Port VDCE0_VI_CLK (Video Input Interface 0 pixel clock)
CLMA6	Register access clock	Clock Controller C_ISO_PCLK
	CLMATSMP	Low Speed IntOsc f_{RL} (240 kHz)
	CLMATMON	Port VDCE1_VI_CLK (Video Input Interface 1 pixel clock)

12.7.1.4 CLMAn internal signal connections

The following table shows the connection of the CLMAn signals to other microcontroller internal modules.

Table 12.114 CLMAn internal signal connections

CLMAn signals	Function	Connected to
CLMA0:		
$\overline{\text{CLMATRES}}$	CLMA0 error reset	Reset Controller $\overline{\text{CLMA0RES}}$
CLMA1:		
CLMATINT	CLMA1 error reset	Error Control Module INTCLMATI1
CLMA2:		
CLMATINT	CLMA2 error reset	Error Control Module INTCLMATI2
CLMA3:		
CLMATINT	CLMA3 error reset	Error Control Module INTCLMATI3
CLMA4:		
CLMATINT	CLMA4 error reset	Error Control Module INTCLMATI4
CLMA5:		
CLMATINT	CLMA5 error reset	Error Control Module INTCLMATI5
CLMA6:		
CLMATINT	CLMA6 error reset	Error Control Module INTCLMATI6

12.7.2 CLMA Enabling

Clock monitoring is started by the clock monitor when CLMAnCTL0.CLMAnCLME = 1.

When the monitored clock is stopped by register operation or standby mode, the corresponding clock monitor is automatically disabled. After the monitored clock starts oscillation again and becomes stable, the clock monitor also starts operation.

A CLMAn that is enabled can be only disabled by reset.
Thus writing CLMAnCTL0.CLMAnCLME = 0 cannot disable the CLMAn.

12.7.3 Start and stop of video input clock monitors CLMA5 and CLMA6

If a clock monitor is enabled by CLMAnCTL0.CLMAnCLME = 1, it can not be stopped by the application program afterwards. Only a reset would stop the clock monitors.

In order to stop the clock monitors CLMA5 and CLMA6, which monitor the external video input clocks of the video channels without applying a reset, the Video input clock monitors control register CLMAOTCTL0 is provided to enable and disable these clock monitors, if required.

Enabling and starting the video input clock monitor (n = 5, 6)

1. Activate the external video input clock.
2. Enable the respective clock monitor CLMAn by CLMAnCTL0.CLMAnCLME = 1.
3. Start video input clock monitoring by CLMAOTCTL0.CLMAnEN = 1.

Stopping the video input clock monitor (n = 5, 6)

1. Stop video input clock monitoring by CLMAOTCTL0.CLMAnEN = 0.
2. The respective external video input clock can be stopped.

12.7.3.1 CLMAOTCTL0 - Video input clock monitors control register (D1M2(H), D1M1(H), D1M1-V2, D1M1A only)

This register controls monitoring of the video clocks by the clock monitors CLMA5 and CLMA6.

Access: This register is readable/writable in 8-bit units.

Address: FFF8 F2C0$_H$

Initial value: 00$_H$

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	CLMA6EN	CLMA5EN
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 12.115 CLMAOTCTL0 Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When written, write the initial value.
1	CLMA6EN	Controls monitoring of the video input clock of video channel 1 by CLMA6 0: video input clock VDCE1_VI_CLK monitoring stopped 1: video input clock VDCE1_VI_CLK monitoring active
0	CLMA5EN	Controls monitoring of the video input clock of video channel 0 by CLMA5 0: video input clock VDCE0_VI_CLK monitoring stopped 1: video input clock VDCE0_VI_CLK monitoring active

NOTE

In the header files the module name of the above register is defined as:
CLMAC

12.7.4 CLMA Function

The clock monitor CLMAn detects an abnormal frequency of the monitored clock.

Overview of CLMA Features

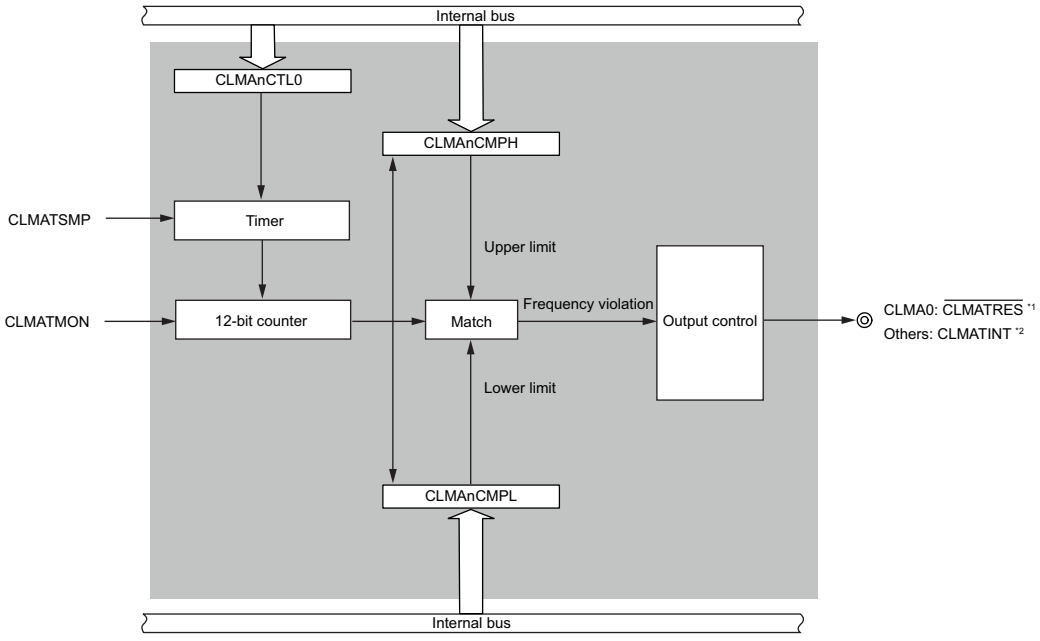
The clock monitor has the following features:

- Monitors the frequency of the input clock CLMATMON by using the sampling clock CLMATSMP.
- Outputs a reset request signal when an abnormal clock frequency is detected.

NOTE

Once enabled, only a reset can disable the CLMAn.

The following figure shows the main components of the Clock Monitor.



Note 1. $\overline{\text{CLMATRES}}$ is connected to Reset Controller.
Note 2. CLMATINT is connected to Error Control Module.

Figure 12.18 Block diagram of the Clock Monitor A

12.7.5 Description of Functions

The Clock Monitor CLMAn is used to verify whether the frequency of a clock (CLMATMON) is within the specified range.

12.7.5.1 Detection of Abnormal Clock Frequencies

Detection Method

1. CLMAn counts the rising edges of the monitored clock CLMATMON within 16 cycles of the sampling clock CLMATSMP and then compares the counter value with the specified thresholds:
   - CLMAnCMPL.CLMAnCMPL[11:0] defines the lower threshold.
   - CLMAnCMPH.CLMAnCMPH[11:0] defines the upper threshold.
2. When CLMATMON's frequency is lower than the limit*, the counter falls below CLMAnCMPL.CLMAnCMPL[11:0].
3. When the frequency of CLMATMON is higher than the limit, the counter exceeds CLMAnCMPH.CLMAnCMPH[11:0].

NOTE*

There is a case that the abnormal state is not detected when the CLMATMON completely stops.

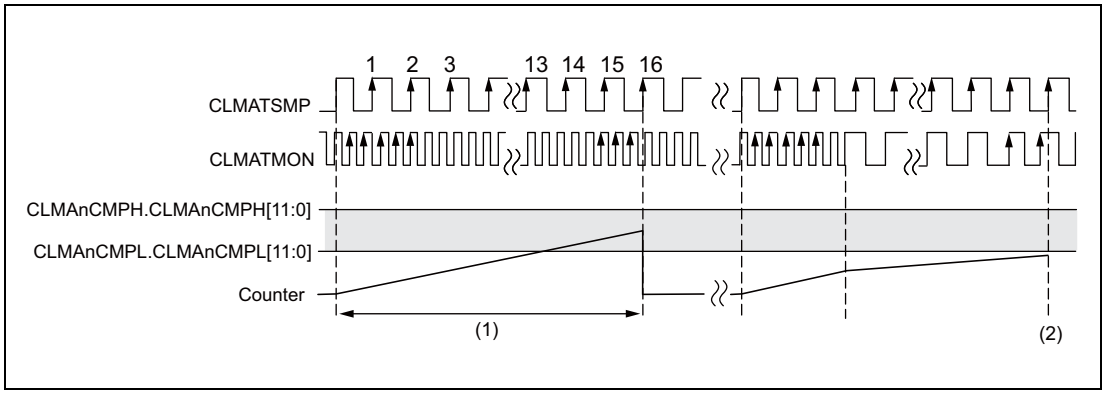


Figure 12.19 Example: $f_{CLMATMON}$ is lower than the specified limit.

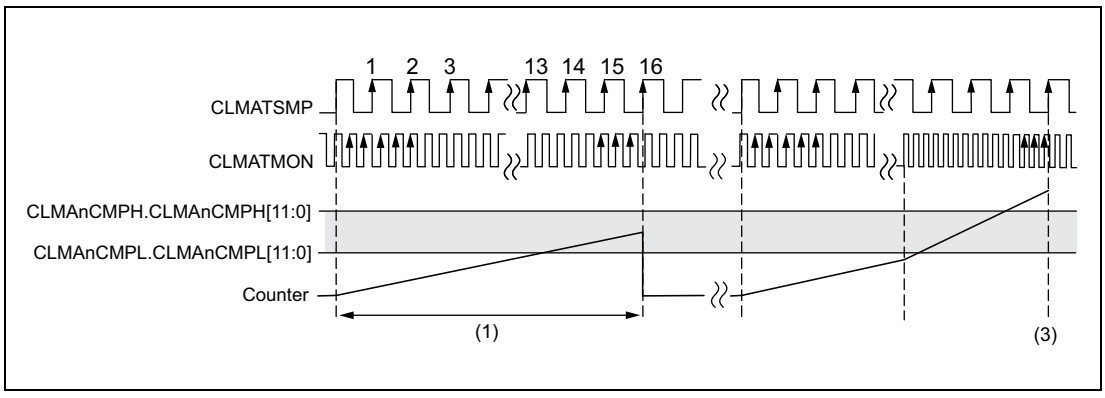


Figure 12.20 Example: $f_{CLMATMON}$ is higher than the specified limit.

NOTE

Even if $f_{CLMATMON}$ exceeds or falls below the specified limits during a sampling interval, the counter might be within the valid range.

Abnormal $f_{CLMATMON}$ is detected after one sampling interval.

(1) Calculation method of the thresholds CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0]

The compare registers CLMAnCMPL and CLMAnCMPH are configured with the minimum and maximum number of clock cycles of CLMATMON that are assumed to be valid within 16 cycles of the sampling clock CLMATSMP.

The expected number of clock cycles is denoted by N.

$$\frac{16}{f_{CLMATSMP}} = \frac{N}{f_{CLMATMON}}$$

$$N = \frac{f_{CLMATMON}}{f_{CLMATSMP}} \times 16$$

Considering the allowed frequency deviations of CLMATMON and CLMATSMP, the threshold values can be calculated by the following formulas:

$$\text{Lower threshold} = N_{min} = \frac{f_{CLMATMON(min)}}{f_{CLMATSMP(max)}} \times 16 - 1$$

$$\text{Upper threshold} = N_{max} = \frac{f_{CLMATMON(max)}}{f_{CLMATSMP(min)}} \times 16 + 1$$

NOTE

The jitter of the PLL is covered by "+1" and "-1" in the formulae.

Example:

When $f_{CLMATSMP}$ = 240 kHz (±8%) and $f_{CLMATMON}$ = 16 MHz (±5%), the recommended threshold values are the followings:

N_{min} = 15,200 / 259.2 × 16 - 1
= 937.27
CLMAnCMPL = 937 = $03A9_H$

N_{max} = 16,800 / 220.8 × 16 + 1
= 1218.39
CLMAnCMPH = 1219 = $04C3_H$

Minimum thresholds

The following restrictions must be taken into account:

- CLMAnCMPL ≥ 0001_H
- CLMAnCMPH ≥ CLMAnCMPL + 0003_H

(2) Definition of the initial value input to the threshold registers

The initial and reset values of the threshold registers are set so that the maximum frequency deviation of the monitored clock is allowed:

- CLMAnCMPL[11:0] = 0001_H
- CLMAnCMPH[11:0] = $03FF_H$

12.7.5.2 Notification of Abnormal Clock Frequency

In case that $f_{CLMATMON}$ is higher than the upper or lower than the lower threshold:

- in case of CLMA0:
  - The reset request signal $\overline{\text{CLMATRES}}$ is set to low level.
  - The system reset SYSRES is generated and CLMAn is reset.
- In case of others (CLMAn: n = 1 to 6)
  - The error signal CLMATINT is stored to error status register in Error Control Module.

12.7.5.3 CLMAn Enable (Write to CLMAnCTL0)

Writing to this register (CLMAnCTL0) is protected by a special sequence of instructions by using the protection command register CLMAnPCMD. Refer to Section 4, Write-Protected Registers.

NOTE

CLMAn can be only disabled by reset. Writing to CLMAnCTL0 cannot disable CLMAn.

12.7.6 Clock Monitor Registers

The Clock Monitor is controlled and operated by the following registers.

Table 12.116 List of Clock Monitor Registers

Module Name	Register Name	Symbol	Address
	CLMAn control register 0	CLMAnCTL0	<CLMAn_base> + 00_H
	CLMAn compare register L	CLMAnCMPL	<CLMAn_base> + 08_H
	CLMAn compare register H	CLMAnCMPH	<CLMAn_base> + $0C_H$
CLMAC	CLMA test register	CLMATEST	FFF8 $C100_H$
CLMAC	CLMA test status register	CLMATESTS	FFF8 $C104_H$
CLMAC	CLMA test register 2	CLMATEST2	FFF8 $F600_H$
CLMAC	CLMA test status register 2	CLMATESTS2	FFF8 $F604_H$
CLMAC	CLMA test register 3	CLMATEST3	FFF8 $F300_H$
CLMAC	CLMA test status register 3	CLMATESTS3	FFF8 $F304_H$
	CLMAn emulation register 0	CLMAnEMU0	<CLMAn_base> + 18_H
Write protection registers			
	CLMA0CTL0 protection command register	CLMA0PCMD	FFF8 $C010_H$
	CLMA0CTL0 protection status register	CLMA0PS	FFF8 $C014_H$
	CLMA1CTL0 protection command register	CLMA1PCMD	FFF8 $F010_H$
	CLMA1CTL0 protection status register	CLMA1PS	FFF8 $F014_H$
	CLMA2CTL0 protection command register	CLMA2PCMD	FFF8 $F090_H$
	CLMA2CTL0 protection status register	CLMA2PS	FFF8 $F094_H$
	CLMA3CTL0 protection command register	CLMA3PCMD	FFF8 $F110_H$
	CLMA3CTL0 protection status register	CLMA3PS	FFF8 $F114_H$
	CLMA4CTL0 protection command register	CLMA4PCMD	FFF8 $F190_H$
	CLMA4CTL0 protection status register	CLMA4PS	FFF8 $F194_H$
	CLMA5CTL0 protection command register	CLMA5PCMD	FFF8 $F210_H$
	CLMA5CTL0 protection status register	CLMA5PS	FFF8 $F214_H$
	CLMA6CTL0 protection command register	CLMA6PCMD	FFF8 $F290_H$
	CLMA6CTL0 protection status register	CLMA6PS	FFF8 $F294_H$
CLMAC	CLMATEST protection command register	PROTCMDCLMA	FFF8 $C200_H$

<CLMAn_base>

The base addresses <CLMAn_base> of the CLMAn are defined in “Register addresses” of the first part in Section 12.7.1.2, Register addresses.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

12.7.6.1 CLMAnCTL0 - CLMAn control register 0

This register enables the clock monitor CLMAn.

Writing to this register is protected by a special sequence of instructions by using the protection command register CLMAnPCMD. For details, see Section 4, Write-Protected Registers.

Access: This register is read/written in 8-bit units.

Address: <CLMAn_base> + 00_H

Initial value: 00_H. This register can be initialized by any reset source.*1

Note 1. CLMAnCTL (n = 0 to 2) can be initialized only by PURES and SYSRES.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CLMAnCLME
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 12.117 CLMAnCTL0 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When written, write the initial value.
0	CLMAnCLME	Enables the clock monitor. 0: CLMAn is disabled.*1 1: Enable the CLMAn.

Note 1. After the CLMAnCTL0.CLMAnCLME bit is set to 1, writing 0 to this bit is ignored. The only condition for clearing the bit is a reset (PURES, SYSRES). In addition, the bit is cleared when CLMATEST.RESCLM bit is set to 1 during self-test of CLMAn.

CAUTION

Write 0 in bit 7 to 1.

12.7.6.2 CLMAnCMPH - CLMAn compare register H

This register specifies the upper limit of frequency.

Refer to **12.7.5.1** (1), Calculation method of the thresholds CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0] for details.

Access: This register is read/written in 16-bit units.
It can only be written, when CLMAn is disabled (CLMAnCTL0.CLMAnCLME = 0).

Address: <CLMAn_base> + $0C_H$

Initial value: $03FF_H$. This register can be initialized by any reset source.*1

Note 1. CLMAnCTL (n = 0 to 2) can be initialized only by PURES and SYSRES.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CLMAnCMPH[11:0]											
Initial value	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.118 CLMAnCMPH Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When written, write the initial value.
11 to 0	CLMAnCMPH [11:0]	Specifies the upper threshold. • The recommended value is $f_{CLMATMON\ (max)}$ / $f_{CLMATSMP\ (min)}$ × 16 + 1. • The minimum value is CLMAnCMPL + 0003_H.

CAUTION

Write 0 in bit 15 to 12.

12.7.6.3 CLMAnCMPL - CLMAn compare register L

This register specifies the lower limit of frequency.

Refer to **12.7.5.1** (1), Calculation method of the thresholds CLMAnCMPL.CLMAnCMPL[11:0] and CLMAnCMPH.CLMAnCMPH[11:0] for details.

Access: This register is read/written in 16-bit units.
It can only be written, when CLMAn is disabled (CLMAnCTL0.CLMAnCLME=0).

Address: <CLMAn_base> + 08_H

Initial value: 0001_H. This register can be initialized by any reset source.*1

Note 1. CLMAnCTL (n = 0 to 2) can be initialized only by PURES and SYSRES.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CLMAnCMPL[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12.119 CLMAnCMPL Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When written, write the initial value.
11 to 0	CLMAnCMPL [11:0]	Specifies the lower threshold. • The recommended value is $f_{CLMATMON\ (min)}$ / $f_{CLMATSMP\ (max)}$ × 16 - 1 • The minimum value is 0001_H.

CAUTION

Write 0 in bit 15 to 12.

12.7.6.4 CLMATEST — CLMA Test Register

This register is used to test CLMA0.

Writing to this register is protected by the special sequence of command register PROTCMDCLMA. For details, see Section 4, Write-Protected Registers.

Access: This register can be read/written in 32-bit units.

Address: FFF8 $C100_H$

Value after reset: $0000\ 0000_H$ This register is initialized by PURES and SYSRES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CLMA0 TESEN	ERR MSK	MONCL KMSK	RES CLM
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 12.120 CLMATEST Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When writing to these bits, write the value after reset. When writing to these bits, write the value after reset.
3	CLMA0TESEN	CLMA0 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
2	ERRMSK*1	CLMA0 Error ($\overline{\text{CLMA0RES}}$) Signal Mask Setting 0: Signal generation enabled 1: Signal generation disabled (masked)
1	MONCLKMSK*1	Monitor Clock Mask Setting 0: Monitor clock enabled 1: Monitor clock disabled (masked)
0	RESCLM*1	Reset Control of CLMA0 0: Reset released 1: Reset executed

Note 1. This bit is enabled when the CLMAnTESEN bit is 1.

12.7.6.5 CLMATESTS — CLMA Test Status Register

This register is used to test CLMA0.

Access: This register can only be read 32-bit units.

Address: FFF8 C104$_H$

Value after reset: 0000 0000$_H$ This register is initialized by PURES and SYSRES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLMA0 ERRS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.121 CLMATESTS Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned.
0	CLMA0ERRS	CLMA0 Error Status 0: Errors are not detected 1: Errors are detected

12.7.6.6 CLMATEST2 — CLMA Test Register 2

This register is used to test CLMA1 and CLMA2.

Writing to this register is protected by the special sequence of command register PROTCMDCLMA2. For details, see Section 4, Write-Protected Registers.

Access: This register can be read/written in 32-bit units.

Address: FFF8 F600$_H$

Value after reset: 0000 0000$_H$ This register is initialized by PURES and SYSRES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	CLMA2 TESEN	CLMA1 TESEN	—	—	MONCL KMSK	RES CLM
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Table 12.122 CLMATEST2 Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When writing to these bits, write the value after reset. When writing to these bits, write the value after reset.
5	CLMA2TESEN	CLMA2 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
4	CLMA1TESEN	CLMA1 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
3, 2	Reserved	When writing to these bits, write the value after reset.
1	MONCLKMSK*1	Monitor Clock Mask Setting 0: Monitor clock enabled 1: Monitor clock disabled (masked)
0	RESCLM*1	Reset Control of CLMA1/2 0: Reset released 1: Reset executed

Note 1. This bit is enabled when the CLMAnTESEN bit is 1.

12.7.6.7 CLMATESTS2 — CLMA Test Status Register 2

This register is used to test CLMA1 and CLMA2.

Access: This register can only be read 32-bit units.

Address: FFF8 F604$_H$

Value after reset: 0000 0000$_H$ This register is initialized by PURES and SYSRES.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLMA2 ERRS	CLMA1 ERRS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.123 CLMATESTS2 Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is returned.
1	CLMA2ERRS	CLMA2 Error Status 0: Errors are not detected 1: Errors are detected
0	CLMA1ERRS	CLMA1 Error Status 0: Errors are not detected 1: Errors are detected

12.7.6.8 CLMATEST3 — CLMA Test Register 3

This register is used to test CLMA3 to CLMA6.

Writing to this register is protected by the special sequence of command register PROTCMDCLMA3. For details, see Section 4, Write-Protected Registers.

Access: This register can be read/written in 32-bit units.

Address: FFF8 F300$_H$

Value after reset: 0000 0000$_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLMA6 TESEN	CLMA5 TESEN	CLMA4 TESEN	CLMA3 TESEN	—	—	MONCL KMSK	RES CLM
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Table 12.124 CLMATEST3 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write the value after reset. When writing to these bits, write the value after reset.
7	CLMA6TESEN	CLMA6 Self-Test Enable/Disable (D1M2(H) only) 0: Test disabled 1: Test enabled
6	CLMA5TESEN	CLMA5 Self-Test Enable/Disable (D1M2(H), D1M1(H), D1M1-V2, D1M1A only) 0: Test disabled 1: Test enabled
5	CLMA4TESEN	CLMA4 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
4	CLMA3TESEN	CLMA3 Self-Test Enable/Disable 0: Test disabled 1: Test enabled
3, 2	Reserved	When writing to these bits, write the value after reset.
1	MONCLKMSK*1	Monitor Clock Mask Setting 0: Monitor clock enabled 1: Monitor clock disabled (masked)
0	RESCLM*1	Reset Control of CLMA3/4/5/6 0: Reset released 1: Reset executed

Note 1. This bit is enabled when the CLMAnTESEN bit is 1.

12.7.6.9 CLMATESTS3 — CLMA Test Status Register 3

This register is used to test CLMA3 to CLMA6.

Access: This register can only be read 32-bit units.

Address: FFF8 F304$_H$

Value after reset: 0000 0000$_H$ This register is initialized by any reset source.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CLMA6 ERRS	CLMA5 ERRS	CLMA4 ERRS	CLMA3 ERRS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12.125 CLMATESTS3 Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	CLMA6ERRS	CLMA6 Error Status (D1M2(H) only) 0: Errors are not detected 1: Errors are detected
2	CLMA5ERRS	CLMA5 Error Status (D1M2(H), D1M1(H), D1M1-V2, D1M1A only) 0: Errors are not detected 1: Errors are detected
1	CLMA4ERRS	CLMA4 Error Status 0: Errors are not detected 1: Errors are detected
0	CLMA3ERRS	CLMA3 Error Status 0: Errors are not detected 1: Errors are detected

12.7.6.10 CLMAnEMU0 - CLMAn emulation register 0

This register provides the bits to emulate a frequency deviation while the microcontroller is set in break mode during debugging.

Access: This register is read/written in 8-bit units.

Address: <CLMAn_base> + 18_H

Initial value: 00_H. This register can be initialized by any reset source.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CLMAnSLFST	CLMAnSLSLW
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 12.126 CLMAnEMU0 Register Contents

Bit Position	Bit Name	Function
1	CLMAnSLFST	Specifies the higher value of $f_{CLMATMON}$ during emulation: 0: CLMATMON is assumed to be within the normal frequency range 1: CLMATMON is assumed to exceed the upper threshold.
0	CLMAnSLSLW	Specifies the lower value of $f_{CLMATMON}$ during emulation: 0: CLMATMON is assumed to be within the normal frequency range 1: CLMATMON is assumed to fall below the lower threshold.

CAUTIONS

1. It is prohibited to emulate a too low and too high CLMATMON at the same time. Thus CLMAnEMU0 must not be set to 03_H.
2. Write 0 in bit 7 to 2.

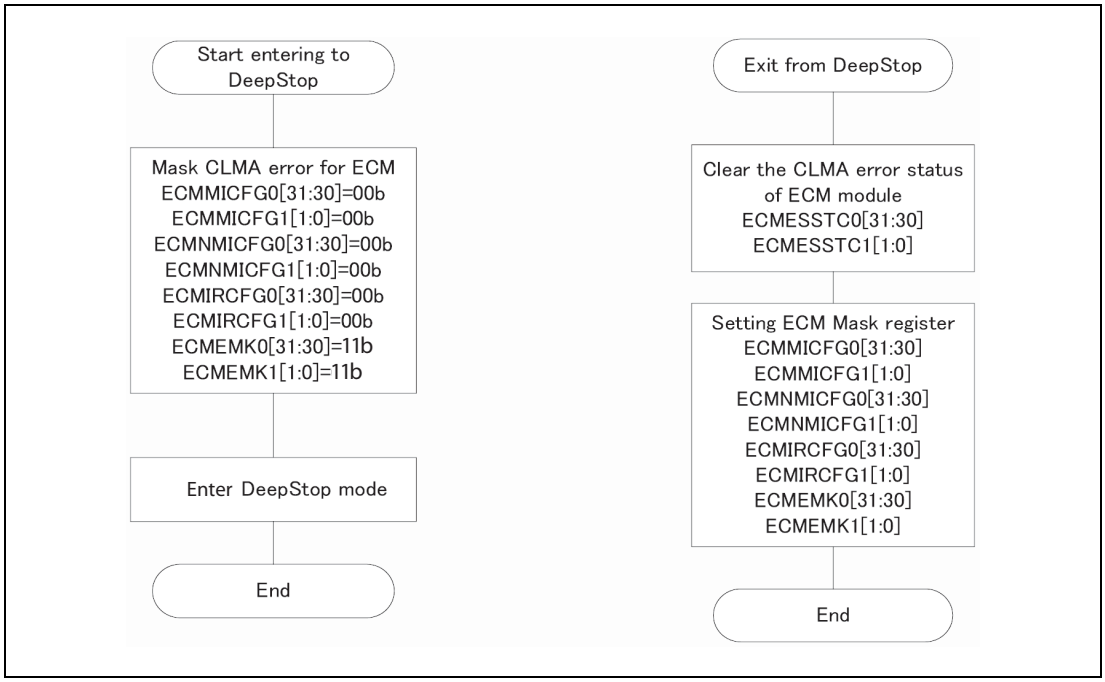
12.7.7 Usage Notes

12.7.7.1 ECM detects false error signals from CLMA at resuming DEEPSTOP

ECM sometimes detects false error signals from CLMA3 to CLMA6 at resuming from DEEPSTOP.

Before entering DEEPSTOP, mask error signals of CLMA3 to CLMA6 on ECM.

And after resuming from DEEPSTOP, clear error status flags of CLMA3 to CLMA6 and set expected mask setting of ECM again.
Details are shown below.



Section 13 Stand-by Controller (STBC)

This section describes the functions of the stand-by controller (STBC), the registers, and various standby modes.

13.1 Functions

The RH850/D1L/D1M supports the following power-save modes:

- HALT mode

  HALT mode can be entered from normal RUN mode by performing the CPU instruction “HALT”. This stops the CPU operation, while all clocks other than the CPU clock continue to operate and all areas remain under power.
  For details, refer to “*RH850 Family User’s Manual: Software*”.

- DEEPSTOP mode

  In order to reduce power consumption, the power supply of the Isolated-Area can be switched off. DEEPSTOP mode is entered when the STBC0PSC.STBC0DISTRG is set to 1.

NOTE

The mode in which the CPU is operating is called RUN mode.

13.1.1 Clock Supply

The Stand-by Controller clock supply is shown in the following table.

Table 13.1 Clock Supply

Module	Unit Clock Name	Connected to
Stand-by Controller	Register access clock	Clock Controller • C_ISO_PCLK in RUN mode • EMCLK in DEEPSTOP mode

13.1.2 Wake-up

13.1.2.1 Wake-up factors

The wake-up events for terminating a power save mode are controlled and monitored by the following Stand-by Controller registers:

- WUF0, WUFMSK0, WUFC0

The assignment of the wake-up events to the wake-up control and status register bits is given in the table below.

For details about the wake-up control and status registers, refer to Section 13.2.2.2, WUF0 — Wake-up factor register, Section 13.2.2.3, WUFMSK0 — Wake-up factor mask register, and Section 13.2.2.4, WUFC0 — Wake-up factor clear registers.

Table 13.2 Wake-up factors Registers Assignments

Name	Assignment of WUFm/WUFMSKm/ WUFCm Register Bits			Function
INTDCUTDI	WUF0[0]	WUFMSK0[0]	WUFC0[0]	OCD
INTP0 (RLIN31RX)	WUF0[1]	WUFMSK0[1]	WUFC0[1]	Port
INTP1	WUF0[2]	WUFMSK0[2]	WUFC0[2]	Port
INTP2 (RLIN32RX)	WUF0[3]	WUFMSK0[3]	WUFC0[3]	Port
INTP3	WUF0[4]	WUFMSK0[4]	WUFC0[4]	Port
INTP4 (RLIN33RX)	WUF0[5]	WUFMSK0[5]	WUFC0[5]	Port
INTP5	WUF0[6]	WUFMSK0[6]	WUFC0[6]	Port
INTP6 (CAN0RX)	WUF0[7]	WUFMSK0[7]	WUFC0[7]	Port
INTP7	WUF0[8]	WUFMSK0[8]	WUFC0[8]	Port
INTP8 (CAN1RX/RLIN30RX)	WUF0[9]	WUFMSK0[9]	WUFC0[9]	Port
INTP9	WUF0[10]	WUFMSK0[10]	WUFC0[10]	Port
INTP10 (CAN2RX/RLIN30RX)	WUF0[11]	WUFMSK0[11]	WUFC0[11]	Port
INTWDTA0	WUF0[12]	WUFMSK0[12]	WUFC0[12]	WDTA0
INTRTCA01S	WUF0[13]	WUFMSK0[13]	WUFC0[13]	RTCA0
INTRTCA0AL	WUF0[14]	WUFMSK0[14]	WUFC0[14]	RTCA0
INTRTCA0R	WUF0[15]	WUFMSK0[15]	WUFC0[15]	RTCA0
INTAWOT0	WUF0[16]	WUFMSK0[16]	WUFC0[16]	AWOT0
ECMTI	WUF0[17]	WUFMSK0[17]	WUFC0[17]	ECM
ECMTNMI	WUF0[18]	WUFMSK0[18]	WUFC0[18]	ECM

CAUTIONS

1. **The interrupt signals INTP0 to INTP10 can not be used for wake-up from DEEPSTOP mode, when the interrupts are connected to ISO area port groups such as P42 or P44. INTP interrupt signals are only wake-up capable when used on AWO port groups JP0 and P0.**
2. **When an external RS-CAN CANnRX or RLIN3 RLIN3nRX signal is to be used for wake up, the corresponding external interrupt INTPx must be configured for the falling edge detection.**
3. **Except in D1M1A and D1M1-V2, CPU clock must be changed to EMCLK when the interrupt signals INTP0 to INTP10 are used for wake-up factor from DEEPSTOP mode. Refer to Figure 13.3, Example of DEEPSTOP mode transition (except for D1M1A, D1M1-V2) for details.**
4. **Except in D1M1A and D1M1-V2, there are possibility that MCU does not wake-up. When using the interrupt signals INTP0 to INTP10 for wake-up factor, set the input pulse width long enough or configure that high speed internal oscillator does not stop in DEEPSTOP (ROSCSTPM.ROSCSTPMSK = 1). To refer the sufficient value, see RH850/D1L/D1M Group DATASHEET.**

On-Chip Debug wake-up

The On-Chip Debug unit (OCD) is generating a wake-up event while the microcontroller runs the application program in the following cases:

- the debugger issues a stop request
- a breakpoint is hit
  Note: A breakpoint hit occurs only in normal mode.

In either case any stand-by mode is terminated, provided the OCD debug event is enabled as a wake-up factor via the WUFMSK0 register.

CAUTION

If the OCD wake-up event is disabled, it is not possible to wake-up the microcontroller from stand-by mode by a manual stop via the debugger.

Thus it is recommended to enable the OCD wake-up for terminating the DEEPSTOP mode by setting WUFMSK0[0] = 0.

When the hot plug-in function is used, make sure to enable this factor. The INTDCUTDI interrupt is required to return from power save mode as the wake-up handling.

13.1.2.2 Wake-up control

Wake-up factors are selected by the wake-up factors control registers:

- Wake-up mask register WUFMSK0
  Each bit of this register is assigned to a certain wake-up event. Wake-up by this event is enabled if its mask bit is set to 0.
- Wake-up factor register WUF0
  Upon occurrence of an unmasked wake-up event, the associated wake-up factor flag is set to 1. By use of this register the application program can identify the wake-up factor.
- Wake-up factor clear register WUFC0
  In order to clear a wake-up factor flag of the wake-up factor register (WUF0), its assigned bit has to be set to 1.

NOTE

The wake-up factor flags in the wake-up factors register (WUF0) indicate only the occurrence of a wake-up factor. Thus an asserted wake-up factor flag does not mean that the transition from DEEPSTOP to normal operation mode is already accomplished.
This means that wake-up factor bits could be set during normal operation mode.

13.1.3 I/O buffer control

This section describes the behavior of the I/O buffers during various DEEPSTOP modes.

13.1.3.1 I/O buffer hold state

During the I/O buffer hold state, the I/O buffer maintains the state it was in before entering this state. Therefore, no external or internal signal can change the state of the I/O buffer until the I/O buffer hold state is terminated.

The port buffers on the Isolated-Area, that are supplied with BnVCC and ISMVCC can be set into or released from IOHOLD state by the application program via the IOHOLD register.
Further the P0 port buffers on the Always-On-Area, that are supplied with EVCC can be set into or released from IOHOLD state by the application program via the IOHOLD register.
The IOHOLD register has a separate bit IOHOLDn for certain groups of the buffers, which are supplied by power supply B0VCC to B5VCC, EVCC and ISMVCC.

13.1.3.2 I/O buffers during DEEPSTOP mode

Before entering DEEPSTOP mode the Isolated-Area buffers can be set into IOHOLD as long as port buffers kept power supplied during DEEPSTOP mode. If power supply of port buffer is expected to stop during DEEPSTOP mode, the corresponding IOHOLD register must be set to 0.

Also the Always-On-Area port buffers of P0 may be set into IOHOLD before entering DEEPSTOP mode. In case P0 is not set into IOHOLD, the port buffers of P0 remain operational in DEEPSTOP mode.

In DEEPSTOP mode, the I/O buffers of port groups transition to I/O buffer hold state.

After wake-up from DeepSTOP, the I/O buffers remain in I/O buffer hold state until canceling the state by software. To cancel I/O buffer hold state, follow the steps shown below.

1. Re-configure the peripheral or port function.
2. Set the corresponding IOHOLD.IOHOLDn = 0

NOTES

1. The output level of the port buffers of the DDR2-SDRAM I/F for D1M2(H) become undefined in DEEPSTOP mode. Thus the DDR2-SDRAM I/F supply SDRBVCC must be switched off.
2. The output level of the port buffers of the
   – MIPI video input interface port group (P40)
   – RSDS video output interface port groups (P44 and P45)
   – Serial Flash Memory and Media Local Bus interface port group (P21)
   – SDR-SDRAM I/F port buffers for D1M1H and D1M1A
   become undefined in DEEPSTOP mode. Thus it is recommended to connect these pins to the respective ground or voltage supply via pull-up or -down resistors.

13.1.4 Clock supply in DEEPSTOP mode

Some clock generators and the clock domain of the Always-On-Area can optionally be stopped in DEEPSTOP mode and restarted automatically after wake-up, provided they are operating at DEEPSTOP entry.

The clock generator's and domain's DEEPSTOP behavior is selected via a stop mask register, that is available for each clock generator and domain, that allows to select stop/continue in DEEPSTOP:

- stop mask = 1: clock generator or domain status does not change in DEEPSTOP
- stop mask = 0: clock generator or domain stops in DEEPSTOP and returns to status as before DEEPSTOP (restarted when operating before DEEPSTOP or remains stopped)

Following clock generators and clock domains feature stop mask registers for stop/continue selection:

- MOSCSTPM: MainOsc
  If the MainOsc is stopped in DEEPSTOP mode (MOSCSTPM.MOSCSTPMSK = 0) the Clock Monitor CLMA2 is also stopped in DEEPSTOP.
- ROSCSTPM: High Speed IntOsc
  If the High Speed IntOsc is stopped in DEEPSTOP mode (ROSCSTPM.ROSCSTPMSK = 0) the Clock Monitor CLMA0 is also stopped in DEEPSTOP mode. Further when the High Speed IntOsc is stopped in DEEPSTOP mode (ROSCSTPM.ROSCSTPMSK = 0) the Clock Monitor CLMA1 must not be used.
- CKSC_AAWOTD_STPM: C_AWO_AWOT domain for the Always-On-Area Timer (AWOT)
- CKSC_ARTCAD_STPM: C_AWO_RTCA domain for the Real-Time Clock (RTCA)
- CKSC_AFOUTS_STPM: C_AWO_FOUT domain for the Frequency Output Function (FOUT)

NOTES

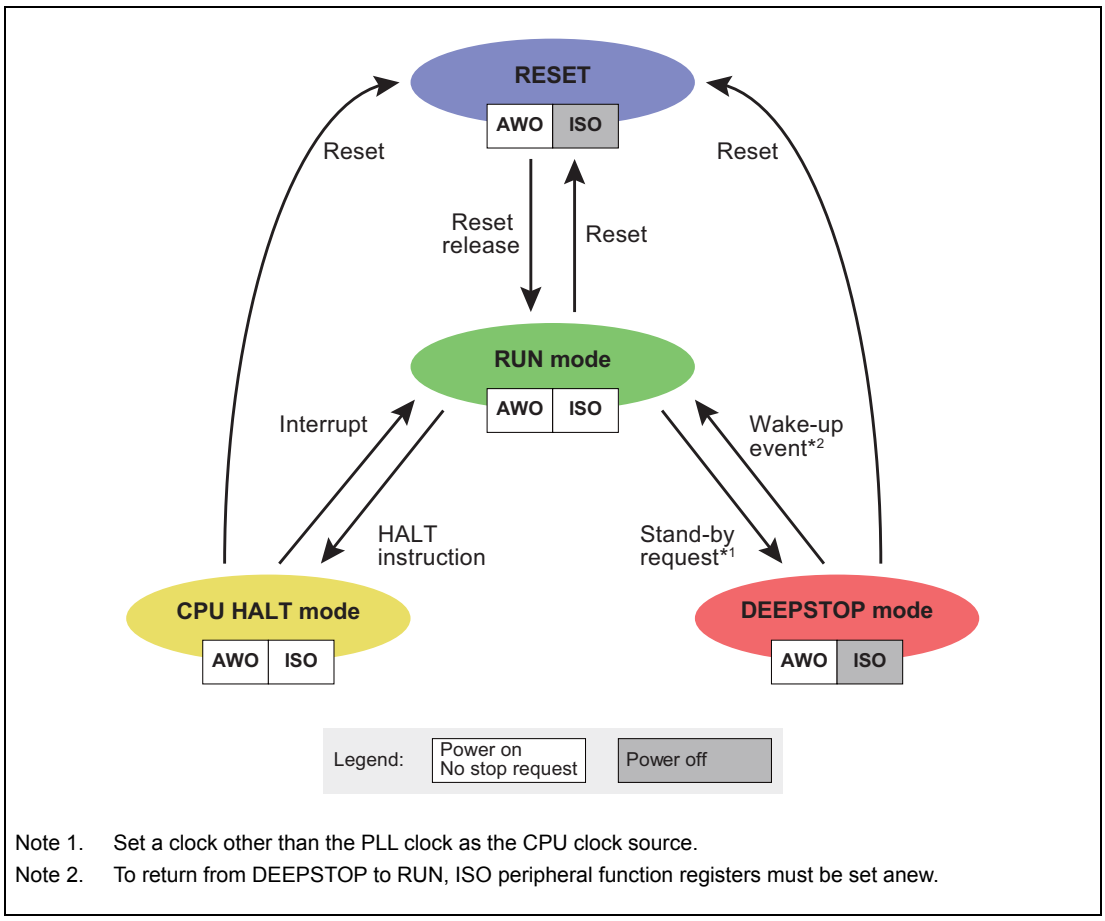
1. The Low Speed IntOsc is always operating and can not be stopped.
   If the High Speed IntOsc is stopped in DEEPSTOP mode (ROSCSTPM.ROSCSTPMSK = 0), the Low Speed IntOsc Clock Monitor CLMA1 must not be used. That means CLMA1 must never be enabled (keep CLMAnCTL0.CLMAnCLME = 0). The failing of Low Speed IntOsc can be detected by CLMA0 and CLMA2 to CLMA6 in this case.
2. The Watchdog Timer on the Always-On-Area (WDTA0) operates with the Low Speed IntOsc clock and thus remains in operation even in DEEPSTOP mode.
3. All PLL circuits reside on the Isolated-Area, thus are completely switched off in DEEPSTOP and must be configured and restarted after wake-up by the application program. The respective Clock Monitors CLMA3 and CLMA4 are also stopped in DEEPSTOP mode.
4. All Isolated-Area clock domains C_ISO_xxx and their selection registers reside on the Isolated-Area, thus are completely switched off in DEEPSTOP and must be configured and restarted after wake-up by the application program.
5. The Video Input Interfaces pixel Clock Monitors CLMA5 and CLMA6 must be stopped by the application software by setting CLMAOTCTL0.CLMA5EN = 0 and CLMAOTCTL0.CLMA6EN = 0 (refer to Section 12.7.3, Start and stop of video input clock monitors CLMA5 and CLMA6 for details).
6. The clock supply does not stop even if clock generator's stop request is not masked (stop mask = 0), while stop masks of clock domains are set to 1. Be sure to set stop mask bit of each domain register (CKSC_AAWOTD_STPM, CKSC_ARTCAD_STPM, CKSC_AFOUTS_STPM) to 0 before entering DEEPSTOP mode.

   [example – clock supply for RTCA]
   (Case-1)
   1. Select MainOSC by RTCA (CKSC_ARTCAS_CTL = 2_H)
   2. Enable "stop mask" of RTCA (CKSC_ARTCAD_STPM = 3_H)
   3. Disable "stop mask" of MainOSC (MOSCSTPM = 2_H)
   4. Transit to DEEPSTOP
   5. Both of MainOSC and RTCATCKI "do not stop"

   (Case-2)
   1. Select MainOSC by RTCA (CKSC_ARTCAS_CTL = 2_H)
   2. Disable "stop mask" of RTCA (CKSC_ARTCAD_STPM = 2_H)
   3. Disable "stop mask" of MainOSC (MOSCSTPM = 2_H)
   4. Transit to DEEPSTOP
   5. Both of MainOSC and RTCATCKI "stop"

13.1.5 Transition to power save mode

The figure below shows the transition between RUN mode, HALT mode and DEEPSTOP mode.



Note 1. Set a clock other than the PLL clock as the CPU clock source.
Note 2. To return from DEEPSTOP to RUN, ISO peripheral function registers must be set anew.

Figure 13.1 State transition diagram of stand-by mode

13.2 Registers

13.2.1 Overview of Stand-by Controller Registers

The Stand-by Controller is controlled and operated by the following registers.

Table 13.3 Overview of Stand-by Controller Registers

Module Name	Symbol	Initial Value	Attribute	Address
SYS	STBC0PSC	0000 0000$_H$	R/W	FFF8 0100$_H$
SYS	WUF0	0000 0000$_H$	R	FFF8 0400$_H$
SYS	WUFMSK0	FFFF FFFF$_H$	R/W	FFF8 0404$_H$
SYS	WUFC0	—	W	FFF8 0408$_H$
SYS	IOHOLD	0000 0000$_H$	R/W	FFF8 0B00$_H$

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

13.2.2 Stand-by controller control registers details

13.2.2.1 STBC0PSC — Stand-by control register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0.
For details, refer to **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (ISORES).

Access: This register can be read/written in 32-bit units.

Address: FFF8 0100$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STBC0 DISTRG	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R

Table 13.4 STBC0PSC Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	STBC0DISTRG	0: Writing “0” has no effect. 1: DEEPSTOP mode is entered
0	Reserved	When written, write the initial value.

13.2.2.2 WUF0 — Wake-up factor register

This register informs of a wake-up event in the isolated area.

WUF0 is initialized by all reset sources except the transition to DeepSTOP mode (ISORES).

Access: This register can be read in 32-bit units.

Address: FFF8 0400_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	WUF18	WUF17	WUF16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUF15	WUF14	WUF13	WUF12	WUF11	WUF10	WUF09	WUF08	WUF07	WUF06	WUF05	WUF04	WUF03	WUF02	WUF01	WUF00
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 13.5 WUF0 Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When written, write the initial value.
18 to 0	WUFy	Indicates the generation of a wake-up event. 0: Wake-up event is not generated 1: Wake-up event is generated

NOTE

While the WUFMSKy bit in the wake-up factor mask register is 1, WUFy is not set to 1 at the generation of a wake-up event.

Wake-up factor

As for the assignment of a wake-up event to the wake-up factor register bit, refer to **Table 13.2, Wake-up factors Registers Assignments**.

The bit to which a wake-up event is not assigned is read as the value “0”.

13.2.2.3 WUFMSK0 — Wake-up factor mask register

This register enables a wake-up event in the isolated area.

WUF0 is initialized by all reset sources except the transition to DeepSTOP mode (ISORES).

Access: This register can be read/written in 32-bit units.

Address: FFF8 0404$_H$

Initial value: FFFF FFFF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	WUFMSK18	WUFMSK17	WUFMSK16
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUFMSK15	WUFMSK14	WUFMSK13	WUFMSK12	WUFMSK11	WUFMSK10	WUFMSK09	WUFMSK08	WUFMSK07	WUFMSK06	WUFMSK05	WUFMSK04	WUFMSK03	WUFMSK02	WUFMSK01	WUFMSK00
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.6 WUFMSK0 Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When written, write the initial value.
18 to 0	WUFMSKy	Enables/disables a wake-up event. 0: Wake-up event is enabled 1: Wake-up event is disabled

NOTE

While the WUFMSKy bit is 1, WUFy of the wake-up factor register is not set to 1 at the generation of a wake-up event.

Wake-up factor

As for the assignment of a wake-up event to the wake-up factor register bit, refer to **Table 13.2, Wake-up factors Registers Assignments**.

13.2.2.4 WUFC0 — Wake-up factor clear registers

This register clears the WUFy bit in the wake-up factor register.

Access: This register can be written in 32-bit units.

Address: FFF8 0408$_H$

Initial value: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	WUFC 18	WUFC 17	WUFC 16
Initial value	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WUFC 15	WUFC 14	WUFC 13	WUFC 12	WUFC 11	WUFC 10	WUFC 09	WUFC 08	WUFC 07	WUFC 06	WUFC 05	WUFC 04	WUFC 03	WUFC 02	WUFC 01	WUFC 00
Initial value	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 13.7 WUFC0 Register Contents

Bit Position	Bit Name	Function
31 to19	Reserved	When read, an undefined value is returned.
18 to 0	WUFCy	Clear the wake-up factor WUFy in the wake-up factor register. 0: WUFy is not modified 1: WUFy is cleared

Wake-up factor

As for the assignment of a wake-up event to the wake-up factor register bit, refer to **Table 13.2, Wake-up factors Registers Assignments**.

13.2.2.5 IOHOLD — Port IOHOLD control register

This register controls the IOHOLD state of the I/O buffer.

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMD0.
For details, refer to **Section 4, Write-Protected Registers**.

This register is initialized by all reset sources except the transition to DeepSTOP mode (ISORES).

Access: This register can be read/written in 32-bit units.

Address: FFF8 0B00$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	IOHOLD6	IOHOLD5	IOHOLD4	IOHOLD3	IOHOLD2	IOHOLD1	IOHOLD0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13.8 IOHOLD Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 7	Reserved	When written, write the initial value.
6	IOHOLD6	This bit controls the IOHOLD function of the port groups P16 and P17 ports. 0: I/O hold state of the respective ports is released 1: I/O hold state of the respective ports is activated
5	IOHOLD5	This bit controls the IOHOLD function of the port group P42 ports. 0: I/O hold state of the respective ports is released 1: I/O hold state of the respective ports is activated
4	IOHOLD4	This bit controls the IOHOLD function of the ports P43_7 to P43_12. 0: I/O hold state of the respective ports is released 1: I/O hold state of the respective ports is activated
3	IOHOLD3	• D1L1, D1L2: This bit controls the IOHOLD function of the ports – P43_0, P43_1 – P44_1 to P44_4, P44_6 to P44_11 – P45_0 to P45_13. • D1L2H, D1M1(H), D1M1-V2, D1M1A: This bit controls the IOHOLD function of the ports – P43_0, P43_1 – P44_0 to P44_11 – P45_0 to P45_13. • D1M2(H): This bit controls the IOHOLD function of the ports P43_0 to P43_6 and of the port groups P46 and P47 ports. 0: I/O hold state of the respective ports is released 1: I/O hold state of the respective ports is activated
2	IOHOLD2	This bit controls the IOHOLD function of the port group P3 ports. 0: I/O hold state of the respective ports is released 1: I/O hold state of the respective ports is activated

Table 13.8 IOHOLD Register Contents (2/2)

Bit Position	Bit Name	Function
1	IOHOLD1	This bit controls the IOHOLD function of the port groups P1 and P2 ports. 0: I/O hold state of the respective ports is released 1: I/O hold state of the respective ports is activated
0	IOHOLD0	This bit controls the IOHOLD function of the port group P0 ports. 0: I/O hold state of the respective ports is released 1: I/O hold state of the respective ports is activated

13.3 DEEPSTOP Mode

In DEEPSTOP mode, the clock and power supply of the Isolated-Area is stopped.
Some clock generators and Always-On-Area clock domains can optionally be stopped.

The transition procedure (example) to DEEPSTOP mode is shown below.

Preparation for DEEPSTOP

- Stop all of the modules, connected to a clock that will be stopped.
- Set the port buffer of the Isolated-Area into hold state via the IOHOLD register.
  Note: Unless the power of corresponding port buffers are kept supplied during DEEPSTOP mode, IOHOLD register must not be set to 1.
- Disable the interrupt handling by the CPU instruction “DI”.
- (In D1M1A and D1M1-V2)
  Set the interrupt control registers.
  - Clear the interrupt flag (ICxxx.RFxxx = 0).
  - Mask the interrupt of non-wake-up factor (ICxxx.MKxxx = 1).
  - Unmask the interrupt of wake-up factors (ICxxx.MKxxx = 0).
- Set the wake-up related registers.
  - Clear the wake-up factor flag (the WUFC0 register).
  - Mask the non-wake-up factors (the WUFMSK0 register).
  - Unmask the wake-up factors (the WUFMSK0 register).
- (Except in D1M1A and D1M1-V2)
  Change CPU clock to EMCLK
- (Except in D1M1A and D1M1-V2)
  Abort entering DEEPSTOP mode if interrupt requests’ flags to abort entering DEEPSTOP mode set.
- Set the clock stop mask and select the clock domains to be stopped and to continue operating. (Set by the CKSC_xxx_STPM.xxxxSTPMSK bit.)
- Designate each clock source for oscillation or for stopping. In addition, set the clock mask and select which clock source is to stop and which clock source is to continue operation. Refer to Section 13.1.4, Clock supply in DEEPSTOP mode for information about these clock generators and domains.
- Stop all active PLLs (set PLLkE.PLLkDISTRG = 1)
- Clear the RESF register.

Start of DEEPSTOP

Set the STBC0DISTRG bit in the STBC0PSC register to 1 to shift to DEEPSTOP mode.

End of DEEPSTOP

When a wake-up event is generated, the microcontroller returns from DEEPSTOP mode.

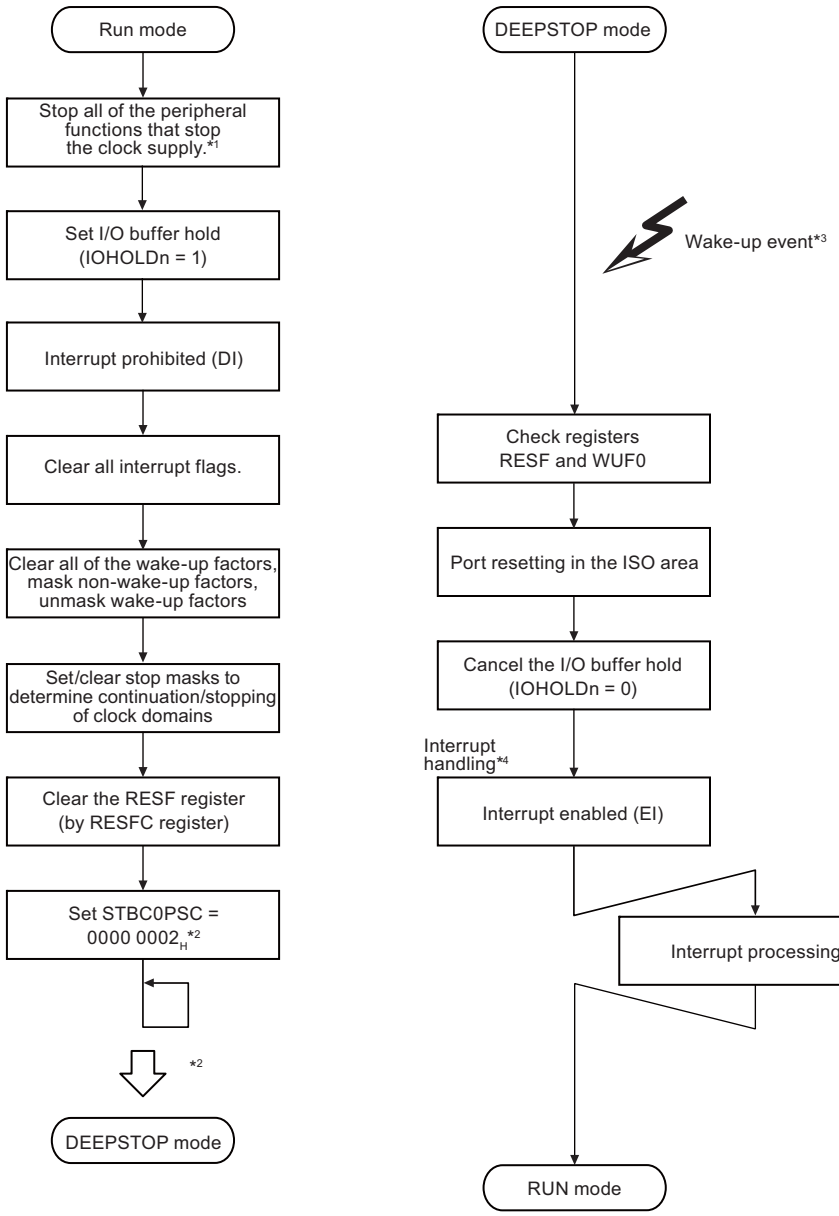
Wake-up handling

- Make sure that all BnVCC supplies are switched on.
- When returned from DEEPSTOP mode due to wake-up factor, the microcontroller starts the operation from the reset address.
- The wake-up factor is determined by the wake-up factor flag (WUF0).
- The ports in the Isolated-Area maintain the I/O buffer hold state.
- Release the I/O buffer hold state in the following order.

  1. Re-configure the modules and port functions.

  2. Release the I/O buffers from hold state by IOHOLD.IOHOLDn = 0

- To execute an interrupt of the wake-up factor after the wake-up, evaluate the information of wake-up factor flag by software and set the interrupt request flag in the interrupt control register. Then, when an interrupt is enabled by the CPU instruction “EI”, the generated wake-up interrupt is to be executed.



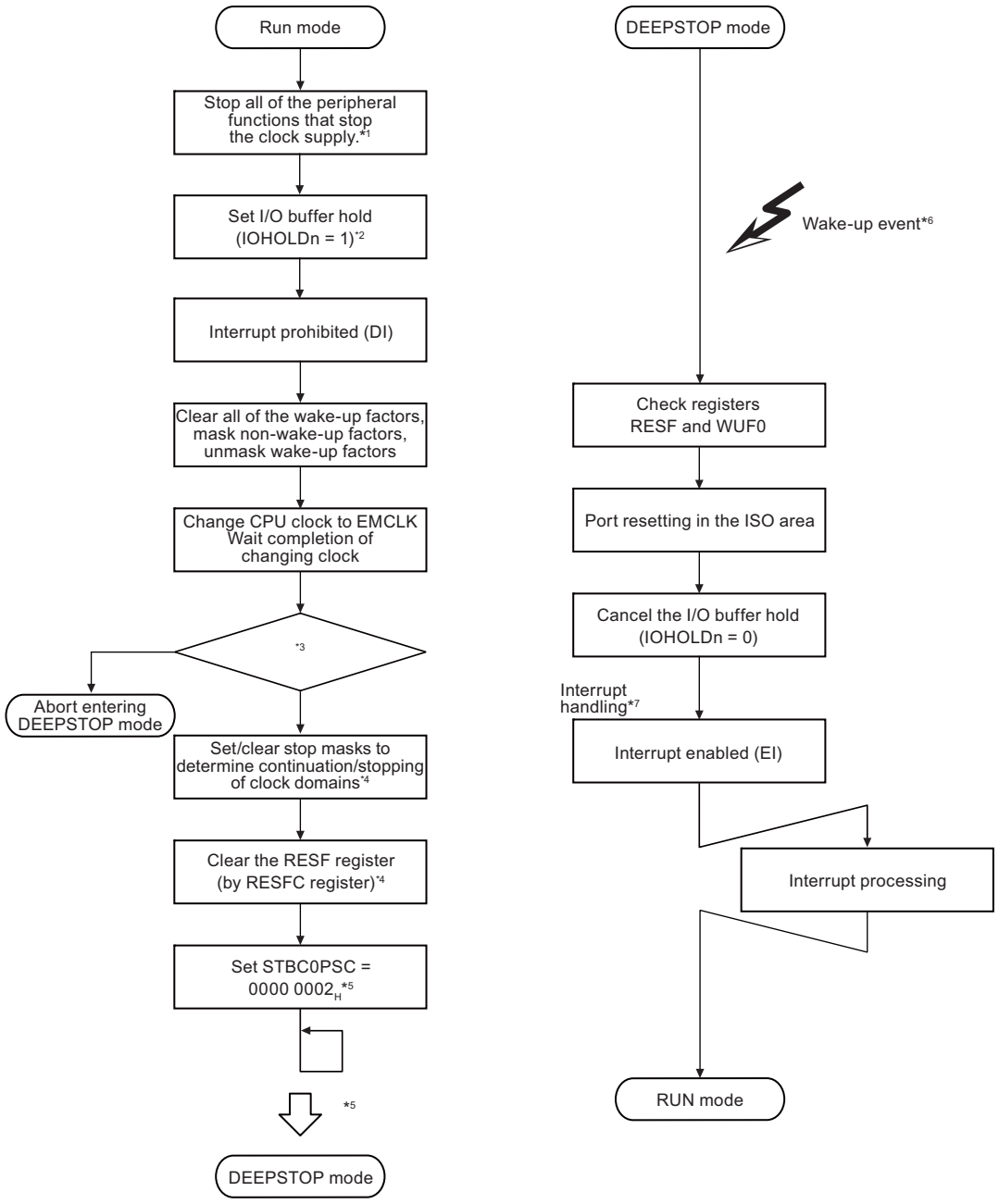
Note 1. When the operation of the peripheral function is stopped during operating due to the transition to the DEEPSTOP mode, the operation of the peripheral function may be incorrect. Before the transition to the DEEPSTOP mode, stop all of the peripheral functions that stop the clock supply.

Note 2. After setting STBC0PSC = 0000 0002_H, wait for the transition to the DEEPSTOP mode by the unconditional loop.

Note 3. The CPU starts the program from the reset vector after the generation of a wake-up event. The return from the DEEPSTOP mode by a reset can be checked by the RESF register. In addition, the generated wake-up event can be checked by the WUF0 register.

Note 4. This processing is optional. It is required to execute the interrupt handling after the wake-up.

Figure 13.2 Example of DEEPSTOP mode transition (D1M1A, D1M1-V2)



Note 1. When the operation of the peripheral function is stopped during operating due to the transition to the DEEPSTOP mode, the operation of the peripheral function may be incorrect. Before the transition to the DEEPSTOP mode, stop all of the peripheral functions that stop the clock supply.

Note 2. Unless the power of corresponding port buffers are kept supplied during DEEPSTOP mode, IOHOLD register must not be set to 1.

Note 3. Are interrupt requests' flags to abort entering DEEPSTOP mode set?

Note 4. These steps can execute before setting wake-up factors.

Note 5. After setting STBC0PSC = 0000 0002_H, wait for the transition to the DEEPSTOP mode by the unconditional loop.

Note 6. The CPU starts the program from the reset vector after the generation of a wake-up event. The return from the DEEPSTOP mode by a reset can be checked by the RESF register. In addition, the generated wake-up event can be checked by the WUF0 register.

Note 7. This processing is optional. It is required to execute the interrupt handling after the wake-up.

Figure 13.3 Example of DEEPSTOP mode transition (except for D1M1A, D1M1-V2)

13.4 Writing to the protected registers

The Stand-by Controller registers shown below are write-protected registers.

- STBC0PSC register
- IOHOLD register

The write-protected registers are protected against the illegal writing due to the error programming operation.

For details on the write-protected sequence, refer to Section 4, Write-Protected Registers.

Section 14 Bus Architecture

14.1 Bus Systems Clock Supply

The clock supply of the bus systems is shown in the following table.

Table 14.1 Bus Systems Clock Supply

Bus System	Clock for the Bus System
PBUS	C_ISO_PCLK
Cross-connects XC	C_ISO_XCCLK

14.2 Cross-connect systems

The cross-connect systems are multilayered, transaction based multi-master connections and allow parallel access to different targets (refer to Section 14.3.1, Transaction based busses for more information).

Several cross-connect systems establish the data transfer paths between the bus masters and the slaves.

In order to cope with different demands in terms of data transfer bandwidth and real-time constrains different groups of masters and slaves are connected via different cross-connect systems:

- “Streaming data access masters” with high bandwidth demand of the video channels is centralized via the XC2 cross-connect and passed over the Sprite Engine to the XC0 cross-connect.
- “Random data access masters”, that mostly access data randomly or have no priority on high bandwidth (CPU Subsystem, Ethernet AVB MAC, Medial Local Bus Interface, JPEG Coding Unit), utilize the XC1 cross-connect, that hosts also the on-chip memories (CPU Subsystem’s local RAM and Code Flash, Retention RAM, Video RAMs) and the interface to external serial flash memory.
- SDRAM interfaces
  The bandwidth demand to external SDRAM varies with the device type:
  - D1M1H and D1M1A: an interface to single data rate SDRAM (SDR-SDRAM) is connected to the XC1 cross-connect
  - D1M2(H): can address double data rate SDRAM (DDR2-SDRAM) and provides several 128-bit wide ports to supply high data rates for the video and graphics masters.
- The connection of the 2D Graphics Processing Unit (GPU2D), that acts also as a bus master, depends on the device:
  - D1M1(H), D1M1-V2 and D1M1A: GPU2D is also a XC1 master
  - D1M2(H): GPU2D can act as a master to the XC1 cross-connect for accessing on-chip memories, but has also a dedicated 128-bit port to the DDR2-SDRAM interface, that boosts the drawing performance in conjunction with the external DDR2-SDRAM.

Multilayered topology

Since the cross-connect systems represent a multilayer topology, masters accessing different slaves do not interfere.

Access of masters accessing the same slave concurrently is regulated by an arbiter. Refer to Section 14.3, Arbitration, bandwidth and latencies for details.

14.2.1 XC0 and XC1 cross-connects and XC Guards

The access to all XC0 and XC1 cross-connect slaves is guarded by the XC Guards. These can be configured to permit or prohibit write or read access to the slave by the various cross-connect masters.

The XC Guards XCGn identify a master by its master ID MSTIDk. The XC0 and XC1 cross-connects form a single master ID domain.
The IDs of all XC0 and XC1 masters are given in the following table.

Table 14.2 XC0 and XC1 cross-connect master IDs

Cross-connect	Master		Master ID k	Symbol
XC1	Debug unit		0	MSTID0
	CPU		1	MSTID1
	DMA Controller		2	MSTID2
	2D Graphics Processing Unit (GPU2D)		4	MSTID4
	JPEG Codec Unit A (JCUA)		5	MSTID5
	Ethernet AVB MAC (ETNB)		6	MSTID6
	Media Local Bus I/F (MLBB) or NAND Flash Controller (NFMA)		7	MSTID7
XC0 for D1L2(H), D1M1(H), D1M2(H)	Sprite Engine (SPEA)	RLE Units	8	MSTID8
		Sprite Unit 0	9	MSTID9
		Sprite Unit 1	10	MSTID10
		Sprite Unit 2	11	MSTID11
XC0 for D1M1-V2, D1M1A	Sprite Engine (SPEA)	RLE Unit 0, Sprite Unit 3	8	MSTID8
		RLE Unit 1, Sprite Unit 0	9	MSTID9
		RLE Unit 2, Sprite Unit 1	10	MSTID10
		RLE Unit 3, Sprite Unit 2	11	MSTID11

The XC0 cross-connect has two layers XC0_0 and XC0_1.

NOTE

The D1L1 devices do not have an XC0 cross-connect.

14.2.1.1 List of XC1 QoS Registers (D1M1(H), D1M1-V2, D1M1A only)

Table 14.3 List of XC1 QoS registers

Master Name XXX	Master number n	Description
CPU0	0	Debug master and DMA
CPU1	1	CPU master
VACC0	2	Cross-connect XC0_0
VACC1	3	Cross-connect XC0_1
JCUA	4	JCUA master
GPU2D	5	GPU2D master
ETNB	6	ETNB master

Base address

The register addresses in the table below are given as offset addresses to the

<base_addr> = FFCD 2300_H.

Table 14.4 List of XC1 QoS registers

Register name	Symbol	Address
XC1 QoS mode	QOS_MODE	<base_addr>
XC1 XXX QoS cycle count max value	XXX_MAX_COUNT	<base_addr> + 4_H + $10_H \times n$
XC1 XXX QoS read access permit count designation	XXX_MAX_RACC	<base_addr> + 8_H + $10_H \times n$
XC1 XXX QoS write access permit count designation	XXX_MAX_WACC	<base_addr> + C_H + $10_H \times n$

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> is defined as “SD_ISO_VDD" for the registers in the table above.

<Symbol> is defined in the above table

(1) QOS_MODE – XC1 QoS mode register (D1M1(H), D1M1-V2, D1M1A only)

This register is used to select QoS mode of XC1.

CAUTION

This register must only be changed while XC1 is not used.

Access: This register can be read/written in 32-bit units.

Address: <base_addr>

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	QOS_MODE[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 14.5 QOS_MODE register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, these bits are always read as 0. When writing, always write 0.
2 to 0	QOS_ MODE[2:0]	XC QoS mode selection 000_B : Qos off 010_B : Interval mode 011_B: Threshold mode others : Setting prohibited

NOTE

- XXX indicates the master name
- n indicates the master number

(2) XXX_MAX_COUNT – XC1 XXX QoS cycle count max value register (D1M1(H), D1M1-V2, D1M1A only)

This register sets the initial value of the QoS cycle counter.

CAUTION

This register must only be changed while XC1 is not used.

Access: This register can be read/written in 32-bit units.

Address: <base_addr> + 4_H + 10_H × n

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	XXX_MAX_COUNT[8:0]								
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.6 XXX_MAX_COUNT register contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When read, these bits are always read as 0. When writing, always write 0.
8 to 0	XXX_MAX_ COUNT[8:0]	QoS cycle counter initial value for XXX master. This register is enabled in interval mode and threshold mode. In other modes, setting value of this register is ignored. Interval mode: This register is set the number of interval cycles after accepting one access. Threshold mode : This register is set to the initial value of the cycle counter. XXX_MAX_RACC / XXX_MAX_WACC times of read / write access are permitted within the period that is set in this register.

NOTE

- XXX indicates the master name
- n indicates the master number

(3) XXX_MAX_RACC – XC1 XXX QoS read access permit count designation register (D1M1(H), D1M1-V2, D1M1A only)

This register sets the number of read access that are permitted in threshold mode.

CAUTION

This register must only be changed while XC1 is not used.

Access: This register can be read/written in 32-bit units.

Address: <base_addr> + 8_H + 10_H × n

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	XXX_MAX_RACC[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 14.7 XXX_MAX_RACC register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, these bits are always read as 0. When writing, always write 0.
2 to 0	XXX_MAX_ RACC[2:0]	The number of read accesses that are permitted in threshold mode This register is enabled in threshold mode. In other modes, setting value of this register is ignored. This register sets the number of read accesses that can be accepted within the period specified in the XXX_MAX_COUNT.

NOTE

- XXX indicates the master name
- n indicates the master number

(4) XXX_MAX_WACC – XC1 XXX QoS write access permit count designation register (D1M1(H), D1M1-V2, D1M1A only)

This register sets the number of write access that are permitted in threshold mode.

CAUTION

This register must only be changed while XC1 is not used.

Access: This register can be read/written in 32-bit units.

Address: <base_addr> + C_H + $10_H \times n$

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	XXX_MAX_WACC[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 14.8 XXX_MAX_WACC register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, these bits are always read as 0. When writing, always write 0.
2 to 0	XXX_MAX_ WACC[2:0]	The number of write accesses that are permitted in threshold mode This register is enabled in threshold mode. In other modes, setting value of this register is ignored. This register sets the number of write accesses that can be accepted within the period specified in the XXX_MAX_COUNT.

NOTE

- XXX indicates the master name
- n indicates the master number

14.2.2 XC2 cross-connect

The XC2 cross-connect forms an own master ID domain. Accesses of the XC2 masters are performed via the XC0 master ports of the Sprite Engine. Thus their XC2 master IDs are not visible on the XC0 and XC1 cross-connects.

The XC2 cross-connect has four layers XC2_0 to XC2_3.

NOTE

The D1L1 devices do not have an XC2 cross-connect.

14.2.3 Cross-connect details

This section gives details about each cross-connect.

NOTES

1. The arrows in the diagrams indicate the data transfer direction.
2. The assignment of the XC Guards XCGn to the cross-connect slaves are given in Table 14.40, Register base addresses <XCGn_base> in Section 14.6.5, Cross-connect Guards (XCG).
3. The indicators in the connection matrix table, provided for each device have the following meaning:

R:	The master can read from the slave.
W:	The master can write to the slave.
R/W;	The master can read from and write to the slave.
–:	The master has no access to the slave.

14.2.3.1 D1L1 bus architecture

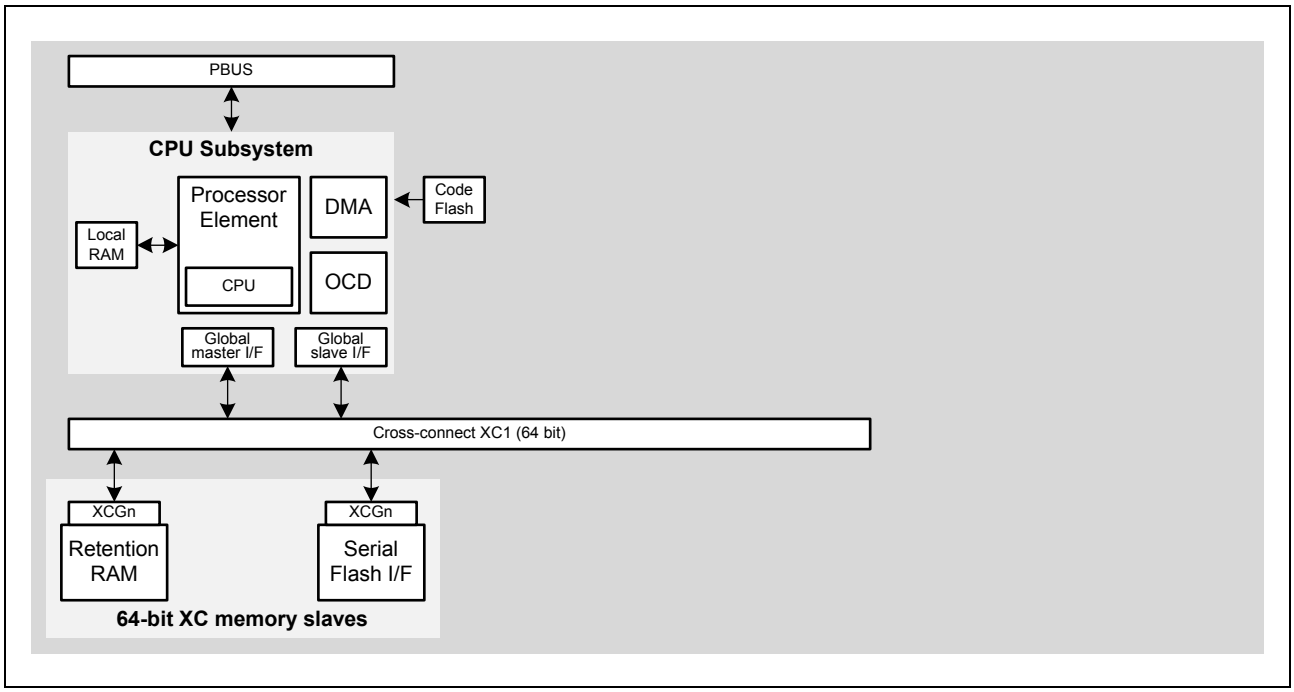
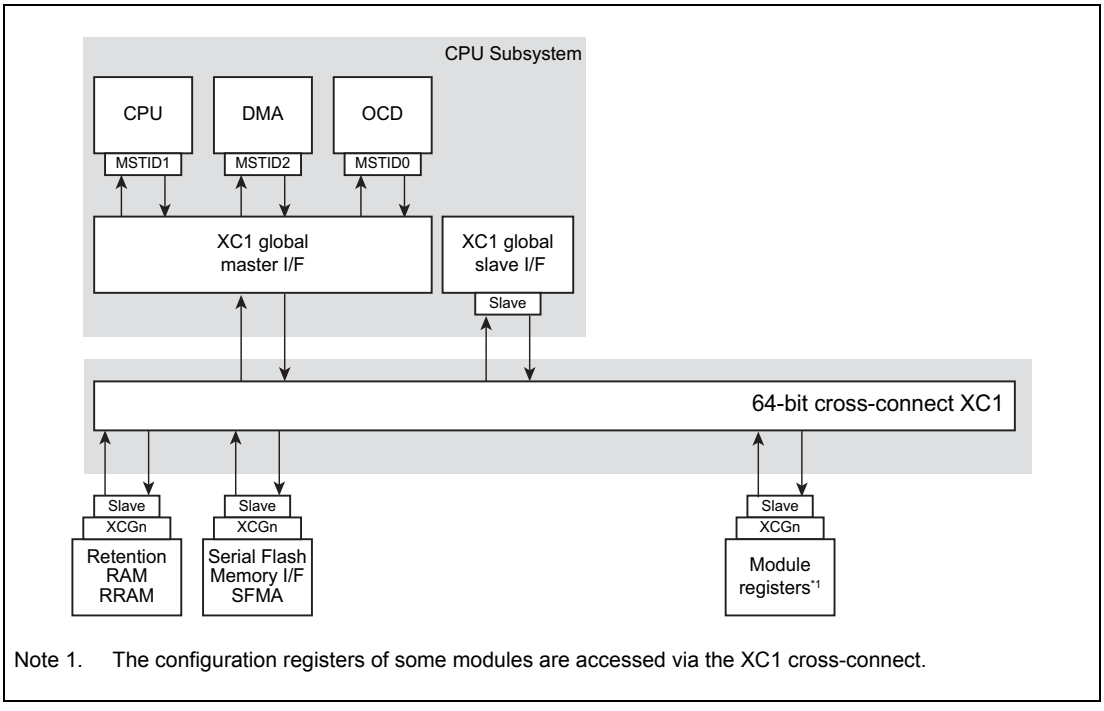


Figure 14.1 D1L1 bus architecture overview

(a) D1L1 cross-connect XC1



Note 1. The configuration registers of some modules are accessed via the XC1 cross-connect.

Figure 14.2 D1L1 cross-connect XC1

(b) D1L1 connection matrix

Table 14.9 D1L1 connection matrix

	Masters		
	CPU Subsystem		
	CPU I/F	Global I/F	
	CPU	DMA	OCD
Slaves	MSTID1	MSTID2	MSTID0
CPU Subsystem global I/F*1	–	–	–
RRAM	R/W	R/W	R/W
SFMA	R/W*2	R/W*2	R/W*2
Module registers	R/W	R/W	R/W

Note 1. Access to LRAM and Code Flash

Note 2. Write access to SFMA registers only.

14.2.3.2 D1L2(H) bus architecture

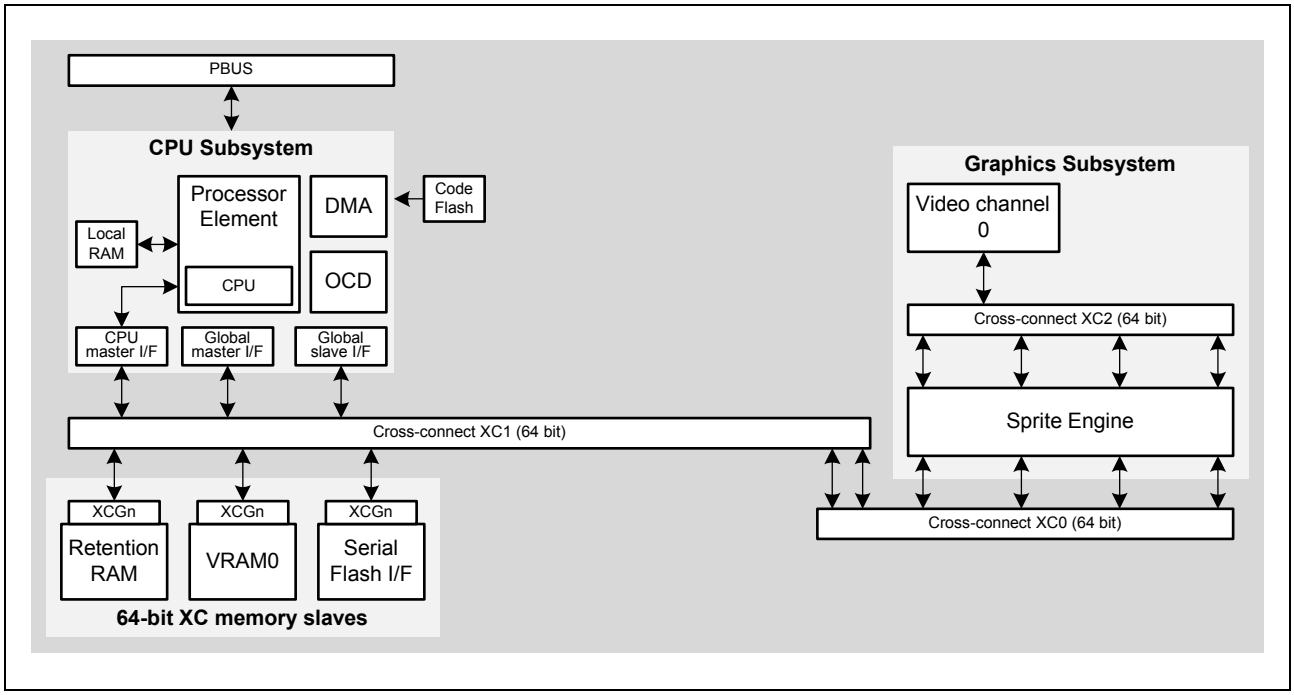


Figure 14.3 D1L2(H) bus architecture overview

(a) D1L2(H) cross-connect XC0

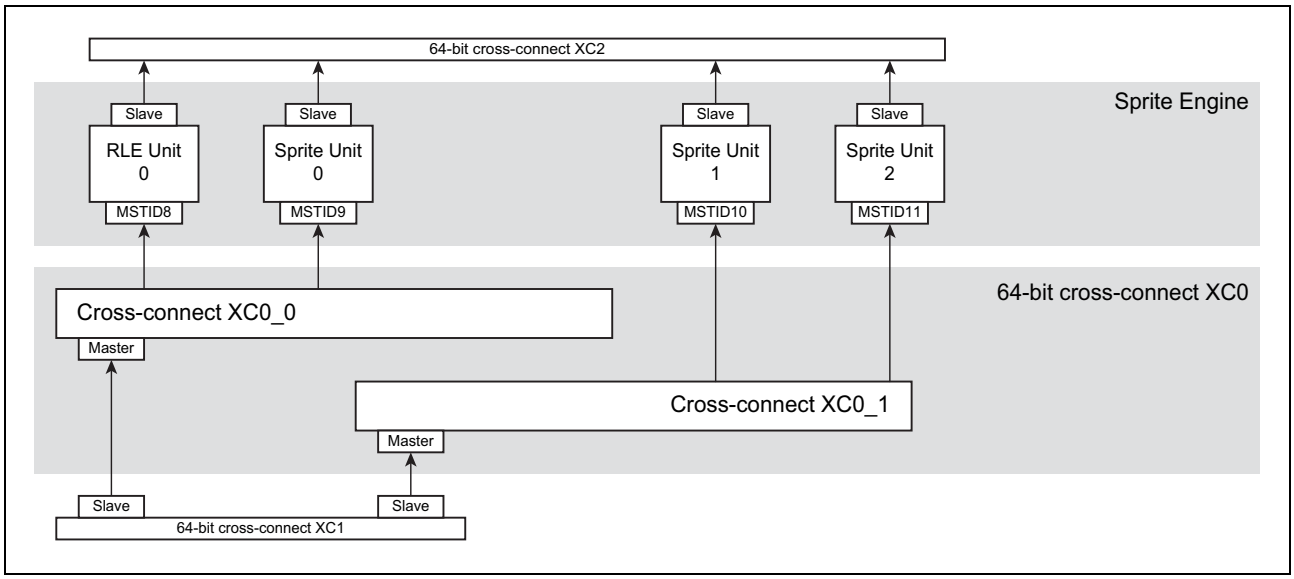
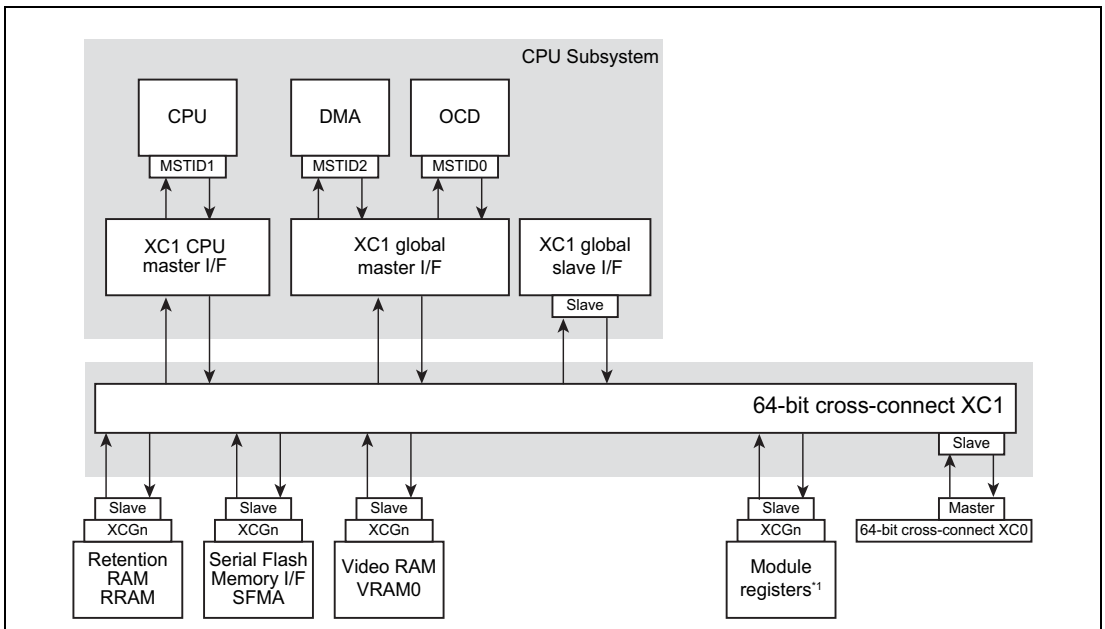


Figure 14.4 D1L2(H) cross-connect XC0

(b) D1L2(H) cross-connect XC1



Note 1. The configuration registers of some modules are accessed via the XC1 cross-connect.

Figure 14.5 D1L2(H) cross-connect XC1

(c) D1L2(H) cross-connect XC2

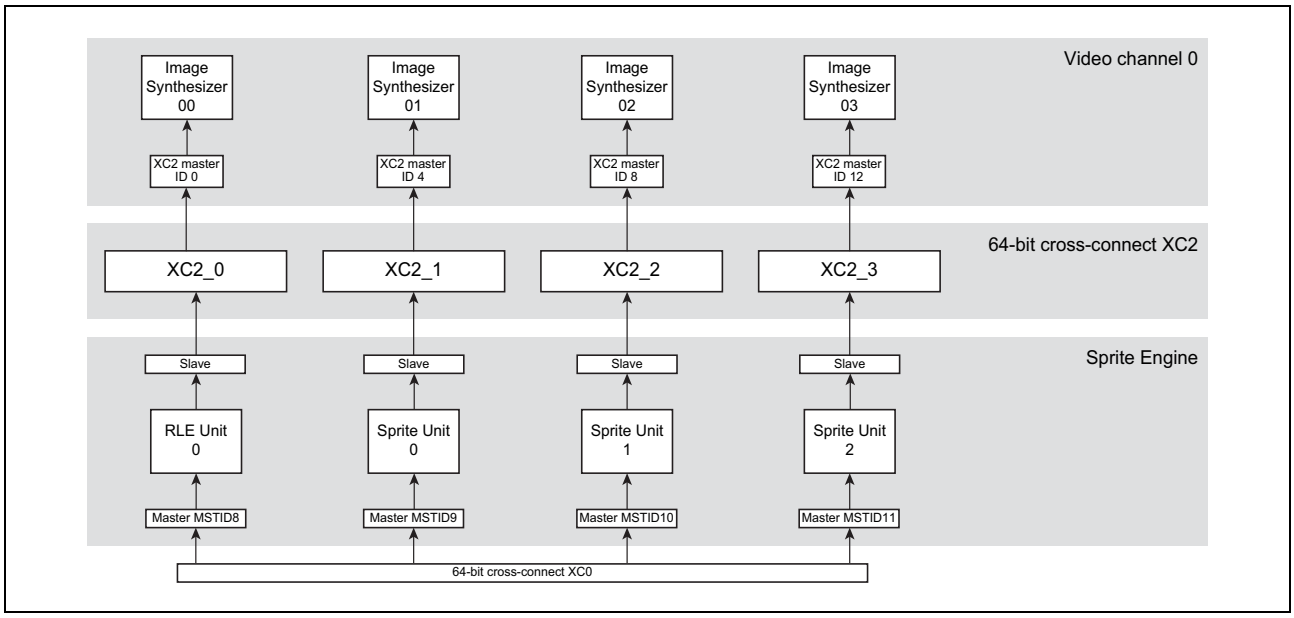


Figure 14.6 D1L2(H) cross-connect XC2

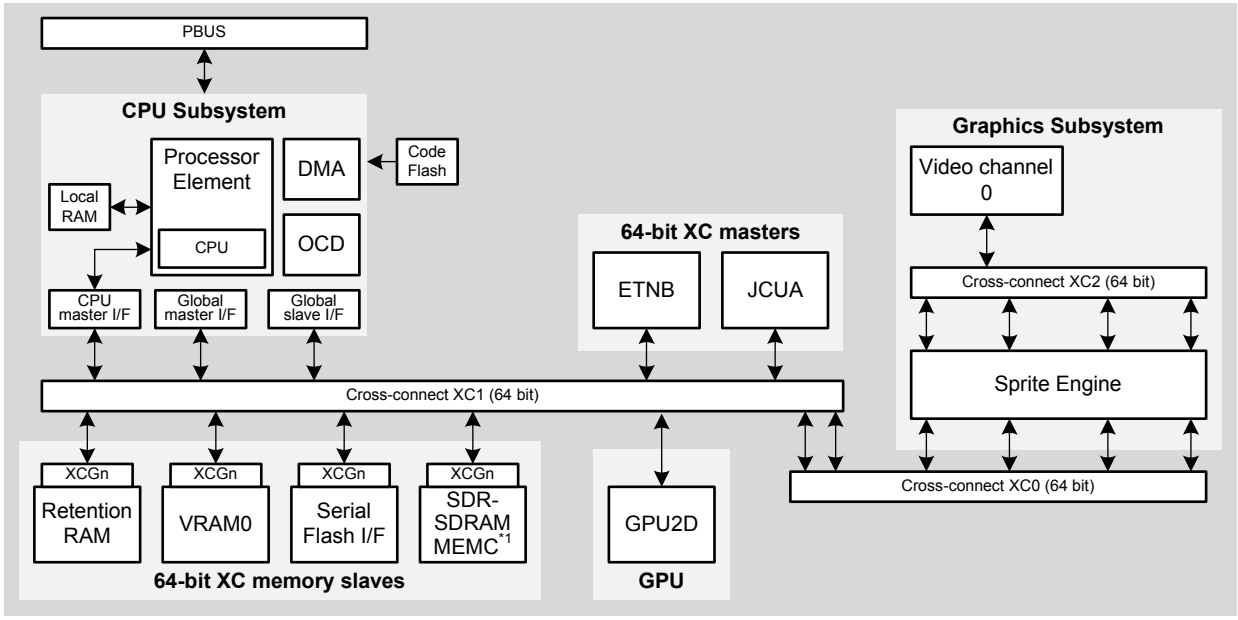
(d) D1L2(H) connection matrix

Table 14.10 D1L2(H) connection matrix

	Masters						
	CPU Subsystem			Sprite Engine			
	CPU I/F	Global I/F					
	CPU	DMA	OCD	RLE Unit*2	Sprite Unit 0*3	Sprite Unit 1*4	Sprite Unit 2*5
Slaves	**MSTID1**	**MSTID2**	**MSTID0**	**MSTID8**	**MSTID9**	**MSTID10**	**MSTID11**
CPU Subsystem global I/F*1	–	–	–	R	R	R	R
RRAM	R/W	R/W	R/W	–	–	–	–
SFMA	R/W*6	R/W*6	R/W*6	R	R	R	R
VRAM0	R/W	R/W	R/W	R	R	R	R
Module registers	R/W	R/W	R/W	–	–	–	–

Note 1. Access to LRAM and Code Flash
Note 2. Image Synthesizer 00 (R) via XC2 cross-connect
Note 3. Image Synthesizer 01 (R) via XC2 cross-connect
Note 4. Image Synthesizer 02 (R) via XC2 cross-connect
Note 5. Image Synthesizer 03 (R) via XC2 cross-connect
Note 6. Write access to SFMA registers only.

14.2.3.3 D1M1(H) bus architecture



Note 1. D1M1H only

Figure 14.7 D1M1(H) bus architecture overview

(a) D1M1(H) cross-connect XC0

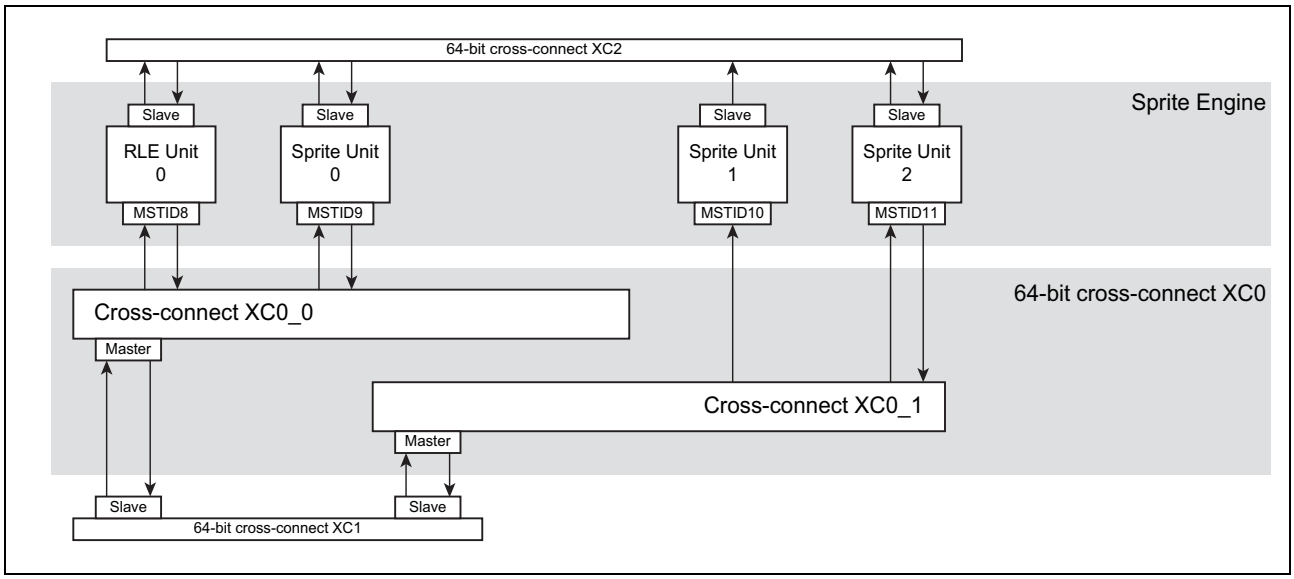
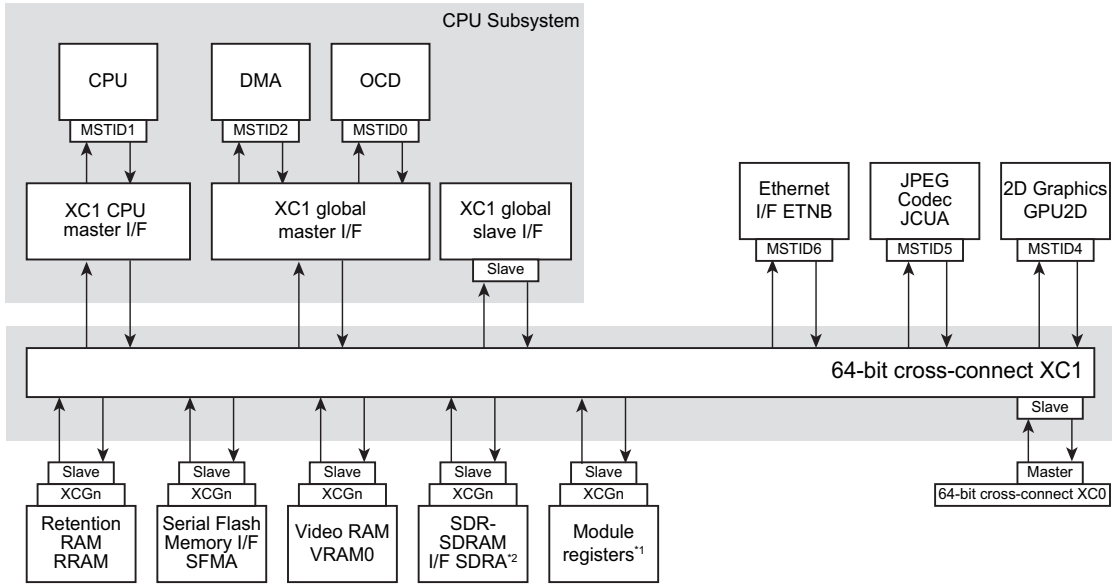


Figure 14.8 D1M1(H) cross-connect XC0

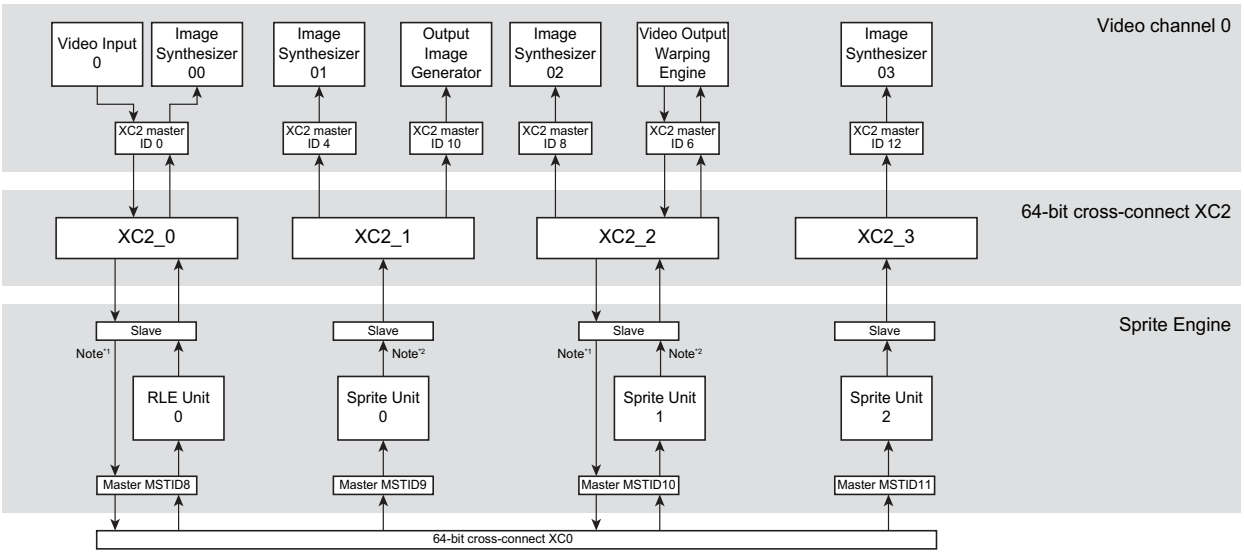
(b) D1M1(H) cross-connect XC1



Note 1. The configuration registers of some modules are accessed via the XC1 cross-connect.
Note 2. D1M1H only

Figure 14.9 D1M1(H) cross-connect XC1

(c) D1M1(H) cross-connect XC2



Note 1. All write accesses of the video channel masters are passed through the Sprite Engine to the XC0 cross-connect.
Note 2. Read accesses of the Warping Engine via Sprite Unit 1 and the Output Image Generator via Sprite Unit 0 must not read from addresses, covered by the Sprite Engine's virtual frame (refer to Section 43, Sprite Engine (SPEA) for details about the virtual frame). This way the Sprite Unit passes the master's read access through to the XC0 cross-connect.

Figure 14.10 D1M1(H) cross-connect XC2

(d) D1M1(H) connection matrix

Table 14.11 D1M1(H) connection matrix

	Masters									
	CPU Subsystem						Sprite Engine			
	CPU I/F	Global I/F								
	CPU	DMA	OCD	GPU2D	ETNB	JCUA	RLE Unit*2	Sprite Unit 0*3	Sprite Unit 1*4	Sprite Unit 2*5
Slaves	MSTID1	MSTID2	MSTID0	MSTID4	MSTID6	MSTID5	MSTID8	MSTID9	MSTID 10	MSTID 11
CPU Subsystem global I/F*1	–	–	–	R/W*6	R/W*6	R/W	R/W	R/W	R/W*6	R/W
RRAM	R/W	R/W	R/W	–	–	–	–	–	–	–
SFMA	R/W*7	R/W*7	R/W*7	R/W*6	–	R/W*6	R/W*6	R/W*6	R/W*6	R/W*6
VRAM0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SDRA	R/W	R/W	R/W	R/W*8	R/W*8	R/W*8	R/W*8	R/W*8	R/W*8	R/W*8
Module registers	R/W	R/W	R/W	–	–	–	–	–	–	–

Note 1. Access to LRAM and Code Flash
Note 2. Video Input 0 (W), Image Synthesizer 00 (R) via XC2 cross-connect
Note 3. Image Synthesizer 01 (R), Output Image Generator (R) via XC2 cross-connect
Note 4. Image Synthesizer 02 (R), Video Output Warping Engine (R/W) via XC2 cross-connect
Note 5. Image Synthesizer 03 (R) via XC2 cross-connect
Note 6. Write path must not be used.
Note 7. Write access to SFMA registers only.
Note 8. Write access to SDRA registers is prohibited.

14.2.3.4 D1M1-V2 bus architecture

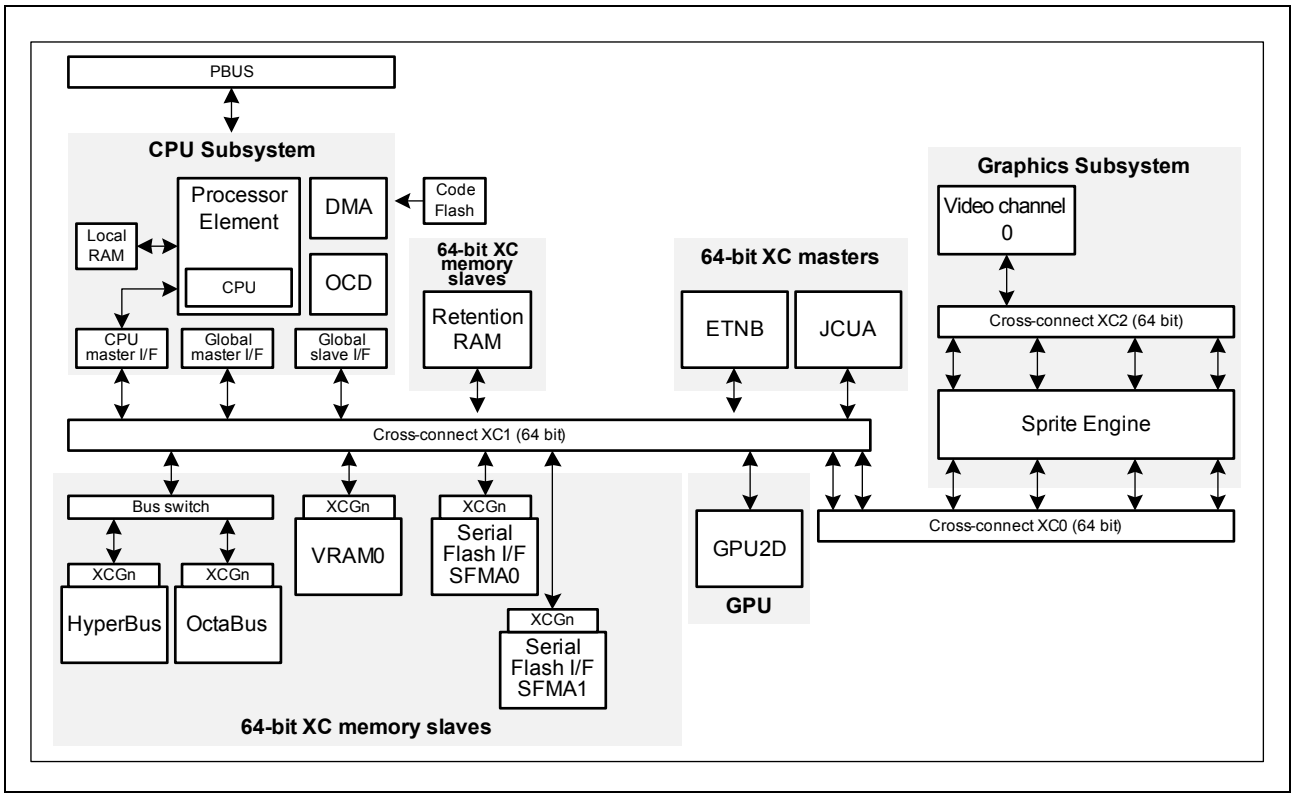


Figure 14.11 D1M1-V2 bus architecture overview

(a) D1M1-V2 cross-connect XC0

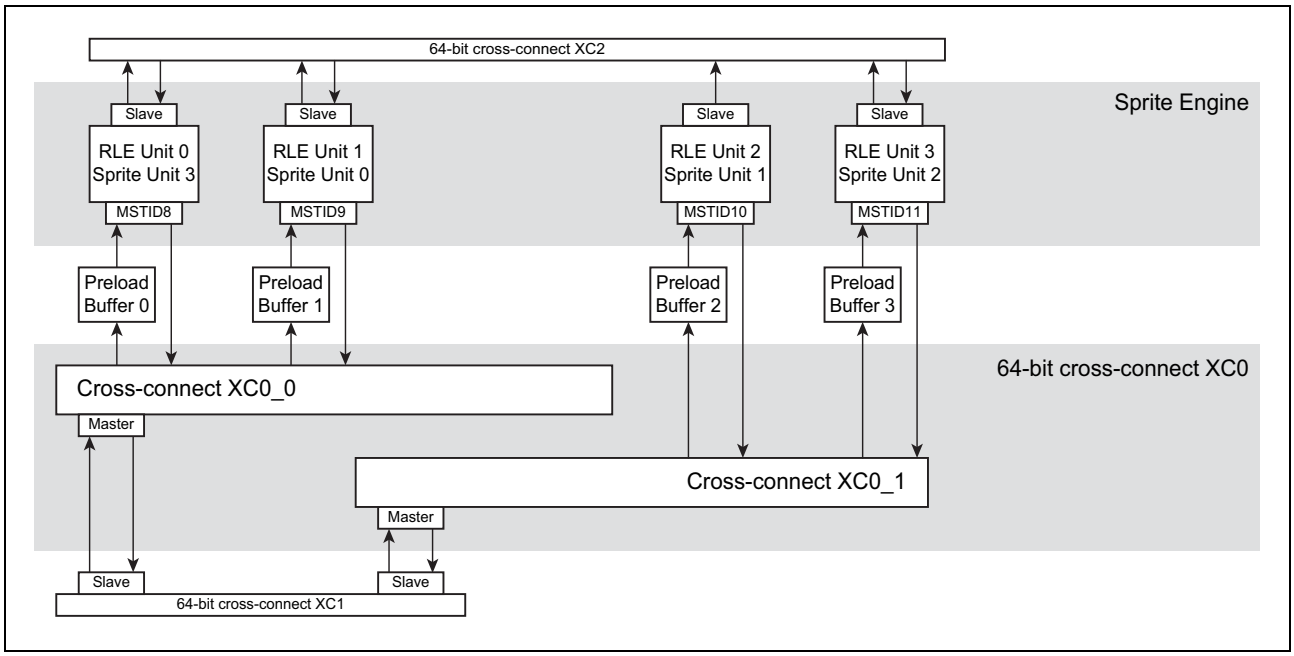
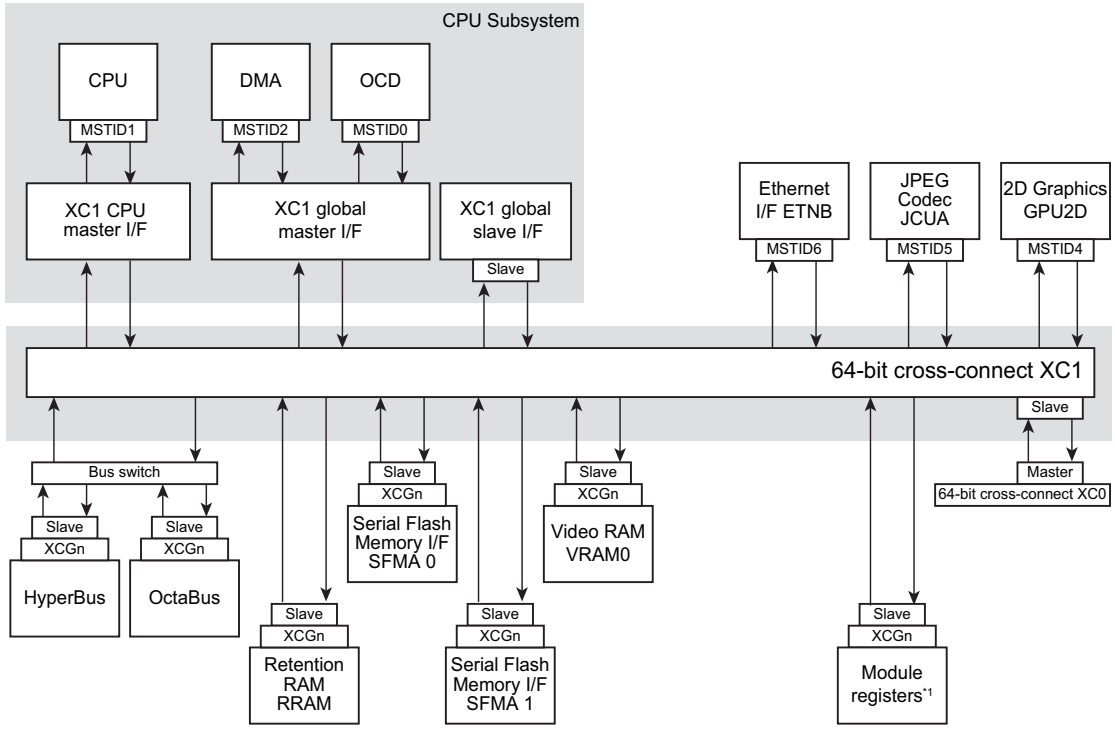


Figure 14.12 D1M1-V2 cross-connect XC0

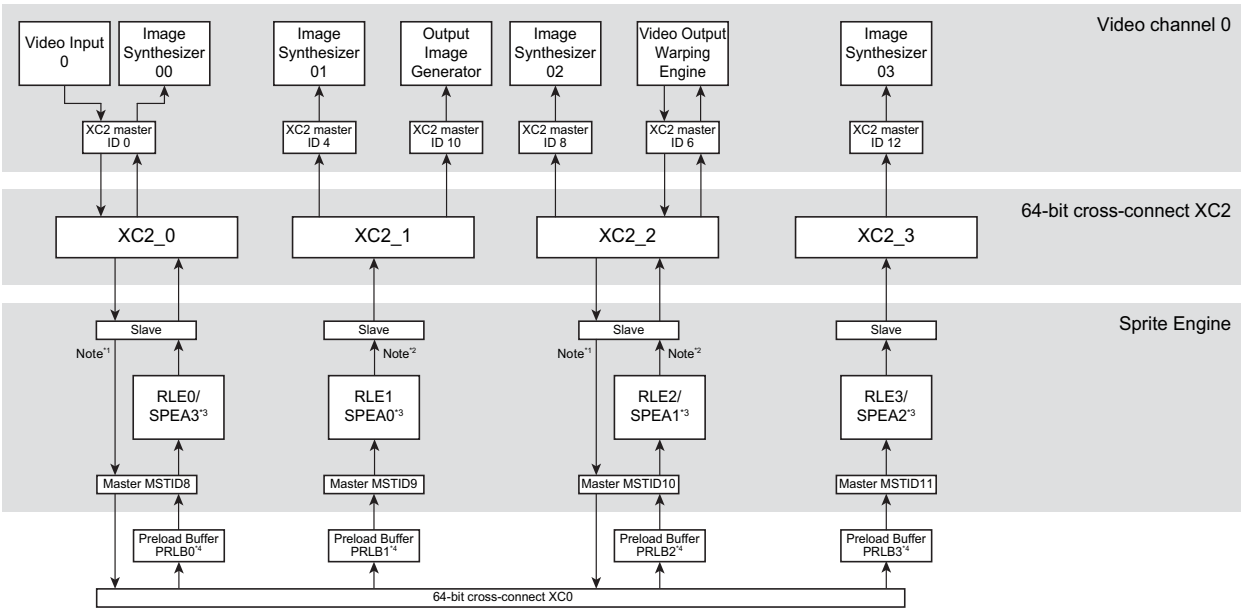
(b) D1M1-V2 cross-connect XC1



Note 1. The configuration registers of some modules are accessed via the XC1 cross-connect.

Figure 14.13 D1M1-V2 cross-connect XC1

(c) D1M1-V2 cross-connect XC2



Note 1. All write accesses of the video channel masters are passed through the Sprite Engine to the XC0 cross-connect.

Note 2. Read accesses of the Warping Engine via Sprite Unit 0 and the Output Image Generator via Sprite Unit 1 must not read from addresses, covered by the Sprite Engine's virtual frame (refer to Section 43, Sprite Engine (SPEA) for details about the virtual frame). This way the Sprite Unit passes the master's read access through to the XC0 cross-connect.

Note 3. Each unit can be selected to be RLE or Sprite Unit. For details refer to Section 43, Sprite Engine (SPEA).

Note 4. For details about the Preload Buffers refer to Section 14.8, Preload Buffer (PRLBn) (D1M1-V2, D1M1A only)

Figure 14.14 D1M1-V2 cross-connect XC2

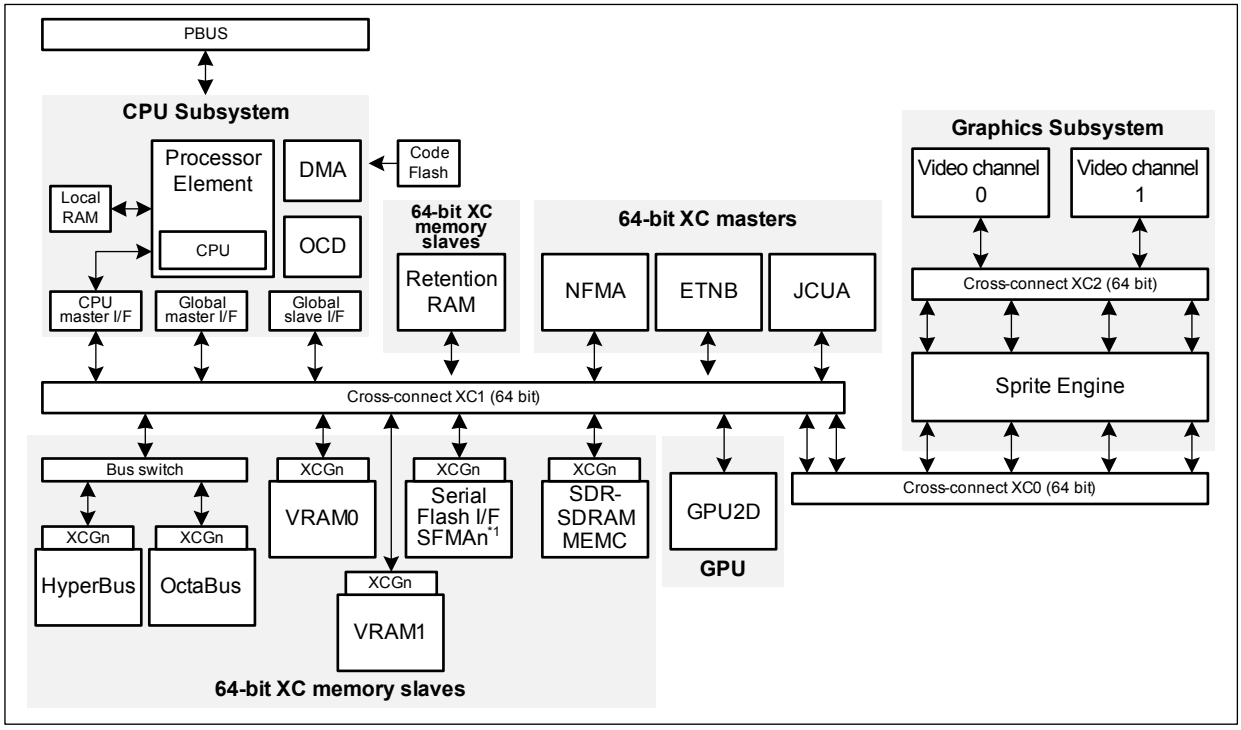
(d) D1M1-V2 connection matrix

Table 14.12 D1M1-V2 connection matrix

	Masters									
	CPU Subsystem						Sprite Engine			
	CPU I/F	Global I/F								
	CPU	DMA	OCD	GPU2D	ETNB	JCUA	RLE0/ SPEA3*2	RLE1/ SPEA0*3	RLE2/ SPEA1*4	RLE3/ SPEA2*5
Slaves	MSTID 1	MSTID 2	MSTID 0	MSTID 4	MSTID 6	MSTID 5	MSTID 8	MSTID 9	MSTID 10	MSTID 11
CPU Subsystem global I/F*1	–	–	–	R/W*6	R/W*6	R/W	R/W	R/W	R/W*6	R/W
RRAM	R/W	R/W	R/W	–	–	–	–	–	–	–
SFMA0	R/W*7	R/W*7	R/W*7	R/W*6	–	R/W*6	R/W*6	R/W*6	R/W*6	R/W*6
SFMA1	R/W*7	R/W*7	R/W*7	R/W*6	–	R/W*6	R/W*6	R/W*6	R/W*6	R/W*6
VRAM0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
HyperBus	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
OctaBus	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Module registers	R/W	R/W	R/W	–	–	–	–	–	–	–

Note 1. Access to LRAM and Code Flash
Note 2. Video Input 0 (W), Image Synthesizer 00 (R) via XC2 cross-connect
Note 3. Image Synthesizer 01 (R), Output Image Generator (R) via XC2 cross-connect
Note 4. Image Synthesizer 02 (R), Video Output Warping Engine (R/W) via XC2 cross-connect
Note 5. Image Synthesizer 03 (R) via XC2 cross-connect
Note 6. Write path must not be used.
Note 7. Write access to SFMA registers only.

14.2.3.5 D1M1A bus architecture



Note 1. n = 0 to 2

Figure 14.15 D1M1A bus architecture overview

(a) D1M1A cross-connect XC0

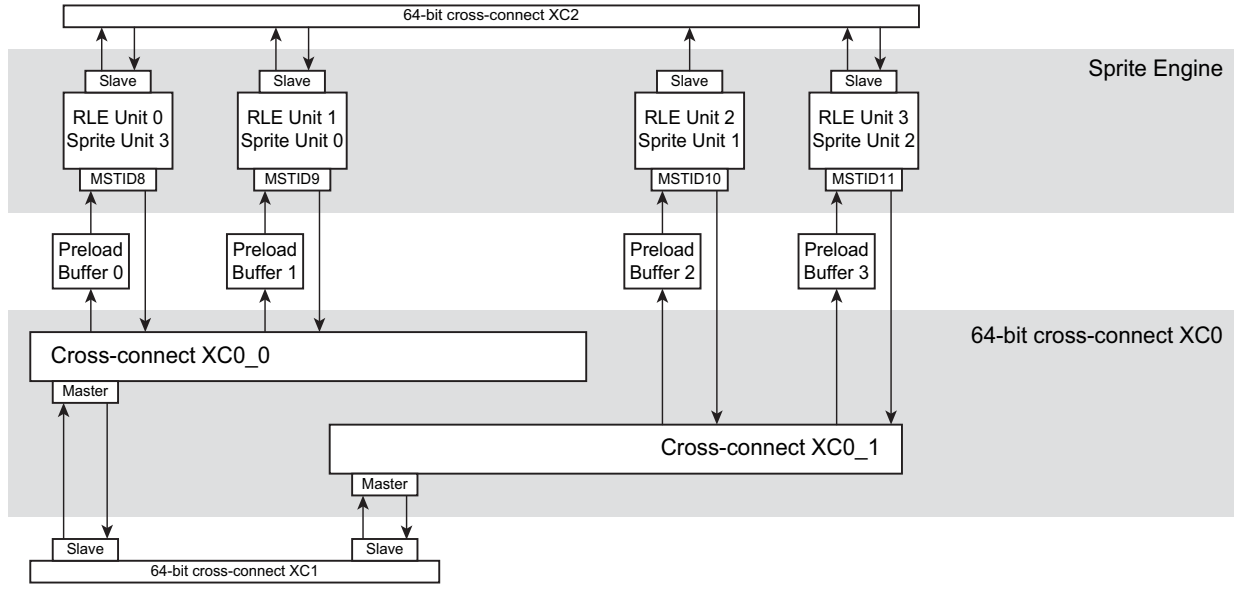
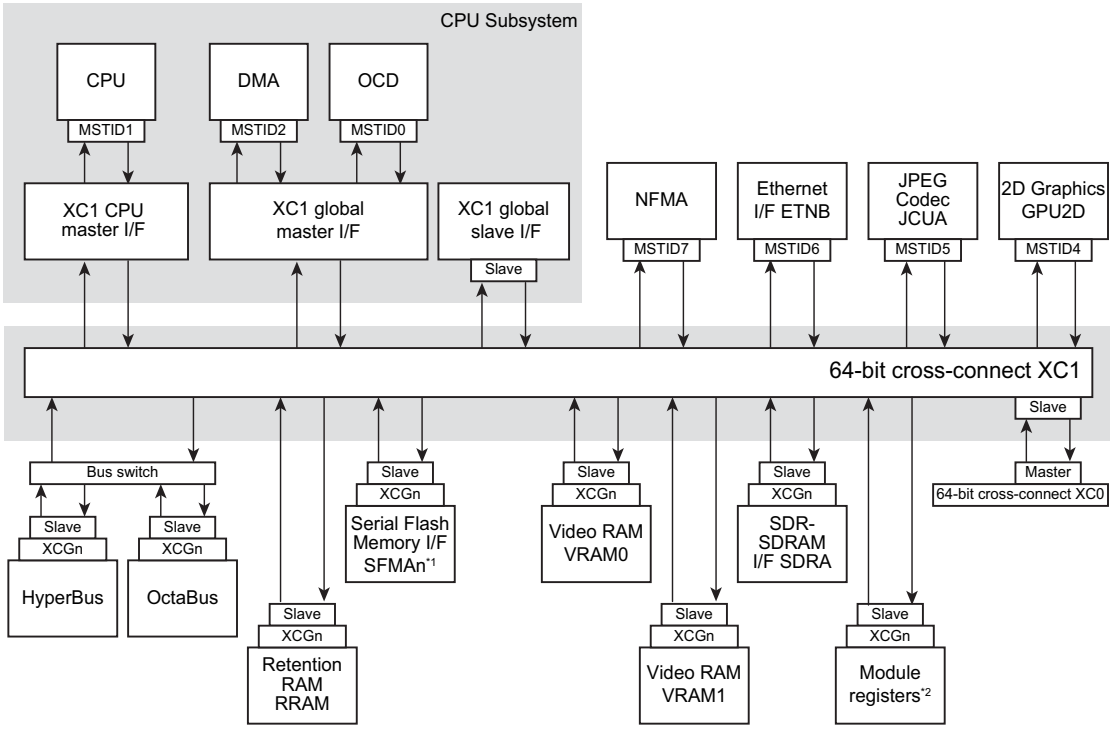


Figure 14.16 D1M1A cross-connect XC0

(b) D1M1A cross-connect XC1

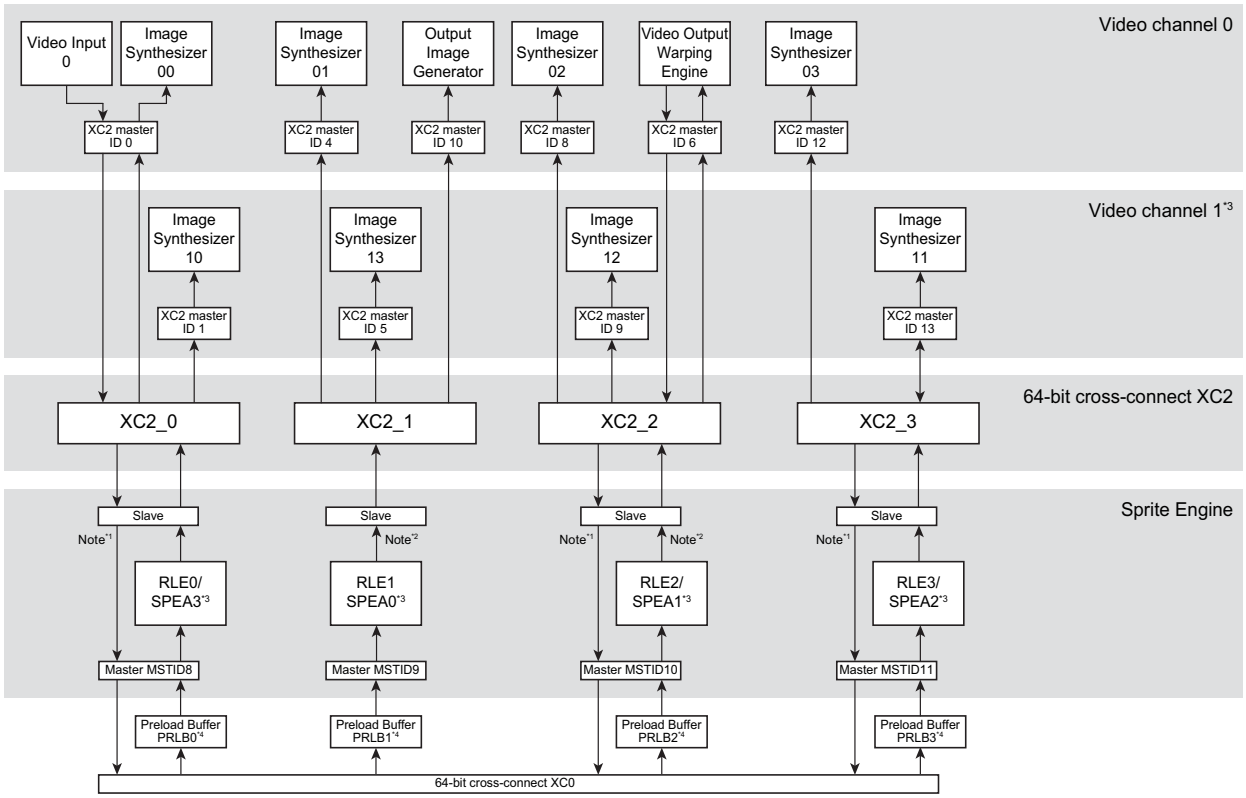


Note 1. n = 0 to 2

Note 2. The configuration registers of some modules are accessed via the XC1 cross-connect.

Figure 14.17 D1M1A cross-connect XC1

(c) D1M1A cross-connect XC2



Note 1. All write accesses of the video channel masters are passed through the Sprite Engine to the XC0 cross-connect.

Note 2. Read accesses of the Warping Engine via Sprite Unit 0 and the Output Image Generator via Sprite Unit 1 must not read from addresses, covered by the Sprite Engine's virtual frame (refer to Section 43, Sprite Engine (SPEA) for details about the virtual frame). This way the Sprite Unit passes the master's read access through to the XC0 cross-connect.

Note 3. Each unit can be selected to be RLE or Sprite Unit. For details refer to Section 43, Sprite Engine (SPEA).

Note 4. For details about the Preload Buffers refer to Section 14.8, Preload Buffer (PRLBn) (D1M1-V2, D1M1A only)

Figure 14.18 D1M1A cross-connect XC2

(d) D1M1A connection matrix

Table 14.13 D1M1A connection matrix

	Masters										
	CPU Subsystem							Sprite Engine			
	CPU I/F	Global I/F									
	CPU	DMA	OCD	GPU2D	ETNB	JCUA	NFMA	RLE0/ SPEA3 *2	RLE1/ SPEA0 *3	RLE2/ SPEA1 *4	RLE3/ SPEA2 *5
Slaves	MSTID 1	MSTID 2	MSTID 0	MSTID 4	MSTID 6	MSTID 5	MSTID 7	MSTID 8	MSTID 9	MSTID 10	MSTID 11
CPU Subsystem global I/F*1	–	–	–	R/W*6	R/W*6	R/W	R/W	R/W	R/W	R/W*6	R/W
RRAM	R/W	R/W	R/W	–	–	–	–	–	–	–	–
SFMA0	R/W*7	R/W*7	R/W*7	R/W*6	–	R/W*6	R/W*6	R/W*6	R/W*6	R/W*6	R/W*6
SFMA1	R/W*7	R/W*7	R/W*7	R/W*6	–	R/W*6	R/W*6	R/W*6	R/W*6	R/W*6	R/W*6
SFMA2	R/W*7	R/W*7	R/W*7	R/W*6	–	R/W*6	R/W*6	R/W*6	R/W*6	R/W*6	R/W*6
VRAM0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
VRAM1	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SDRA	R/W	R/W	R/W	R/W*8	R/W*8	R/W*8	R/W*8	R/W*8	R/W*8	R/W*8	R/W*8
HyperBus	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
OctaBus	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Module registers	R/W	R/W	R/W	–	–	–	–	–	–	–	–

Note 1. Access to LRAM and Code Flash
Note 2. Video Input 0 (W), Image Synthesizer 00 (R) via XC2 cross-connect
Note 3. Image Synthesizer 01 (R), Output Image Generator (R) via XC2 cross-connect
Note 4. Image Synthesizer 02 (R), Video Output Warping Engine (R/W) via XC2 cross-connect
Note 5. Image Synthesizer 03 (R) via XC2 cross-connect
Note 6. Write path must not be used.
Note 7. Write access to SFMA registers only.
Note 8. Write access to SDRA registers is prohibited.

14.2.3.6 D1M2(H) bus architecture

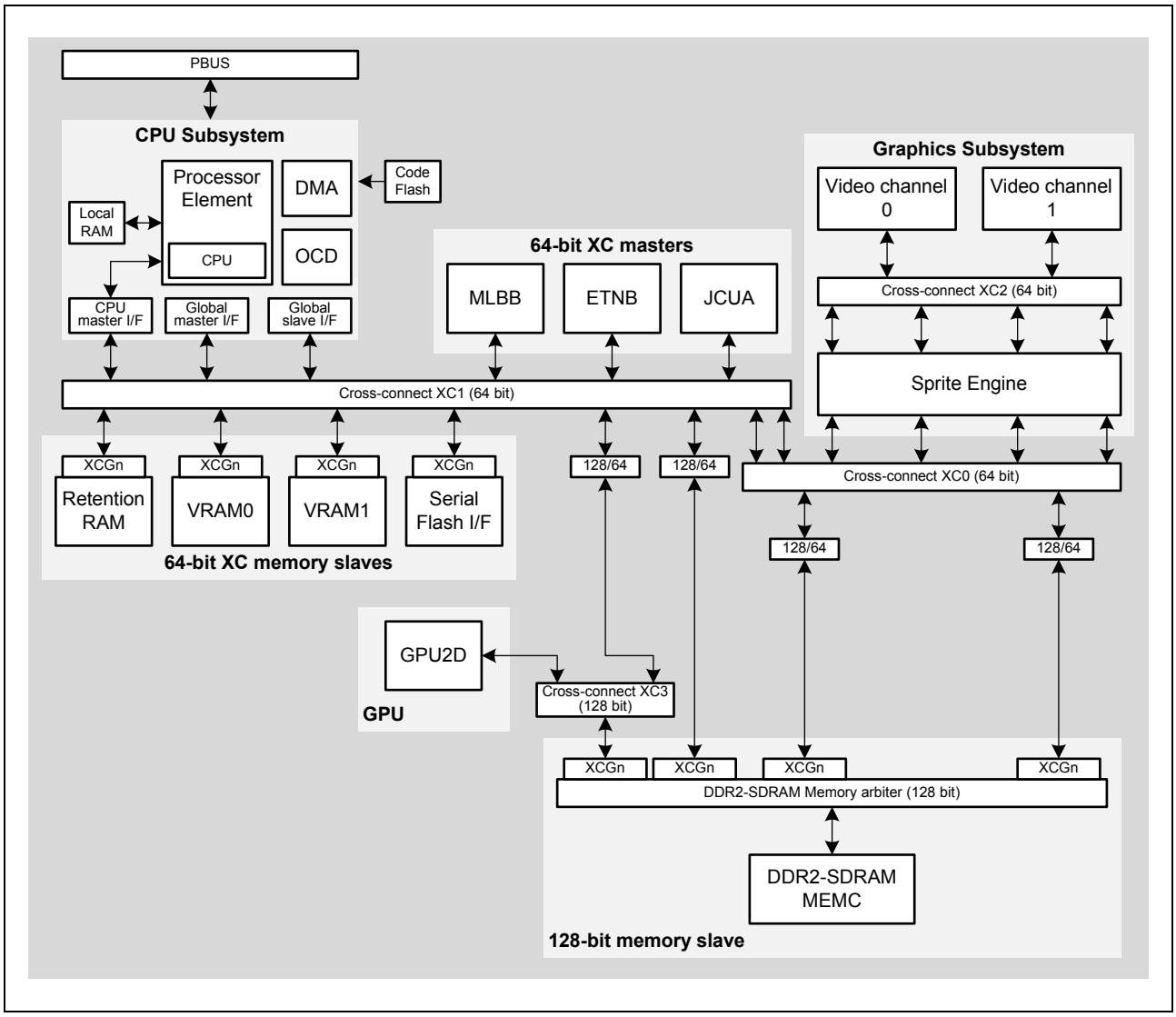


Figure 14.19 D1M2(H) bus architecture overview

DDR2-SDRAM interface

The DDR2-SDRAM Memory Controller (SDRB) provides four 128-bit wide ports to supply high data rates for the video and graphics masters. Refer to Section 14.3.4, D1M2(H) DDR2-SDRAM interface arbitration and bandwidth for further details.

D1M2(H) 2D Graphics Processing Unit GPU2D

The 2D Graphics Processing Unit (GPU2D) of the D1M2(H) devices acts as a master on the XC3 cross-connect towards two slaves:

- 128-bit port to the DDR2-SDRAM port 1, that boosts the drawing performance in conjunction with the external DDR2-SDRAM
- 64-bit port to the XC1 cross-connect for accessing on-chip memories.

(a) D1M2(H) cross-connect XC0

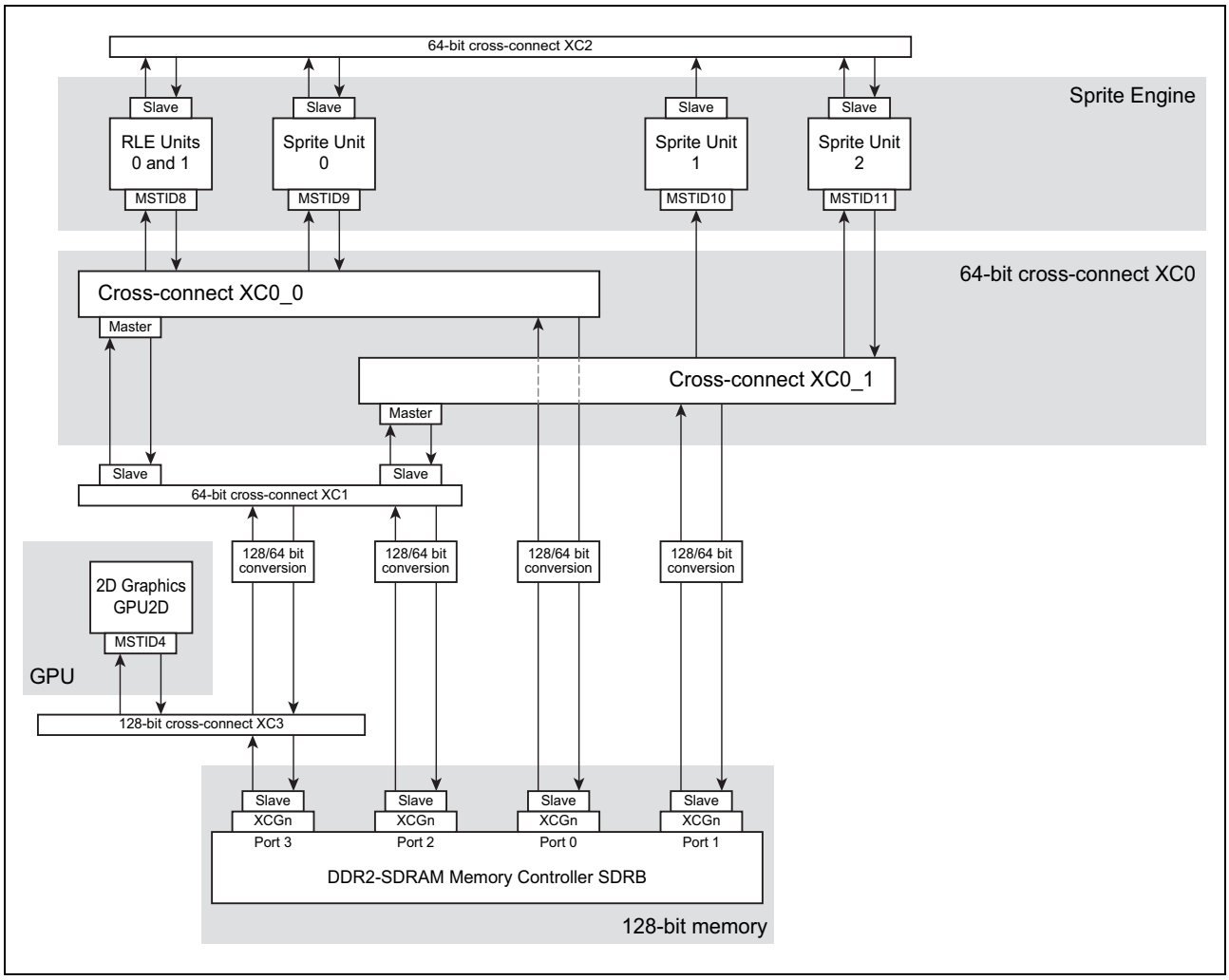
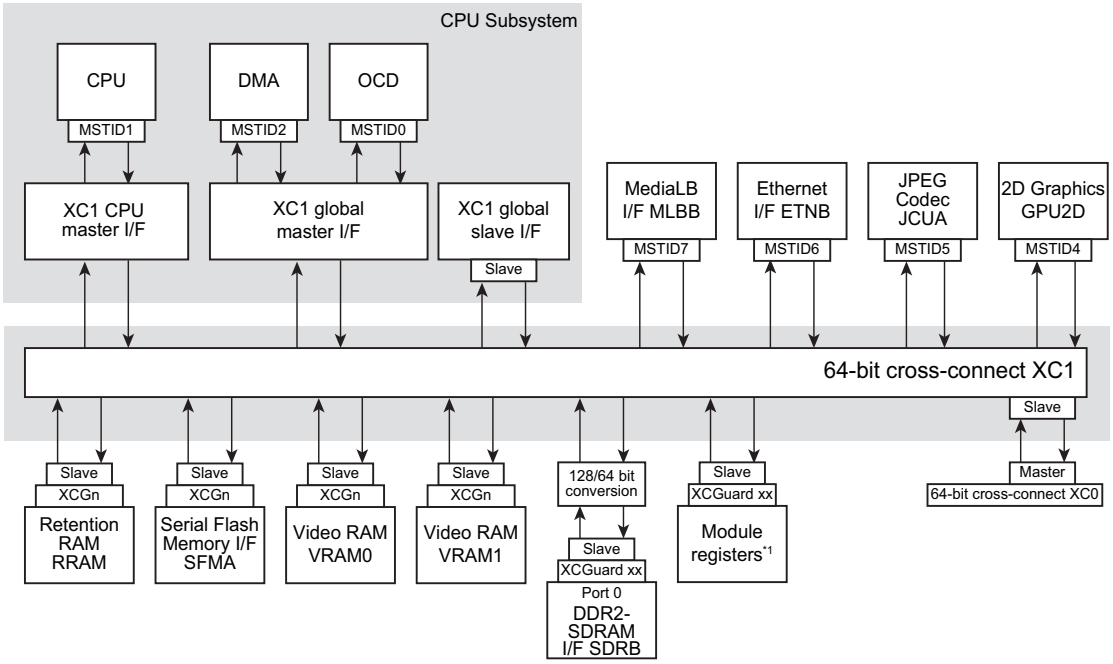


Figure 14.20 D1M2(H) cross-connect XC0

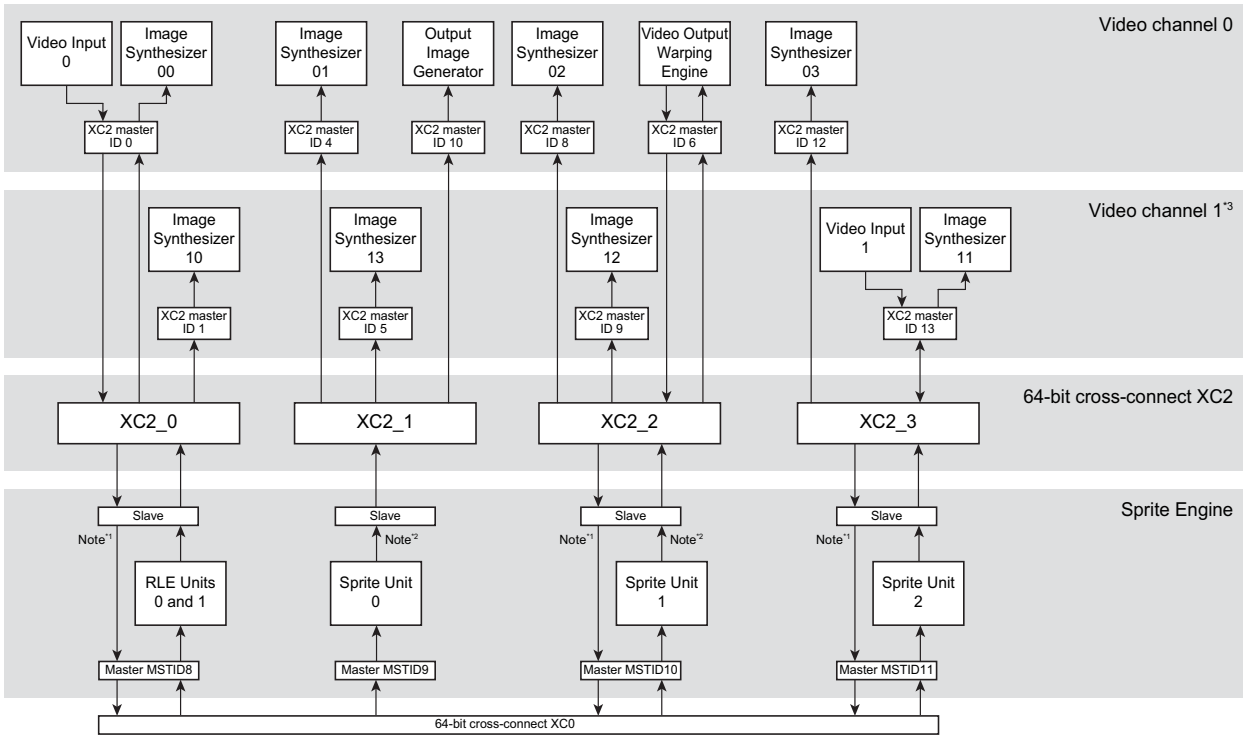
(b) D1M2(H) cross-connect XC1



Note 1. The configuration registers of some modules are accessed via the XC1 cross-connect.

Figure 14.21 D1M2(H) cross-connect XC1

(c) D1M2(H) cross-connect XC2



Note 1. All write accesses of the video channel masters are passed through the Sprite Engine to the XC0 cross-connect.

Note 2. Read accesses of the Warping Engine via Sprite Unit 0 and the Output Image Generator via Sprite Unit 1 must not read from addresses, covered by the Sprite Engine's virtual frame (refer to Section 43, Sprite Engine (SPEA) for details about the virtual frame). This way the Sprite Unit passes the master's read access through to the XC0 cross-connect.

Note 3. The video channel 1 is only available for D1M2(H) devices.

Figure 14.22 D1M2(H) cross-connect XC2

(d) D1M2(H) connection matrix

Table 14.14 D1M2(H) connection matrix

		Masters										
		CPU Subsystem										
		CPU I/F	Global I/F						Sprite Engine			
		CPU	DMA	OCD	GPU2D	MLBB	ETNB	JCUA	RLE Units*2	Sprite Unit 0*3	Sprite Unit 1*4	Sprite Unit 2*5
Slaves		MSTID 1	MSTID 2	MSTID 0	MSTID 4	MSTID 7	MSTID 6	MSTID 5	MSTID 8	MSTID 9	MSTID 10	MSTID 11
CPU Subsystem global I/F*1		–	–	–	R/W*6	R/W	R/W*6	R/W	R/W	R/W	R/W*6	R/W
RRAM		R/W	R/W	R/W	–	–	–	–	–	–	–	–
SFMA		R/W*7	R/W*7	R/W*7	R/W*6	–	–	R/W*6	R/W*6	R/W*6	R/W*6	R/W*6
VRAM0		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
VRAM1		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SDRB	Port 0	–	–	–	–	–	–	–	R/W	R/W	–	–
	Port 1	–	–	–	–	–	–	–	–	–	R/W	R/W
	Port 2	R/W	R/W	R/W	–	R/W	R/W	R/W	–	–	–	–
	Port 3	–	–	–	R/W	–	–	–	–	–		
Module registers		R/W	R/W	R/W	–	–	–	–	–	–	–	–

Note 1. Access to LRAM and Code Flash

Note 2. Video Input 0 (W), Image Synthesizer 00 (R), Image Synthesizer 10 (R) via XC2 cross-connect

Note 3. Image Synthesizer 01 (R), Image Synthesizer 13 (R), Output Image Generator (R) via XC2 cross-connect

Note 4. Image Synthesizer 02 (R), Image Synthesizer 12 (R), Video Output Warping Engine (R/W) via XC2 cross-connect

Note 5. Video Input 1 (W), Image Synthesizer 03 (R), Image Synthesizer 11 (R) via XC2 cross-connect

Note 6. Write path must not be used.

Note 7. Write access to SFMA registers only.

14.3 Arbitration, bandwidth and latencies

This section describes the arbitration policies of the slave interfaces, the data transfer bandwidth, provided to the masters, and defines the latencies needs to be taken into account, when a master initiates an access to a slave.

14.3.1 Transaction based busses

The cross-connect systems and the CPU Subsystem internal busses operate transactions based in order to avoid blocked busses due to not completed data transfers.

The diagram below shows the handling of transfers.
While the “classical” bus is blocked until the slave responses (“T1 response 001”) to a master request (“M1 request 001”), the “transaction bus” handles several master requests (“M1 request 001”, “M2 request 002”, “M1 request 003”) concurrently and routes the slave responses (“T2 response 002”, “T1 response 001”, “T1 response 003”) to the requesting masters.

Since bus blocking times are avoided, the bus effective bandwidth increases.

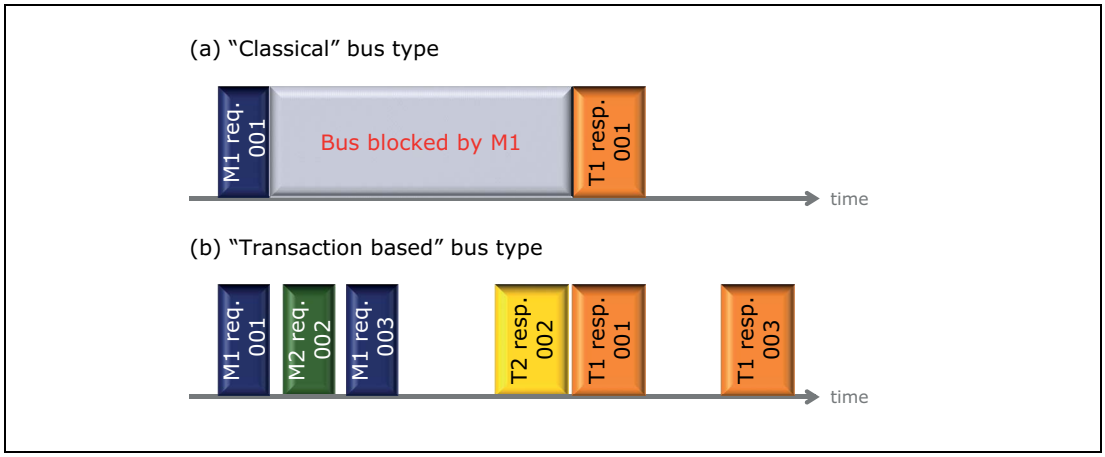


Figure 14.23 “Classical” vs. “transaction based” bus types

14.3.2 Cross-connects arbitration and bandwidth

Cross-connect arbitration

The slave modules of the cross-connect system arbitrate accesses of the masters according to a fair round robin scheme. This ensures an equal minimum bandwidth assigned to each master.

Cross-connect bandwidth

The maximum total bandwidth, a slave can provide, is determined by the cross-connect system’s bus width and clock frequency. All cross-connect systems operate with 64 data bits, the maximum slave bandwidth depends on the device:

- D1L2(H): 64 bit x 60 MHz = 480 MB/sec
- D1M1: 64 bit x 80 MHz = 640 MB/sec
- D1M1H: 64 bit x 100 MHz = 800 MB/sec
- D1M1-V2: 64 bit x 80 MHz = 640 MB/sec
- D1M1A: 64 bit x 120 MHz = 960 MB/sec

- D1M2(H): 64 bit x 120 MHz = 960 MB/sec

14.3.3 Cross-connect access latencies

In the following minimum access latencies are specified, when a cross-connect master writes or reads data from a cross-connect slave.

Explanation to the latency tables

- The numbers given in the table refer to the number clock cycles of the cross-connect operation clock C_ISO_XCCLK.
- The cycle numbers specify the minimum number of clock cycles under the condition that no additional arbitration takes place, because of several master acting on the same slave.
- JCUA: JPEG Codec Unit performs non-blocking data write to memory via a buffer. This means a memory write from the JCUA's perspective is always performed in 1 XCCLK cycle, provided the buffer is not full, while the real transfer from the buffer to the memory needs more cycles.
  The JCUA write cycle indicators in the Table 14.15, Minimum write access latency*1 mean:
  (latency cycle for JCUA) / (transfer cycles from buffer to target)
- VIn: Video Input of video channel n, writing data to memory through the Sprite Engine
- ISn: Image Synthesizers of the video channel n, i.e.
  - video channel 0 Image Synthesizers 00 to 03
  - video channel 1 Image Synthesizers 10 to 13

  reading data from memory through the Sprite Engine
- VOWE: Video Output Warping Engine of video channel 0, reading/writing data from/to memory through the Sprite Engine

14.3.3.1 Minimum write access latency

Table 14.15 Minimum write access latency*1

	Master												
Slave	CPU	JCUA	ETNB	MLB	VI0	IS0	VI1	IS1	VOWE	GPU2D	NFMA	Write target	
CPU Sub-system	8	–	–	–	–	–	–	–	–	–	–	Code Flash	
	3	–	–	16*2	18*2	–	18*2	–	18*2	26*2	16	Local memory	
RRAM	19	–	–	–	–	–	–	–	–	–	–	Retention RAM	
VRAMn	6*3	22	9	6	8	–	8	–	8	16	6	32-bit access	direct via XC1
	8*3	23	10	7	9	–	9	–	9	17	7	32-bit ECC mode	via VRAM Wrapper
	8*3	23	10	7	9	–	9	–	9	17	7	32-bit mirror	
	8*3	23	10	7	9	–	9	–	9	17	7	Wrapping mode 24-bit RGB888	
	8*3	23	10	7	9	–	9	–	9	17	7	Wrapping mode 18-bit RGB666	
	8*3	23	10	7	9	–	9	–	9	17	7	Wrapping mode 24-bit αRGB6666	
SFMA	–	–	–	–	–	–	–	–	–	–	–	External flash pages	
	23*3	–	–	–	–	–	–	–	–	–	–	SFMA registers	
SDRA/ SDRB	19*3	32	21	18	16	–	16	–	16	15	18	Direct access via XC0	
	19*3	32	21	18	16	–	16	–	16	15	18	Access to mirrors via VRAM Wrapper	
HYPB	31	47	34	–	33	–	–	–	33	41	36	External RAM	
OCTA	54	70	57	–	56	–	–	–	56	64	59	External RAM	

Note 1. Numbers refer to C_ISO_XCCLK clock cycles.
Note 2. Access via CPU Subsystem global slave I/F.
Note 3. Access via CPU master I/F.

14.3.3.2 Minimum read access latency

Table 14.16 Minimum read access latency*1 (1/2)

	Master												
Slave	CPU	JCUA	ETNB	MLB	VI0	IS0	VI1	IS1	VOWE	GPU2D	NFMA	Read target	
CPU Sub-system	5	14*2	19*2	18*2	–	22*2	–	22*2	22*2	25*2	–	Code Flash	
	1	13*2	18*2	17*2	–	23*2	–	23*2	23*2	24*2	18	Local RAM	
RRAM	21*3	–	–	–	–	–	–	–	–	–	–	Retention RAM	
VRAMn	7*3	5	9	8	–	14	–	14	14	15	8	32-bit access	direct via XC1
	9*3	7	11	10	–	15	–	16	16	17	9	32-bit ECC mode	via VRAM Wrapper
	8*3	6	10	9	–	15	–	15	15	16	9	32-bit mirror	
	8*3	6	10	9	–	15	–	15	15	16	9	Wrapping mode 24-bit RGB888	
	8*3	6	10	9	–	15	–	15	15	16	9	Wrapping mode 18-bit RGB666	
	8*3	6	10	9	–	15	–	15	15	16	9	Wrapping mode 24-bit αRGB6666	
SFMA	31*3	29	–	–	–	38	–	38	38	39	–	External flash pages	
	20*3	–	–	–	–	–	–	–	–	–	–	SFMA registers	

Table 14.16 Minimum read access latency*1 (2/2)

Slave	Master											Read target
	CPU	JCUA	ETNB	MLB	VI0	IS0	VI1	IS1	VOWE	GPU2D	NFMA	
SDRA/ SDRB	20*3	19	24	23	–	25	–	25	25	22	20	Direct access via XC0
	22*3	19	24	23	–	25	–	25	25	22	20	Access to mirrors via VRAM Wrapper
HYPB	79	77	81	–	–	86	–	86	86	87	82	External RAM or Flash
OCTA	86	84	88	–	–	93	–	93	93	94	89	External RAM or Flash

Note 1. Numbers refer to C_ISO_XCCLK clock cycles.
Note 2. Access via CPU Subsystem global slave I/F.
Note 3. Access via CPU master I/F.

14.3.4 D1M2(H) DDR2-SDRAM interface arbitration and bandwidth

The DDR2-SDRAM internal bus interface is 128-bit wide and operates with 120 MHz, thus provides a total bandwidth of 1.92 GB/s.
The external DDR2-SDRAM interface is 32-bit wide and can operate with 480 MHz.

It features four read/write ports. Arbitration is ruled by a programmable priority generator, that allows to dynamically adjust priorities of the ports.

Refer to Section 16, DDR2-SDRAM Memory Controller (SDRB) for details.

14.4 Bus Switch for external memory interfaces (D1M1-V2, D1M1A only)

The HYPBSEL selects HyperBus or OctaBus usage and the SFMASEL selects SFMA1 or SFMA2.

These interfaces are configurable via the bus switch control register XCRAMCFG0.

The HyperBus and OctaBus interface share also the ports P21_[9:0] and P22_10.

The SFMA1 and SFMA2 interface share also the ports P21_[9:6].

The related ports are also selected by the bus switch control register XCRAMCFG0.

(1) XCRAMCFG0 – Bus switch control register

Access: This register can be read/written in 32-bit units.

Address: FFC0 6050_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	CSSEL2	CSSEL	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	SFMASEL	HYPBSEL	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 14.17 XCRAMCFG0 register contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, these bits are always read as 0. When writing, always write 0.
21	CSSEL2	Select P45_1 4th alternative function (D1M1A only) 0: VDCE0_VO_TCON3 1: SFMA2SSL, which is selected by XCRAMCFG0.CSSEL
20	CSSEL	Select P45_13 5th alternative function (D1M1A only) 0: SFMA2SSL 1: OCTA0 MCS1
19 to 10	Reserved	When read, these bits are always read as 0. When writing, always write 0.
9	SFMASEL	Bus type selection at cross-connect XC1 (D1M1A only) 0: SFMA1 1: SFMA2
8	HYPBSEL	Bus type selection at cross-connect XC1 0: HyperBus 1: OctaBus
7 to 0	Reserved	When read, these bits are always read as 0. When writing, always write 0.

CAUTION

XCRAMCFG0 must be set according to the HyperBus or OctaBus and SFMA1 or SFMA2 selection before the XC0 cross-connect is released from reset, i.e. before MRSTC.XC0RES = 1 is set.

14.5 PBUS structure

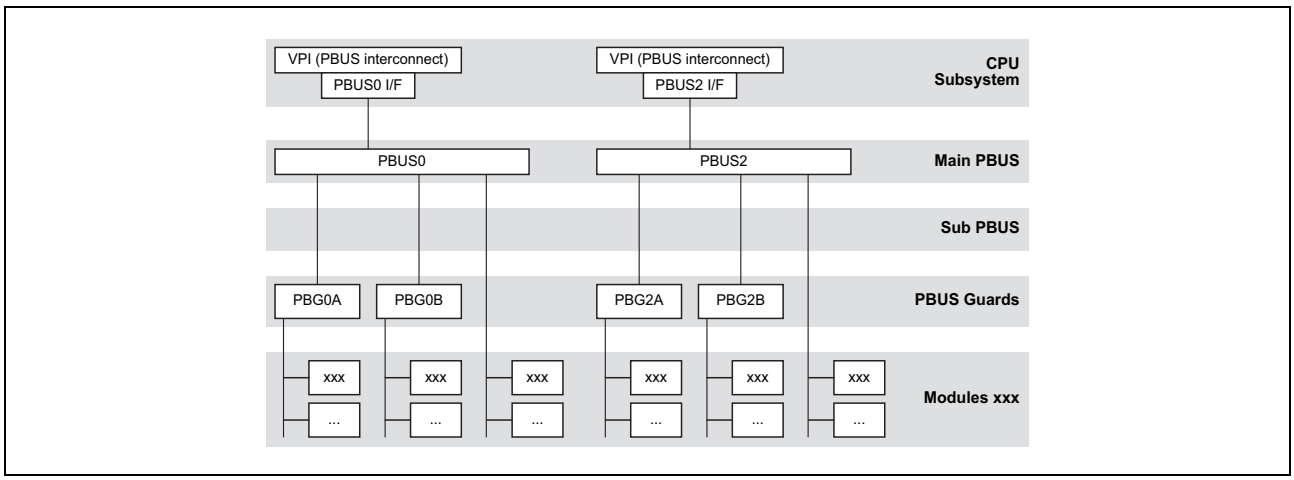


Figure 14.24 PBUS0 and PBUS2 structure

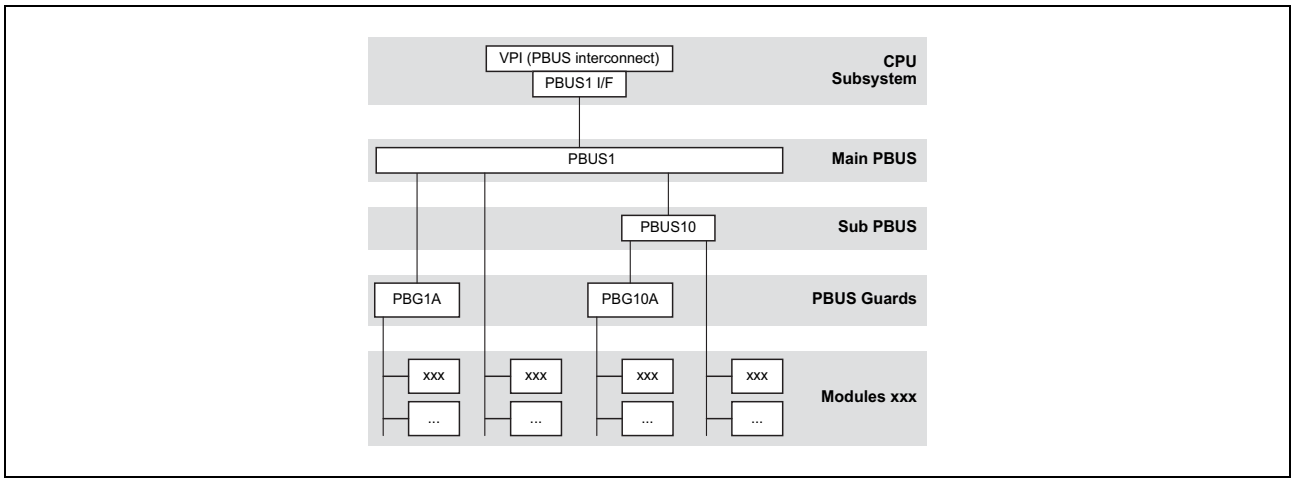
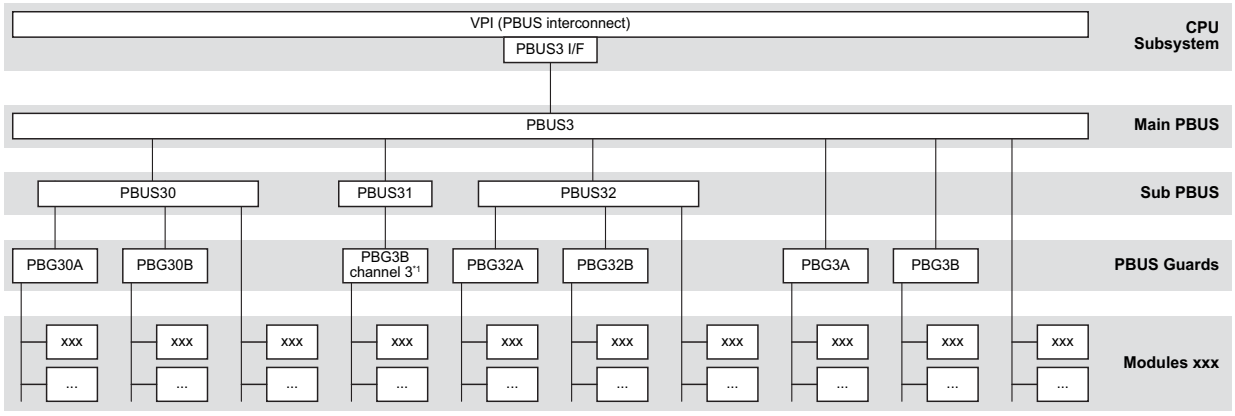


Figure 14.25 PBUS1 structure



Note 1. All PBUS31 modules are guarded by PBG3B guard channel 3.

Figure 14.26 PBUS3 structure

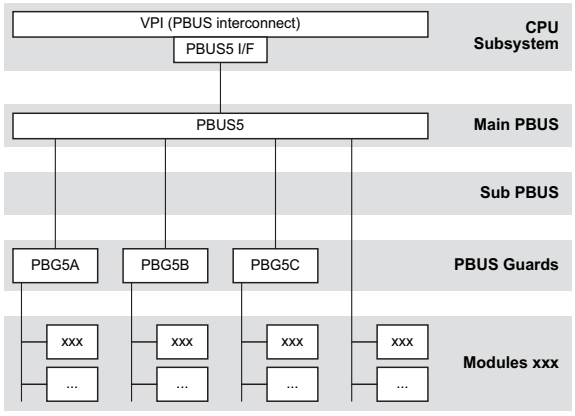


Figure 14.27 PBUS5 structure

Table 14.18 PBUS structure (1/8)

PBUS		PBUS Guard		
Main	Sub	Guard	Channel	Module
PBUS0	—	—		PBG0A
	—	—		PBG0B
	—	PBG0A	0	Data Flash
			1	FACI
			2	FCURAM area
			3	FACI command-issuing area
			4	ICU-S
			5	Reserved
			6	Code Flash ECC / Data Flash ECC
			7	P0A_ERRSLV
			8 to 15	Reserved
	—	PBG0B	0	Interrupt Controller INTC2
			1	DMA Controller
			2 to 15	Reserved
		—		SYSCTRL
		—		ECCIC1
		—		ECCCPU1
		—		ECCAXC
		—		Data Flash Memory Control (DCIB)
PBUS1	—	—		PBG1A
	—	PBG1A	0	SSIF0
			1	SSIF1
			2	LCBI0
			3	Reserved
			4	P1A_ERRSLV
			5	SDRB0
			6	DCRA0
			7 to 15	Reserved
	PBUS10	—		PBG10A
		PBG10A	0	PCMP0
			1	SG0
			2	SG1
			3	SG2
			4	SG3
			5	SG4
			6	P10A_ERRSLV
			7 to 15	Reserved

Table 14.18 PBUS structure (2/8)

PBUS		PBUS Guard		
Main	**Sub**	**Guard**	**Channel**	**Module**
PBUS2	—	—		PBG2A
	—	—		PBG2B
	—	PBG2A	0	ETNB0
			1	MLBB0
			2	GPU2D0
			3	VRAM0
			4	VRAM1
			5	SPEA0
			6	MIPI0
			7	VOCA0
			8	SDRA_TREST*1
			9	P2A_ERRSLV
			10 to 15	Reserved
	—	PBG2B	0	Reserved
			1	PM_R*2
			2	PM_SF0*2
			3	PM_SF1*2
			4	PM_V0*2
			5	PM_V1*2
			6	Reserved
			7	PM_SD*2
			8	Reserved
			9	Reserved
			10	Reserved
			11	PM_HB*2
			12	P2B_ERRSLV
			13	NFMA
			14 to 15	Reserved

Note 1. SDRA_TREST: SDR-SDRAM transaction restrictor
Note 2. PM_xx: Performance monitor

Table 14.18 PBUS structure (3/8)

PBUS		PBUS Guard		
Main	**Sub**	**Guard**	**Channel**	**Module**
PBUS3	—	—		PBG3A
	—	—		PBG3B
	—	PBG3A	0	Reserved
			1	ECCRCAN0 / ECCRCFD0,1*1
			2	Reserved
			3	RRAMECC0
			4	RLN30
			5	RLN31
			6	RLN32
			7	RLN33
			8	RSCAN0 / RSCFD0*1
			9 to 15	Reserved
		PBG3B	0	TAUB0
			1	TAUB1
			2	TAUB2
			3	All modules of PBUS31
			4	TAUJ0
			5	OSTM0
			6	OSTM1
			7	WDTA1
			8	ADCE0
			9	Reserved
			10	P3A_ERRSLV
			11	P3B_ERRSLV
			12	Reserved
			13 to 15	Reserved

Note 1. Devices with RS-CAN or RS-CANFD interfaces have different product names, refer to **Section 1.5, Ordering Information**.

Table 14.18 PBUS structure (4/8)

PBUS		PBUS Guard		
Main	**Sub**	**Guard**	**Channel**	**Module**
PBUS3	PBUS30	—		PBG30A
		—		PBG30B
		PBG30A	0	CSIG0*1
			1	CSIG1*1
			2	CSIG2*1
			3	CSIG3*1
			4	CSIH0*1
			5	CSIH1*1
			6	CSIG0*2
			7	CSIG1*2
			8	CSIG2*2
			9	CSIG3*2
			10	CSIH0*2
			11	CSIH1*2
			12	RIIC0
			13	RIIC1
			14	ISM0
			15	P30A_ERRSLV
		PBG30B	0	ECCISM0
			1	P30B_ERRSLV
			2 to 15	Reserved

Note 1. CSIGn/CSIHn registers whose offset address are within 0000 0000_H to 0000 $0FFF_H$.
Note 2. CSIGn/CSIHn registers whose offset address are within 0000 1000_H to 0000 $1FFF_H$.

Table 14.18 PBUS structure (5/8)

PBUS		PBUS Guard		
Main	**Sub**	**Guard**	**Channel**	**Module**
PBUS3	PBUS31	PBG3B (shared with modules connected to PBUS3)	3	PWM Generator PWGA0
				PWM Generator PWGA1
				PWM Generator PWGA2
				PWM Generator PWGA3
				PWM Generator PWGA4
				PWM Generator PWGA5
				PWM Generator PWGA6
				PWM Generator PWGA7
				PWM Generator PWGA8
				PWM Generator PWGA9
				PWM Generator PWGA10
				PWM Generator PWGA11
				PWM Generator PWGA12
				PWM Generator PWGA13
				PWM Generator PWGA14
				PWM Generator PWGA15
				PWM Generator PWGA16
				PWM Generator PWGA17
				PWM Generator PWGA18
				PWM Generator PWGA19
				PWM Generator PWGA20
				PWM Generator PWGA21
				PWM Generator PWGA22
				PWM Generator PWGA23
				PWM Generator clock divider PWBA0
				SLPWGA0
				PWM Generator diagnostic control PWSA0

Table 14.18 PBUS structure (6/8)

PBUS		PBUS Guard		
Main	**Sub**	**Guard**	**Channel**	**Module**
PBUS3	PBUS32	—		PBG32A
		—		PBG32B
		PBG32A	0	Port JP0 control*1
			1	Port P0 control*2
			2	ECM (Master)
			3	AWOT0
			4	RTCA0
			5	WDTA0
			6	Port JP0 control*1
			7	Port P0 control*2
			8	Analog port filter control FCLA0
			9	Port JP0 control*1
			10	Port JP0 control*1
			11	Port JP0 control*1
			12	Port P0 control*2
			13	Port P0 control*2
			14	Port P0 control*2
			15	Analog port filter control FCLA1
		PBG32B	0	P32A_ERRSLV
			1	P32B_ERRSLV
			2	ECM (Checker)
			3	ECM (Common)
			4 to 15	Reserved

Note 1. Guard settings for PBG32A channel 0, 6, 9, 10, 11 (FSGD32APROTn register) must be set to same values. Otherwise correct function is not guaranteed.

Note 2. Guard settings for PBG32A channel 1, 7, 12, 13, 14 (FSGD32APROTn register) must be set to same values. Otherwise correct function is not guaranteed.

Table 14.18 PBUS structure (7/8)

PBUS		PBUS Guard		
Main	**Sub**	**Guard**	**Channel**	**Module**
PBUS5	—	—		PBG5A
				PBG5B
				PBG5C
		PBG5A	0 to 1	Reserved
			2	Port P1 to P47 control*1
			3	FLMD
			4 to 5	Reserved
			6	SELF
			7	PRMR0 (access to system control registers PRMR0CFGx), SELB (except SLPWGA0)
			8	BERR0
			9	Port P1 to P47 control*1
			10	PRDNAME and option byte registers
			11	SD_ISO_VDD (QOS registers)
			12	SYS (Stand-by Controller, RESF,RESFR, RESFC, RESFCR, SWRESA, PROTCMD0, PROTS0)
			13	Port P1 to P47 control*1
			14	Port P1 to P47 control*1
			15	Port P1 to P47 control*1
		PBG5B	0	Digital port filter control DNFA0
			1	Digital port filter control DNFA1
			2	Digital port filter control DNFA2
			3	Digital port filter control DNFA3
			4	Reserved
			5	Digital port filter control DNFA5
			6	Digital port filter control DNFA6
			7	Digital port filter control DNFA7
			8	Digital port filter control DNFA8
			9	Digital port filter control DNFA9
			10	Digital port filter control DNFA10
			11	Reserved
			12	XC Guard XCG15 (OctaBus)
			13 to 15	Reserved

Note 1. Guard settings for PBG5A channel 2, 9, 13, 14, 15 (FSGD5APROTn register) must be set to same values. Otherwise correct function is not guaranteed.

Table 14.18 PBUS structure (8/8)

PBUS		PBUS Guard		
Main	**Sub**	**Guard**	**Channel**	**Module**
PBUS5	—	PBG5C	0	P5A_ERRSLV
			1	P5B_ERRSLV
			2	P5C_ERRSLV
			3	XC Guard XCG3 (RRAM)
			4	XC Guard XCG4 (VRAM0)
			5	XC Guard XCG5 (VRAM1)
			6	XC Guard XCG6 (SFMA)
			7	XC Guard XCG7 (SDRB port 3)
			8	XC Guard XCG8 (XC1 modules registers)
			9	XC Guard XCG9 (SDRB port 2)
			10	XC Guard XCG10 (SDRB port 0)
			11	XC Guard XCG11 (SDRB port 1)
			12	IDMODI
			13	Reserved
			14	XC Guard XCG13 (SFMA1)
			15	XC Guard XCG14 (Hyper BUS)
		—		CLMA0
				CLMA1
				CLMA2
				CLMA3
				CLMA4
				CLMA5
				CLMA6
				CLMAC
				SYS (others)
				PWRG

Note that some of the modules listed above are described in RH850/D1x Flash Memory User's Manual: Hardware Interface.

14.6 Bus Guards

Several bus guards allow to grant or prohibit certain access rights to certain bus masters for accessing particular busses and their attached modules.

- Processor Element Guard (PEG)
  regulates access of all master, except the CPU, to CPU's local RAM
- PE's Internal Peripheral Guard (IPG)
  protects the Processor Element's control and configuration registers against illegal access
- PBUS guards (PBG)
  regulates access to all PBUS modules
- Cross-connect guards (XCG)
  regulates access to all cross-connect slave modules

In case a bus guard detects an access violation, error notifications are forwarded to the Error Control Module, and thus can generate interrupts or a reset.

14.6.1 SPID and PEID assignment

Table 14.19 SPID and PEID assignment

		Device
SPID[1:0] SPID is set by the CPU system.	0 to 3*1	CPU
	0 to 3*2	DMA
	0	Cross-connect XC1 master (via XC1 global I/F)
PEID[2:0]	0	Debug master*3
	1	CPU
	1 to 7*2	DMA
	4	Cross-connect XC1 master (via XC1 global I/F)

Note 1. SPID from CPU is set in the MCFG0 register of the CPU Subsystem. Default SPID is 01_B.
Refer to the RH850 Family User's Manual: Software

Note 2. In the case of DMA access, PEID/SPID is the value in the DMnnCM register of the DMA Controller.

Note 3. PEG does not protect access from debug master that PEID is set to 0.

14.6.2 PE Guard (PEG)

PE Guard error signals

Table 14.20 PE Guard error signals

CFG signal	Function	Connected to
PE1_PEG_RDERR	Guard error during Processor Element read access from Global Interconnect (GVCI)	Error Control Module INTPEGRD*1
PE1_PEG_WRERR	Guard error during Processor Element write access from Global Interconnect (GVCI)	

Note 1. This Error Control Module input signal is a logical OR of both error signals.
Refer to Section 48, Error Control Module (ECM) for details.

14.6.2.1 Overview of the PEG Function

The PEG is a constituent of the slave guard system to prevent unauthorized access to the resources in the PE from the external master. This function protects access to the local RAM in the PE. In the initial state after a reset, all access by masters other than the PE itself is disabled. Setting the registers listed in Section 14.6.2.3, List of PEG Protection Setting Registers, enables access by masters other than the PE itself.

(1) Detecting PE guard violation
If the external master outside the PE makes an unauthorized access to the resource area in the PE for which PE guard is set, the access is detected as a PE guard violation.

(2) Blocking unauthorized accesses
When a PE guard violation is detected, unauthorized accesses to the internal resources of the PE are blocked to prevent the contents of PE resources from being modified illegally.

(3) Notifying occurrence of violation
An error response to an unauthorized access is sent to the request source of external master.

When the DMAC makes an unauthorized access, a DMA transfer error occurs due to an error response.

NOTE

PEG does not protect access from debug master that PEID is set to 0.

14.6.2.2 Protection Made by SPID

NOTE

For a list of SPID numbers refer to Section 14.6.1, SPID and PEID assignment.

- Setting PEG Protection
  - Up to four areas can be set depending on the Local RAM address of the own PE.
  - The area range is specified by the base address and the mask bit (4 Kbytes to 4 Gbytes).
  - “Read enable” and “write enable” can be set for each area.
  - “Enable” or “disable” can be selected on each system protection identifier (SPID) basis for each area.
- Access permission by the system protection identifier (SPID) (see Figure 14.28)

1. When the Local RAM area is to be accessed, go to step 2.
   Otherwise, return an error response.
2. When any of enabled area 0 to area 3 is to be accessed, go to step 3.
   Otherwise, return an error response.
3. Are all the conditions below for the relevant area met?
   - The system protection identifier (SPID) is enabled.
   - Required operations (read/write) are enabled.
     Otherwise, return an error response.

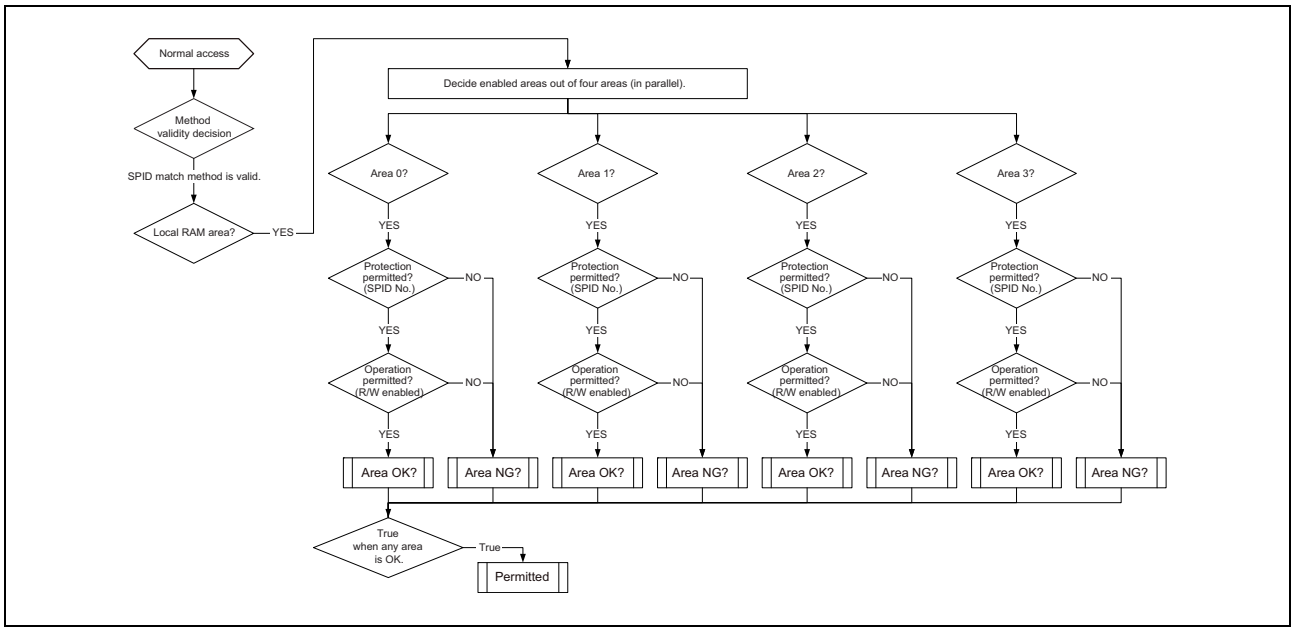


Figure 14.28 Access Permission by the System Protection Identifier (SPID)

14.6.2.3 List of PEG Protection Setting Registers

Make necessary settings for the following registers to protect PE resources from unauthorized accesses by the external master.

Accesses to registers listed in Table 14.21 are not protected by the PEG. Make access protection settings by the MPU or IPG as needed.

Table 14.21 Base Address of PEG Register: FFFE E600$_H$

Address Offset	Size (Byte)	Register Name	Symbol	Right	R/W	Operable Bit				Initial Value
						1	8	16	32	
+00C$_H$	2	PE guard SPID master decision control register	PEGSP	—	R/W	—	√	√	—	0000$_H$
+080$_H$	4	PE guard area 0 mask setting register	PEGG0MK	—	R/W	—	√	√	√	0000 0000$_H$
+084$_H$	4	PE guard area 0 base setting register	PEGG0BA	—	R/W	—	√	√	√	0000 0000$_H$
+090$_H$	4	PE guard area 1 mask setting register	PEGG1MK	—	R/W	—	√	√	√	0000 0000$_H$
+094$_H$	4	PE guard area 1 base setting register	PEGG1BA	—	R/W	—	√	√	√	0000 0000$_H$
+0A0$_H$	4	PE guard area 2 mask setting register	PEGG2MK	—	R/W	—	√	√	√	0000 0000$_H$
+0A4$_H$	4	PE guard area 2 base setting register	PEGG2BA	—	R/W	—	√	√	√	0000 0000$_H$
+0B0$_H$	4	PE guard area 3 mask setting register	PEGG3MK	—	R/W	—	√	√	√	0000 0000$_H$
+0B4$_H$	4	PE guard area 3 base setting register	PEGG3BA	—	R/W	—	√	√	√	0000 0000$_H$

To set PEG Protection, register setting order as described below must be followed.

1. Set PEGGnMK(mask setting register)
2. Set PEGGnBA(base setting register)

14.6.2.4 Register Set

(1) PE Guard SPID Master Decision Control Register (PEGSP)

This register is used to enable or disable access by an external master to the resources in the PE. The initial value of the SPEN bit is 0, which disables access to PE resources by an external master. Setting the SPEN bit to 1 enables access by an external master under the conditions set by PEGGnMK and PEGGnBA.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 14.22 PEGSP register contents

Bit Position	Bit Name	Function
15 to 1	—	Reserved. These bits are always read as 0. The write value should always be 0.
0	SPEN	This bit enables or disables detection of accesses by the external master having SPID. 0: Detection of accesses by the external master having SPID is disabled. 1: Detection of accesses by the external master having SPID is enabled.

(2) PE Guard Area n Mask Setting Register (PEGGnMK)

In combination with the PEGGnBA register, this register specifies a range or ranges within PE guard protection area n. Setting a GnMASK bit to 1 masks the corresponding address bit and places the address bit outside the range of address comparison target in the PE guard protection area n. The minimum unit of setting the PE guard protection area n is 4 Kbytes.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GnMASK															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GnMASK				—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.23 PEGGnMK register contents

Bit Position	Bit Name	Function
31 to 12	GnMASK	These bits determine whether to mask upper bits of the address that specifies the range of PE guard protection area n. 0: Target address bits are compared when determining the PE guard area. 1: Target address bits are not compared when determining the PE guard area.
11 to 0	—	Reserved. These bits are always read as 0. The write value should always be 0.

(3) PE Guard Area n Base Setting Register (PEGGnBA)

In combination with the PEGGnMK register, this register specifies a range or ranges within PE guard protection area n. Setting the GnEN bit to 1 brings the address enable conditions specified by this register and the PEGGnMK register into effect.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GnBASE															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GnBASE				—	—	—	—	GnSP3	GnSP2	GnSP1	GnSP0	—	GnWR	GnRD	GnEN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.24 PEGGnBA register contents

Bit Position	Bit Name	Function
31 to 12	GnBASE	Base address that specifies PE guard protection area n
11 to 8	—	Reserved. These bits are always read as 0. The write value should always be 0.
7	GnSP3	Enables accesses to PE guard protection area n by the external master. 0: Access by an external master having SPID = 3 is disabled. 1: Access by an external master having SPID = 3 is enabled.
6	GnSP2	Enables accesses to PE guard protection area n by the external master. 0: Access by an external master having SPID = 2 is disabled. 1: Access by an external master having SPID = 2 is enabled.
5	GnSP1	Enables accesses to PE guard protection area n by the external master. 0: Access by an external master having SPID = 1 is disabled. 1: Access by an external master having SPID = 1 is enabled.
4	GnSP0	Enables accesses to PE guard protection area n by the external master. 0: Access by an external master having SPID = 0 is disabled. 1: Access by an external master having SPID = 0 is enabled.
3	—	Reserved. This bit is always read as 0. The write value should always be 0.
2	GnWR	Enables write access to PE guard protection area n. 0: Write access is disabled. 1: Write access is enabled.
1	GnRD	Enables read access to PE guard protection area n. 0: Read access is disabled. 1: Read access is enabled.
0	GnEN	PE guard protection area n enable 0: Settings for access enable conditions are disabled 1: Settings for access enable conditions are enabled

NOTE

PEGGnBA is cleared when PEGGnMK is written.

14.6.3 PE's Internal Peripheral Guard (IPG)

14.6.3.1 Overview of the IPG Function

The IPG is a system to prevent unauthorized accesses to peripheral devices from the CPU core equipped with the IPG. The IPG achieves the following functions. The IPG covers accesses to resources except the ROM and the local RAM.

(1) Detecting violation of peripheral device protection
If the CPU makes an unauthorized access to an area (peripheral device) for which peripheral device protection is set, the access is detected as "violation of peripheral device protection."

(2) Storing unauthorized-access information
When a violation of peripheral device protection is detected, the unauthorized-access information is stored in the IPG's internal register.

(3) Blocking unauthorized accesses
When a violation of peripheral device protection is detected, unauthorized accesses to peripheral devices are blocked to prevent contents of peripheral devices from being modified illegally.

(4) Notifying violation
When a violation of peripheral device protection is detected, an exception is generated.

NOTE

The RH850G3M makes a request for generating a SYSERR (asynchronous) exception via SysErrGen.

(5) Invalidating subsequent accesses
When a violation of peripheral device protection is detected, subsequent accesses (regardless of authorized or unauthorized accesses) are blocked until the error flag is cleared.

NOTE

Even if a request for generating an exception is immediately sent to the CPU in step (4) above, a subsequent access issued by the CPU before receiving a request from the IPG may illegally modify contents of peripheral devices. (If an access after a violation has occurred is also a violation, access protection is made.)

14.6.3.2 IPG Function

(1) This function invalidates accesses according to their attributes (including address, transfer type, and access right).

(2) After an access right violation is detected until the error flag (described later) is cleared by writing by the software, subsequent accesses are invalidated. However, invalidation is applied only to accesses from the CPU and is not applied to accesses from outside the CPU. Invalidation is performed for accesses within the IPG protection range regardless of the IPG protection setting.

(3) When a request for accessing different peripheral devices simultaneously is made due to misalignment or double-word access, the access is executed when all such accesses are enabled.

14.6.3.3 IPG Protection Setting Registers for Illegal Users

To protect peripheral devices from unauthorized accesses by programs in user mode, necessary settings are required for the registers listed below.

- Accesses in user mode are to be detected.
- This register set is intended for IPG settings related to user mode and reading the IPG settings.

Table 14.25 Base Address of IPG Register: FFFE E000$_H$

Address Offset	Size (Byte)	Register Name	Symbol	Right *1	R/W	Operable Bit 1	8	16	32	Initial Value
+002$_H$	2	Peripheral device protection violation access information register	IPGECRUM	SV	R/W	–	–	√	–	Undefined (retained)
+008$_H$	4	Peripheral device protection violation access address register	IPGADRUM	SV	R/W	–	–	–	√	Undefined (retained)
+00D$_H$	1	Peripheral device protection enable register	IPGENUM	SV	R/W	√	√	–	–	00$_H$
+020$_H$	1	Peripheral device protection setting register 0	IPGPMTUM0	SV	R/W	√	√	–	–	00$_H$
+022$_H$	1	Peripheral device protection setting register 2	IPGPMTUM2	SV	R/W	√	√	–	–	00$_H$
+023$_H$	1	Peripheral device protection setting register 3	IPGPMTUM3	SV	R/W	√	√	–	–	00$_H$
+024$_H$	1	Peripheral device protection setting register 4	IPGPMTUM4	SV	R/W	√	√	–	–	00$_H$

Note 1. Registers for which “SV” is described are accessible by accesses with SV right (PSW.UM = 0).

14.6.3.4 Register Set

(1) Peripheral Device Protection Violation Access Information Register (IPGECRUM)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DS[3:0]				EX	WR	RD	VD
Initial value	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

×: Undefined (retained)

Table 14.26 IPGECRUM register contents

Bit Position	Bit Name	Function
15 to 8	—	Reserved. These bits are always read as 0. The write value should always be 0.
7 to 4	DS[3:0]	These bits store the data size of access that made a violation. 1000: Double-word (8 bytes) 0100: Word (4 bytes) 0010: Half-word (2 bytes) 0001: Byte Other than above: RFU
3	EX	This bit is set to 1 when a violation occurred in an instruction fetch read access. In other cases, this bit is cleared to 0.
2	WR	This bit is set to 1 when a violation occurred in a write access or bit operation or CAXI. In other cases, this bit is cleared to 0.
1	RD	This bit is set to 1 when a violation occurred in a read access or bit operation or CAXI. In other cases, this bit is cleared to 0.
0	VD	This bit is set to 1 when a violation of peripheral device protection is detected by a program with the relevant right. If another violation of peripheral device protection is detected, data of this IPGECRUM register and the IPGADRUM register is updated.

NOTE

When the IRE bit value of the IPGENUM register (described later) is 0 and violation of peripheral device protection by a program operating in user mode is an instruction fetch read access, no bit of this register is updated.

(2) Peripheral Device Protection Violation Access Address Register (IPGADRUM)

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	EADR[31:16]															
Initial value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:0]															
Initial value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

×: Undefined (retained)

Table 14.27 IPGADRUM register contents

Bit Position	Bit Name	Function
31 to 0	EADR[31:0]	These bits store the address of the access in which a violation occurred.

NOTE

When the IRE bit value of the IPGENUM register (described later) is 0 and violation of peripheral device protection by a program operating in user mode is an instruction fetch read access, no bit of this register is updated.

(3) Peripheral Device Protection Enable Register (IPGENUM)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	IRE	E
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 14.28 IPGENUM register contents

Bit Position	Bit Name	Function
7 to 2	—	Reserved. These bits are always read as 0. The write value should always be 0.
1	IRE	This bit sets whether to store the access information in the peripheral device protection violation access address register and the peripheral device protection violation access information register when a violation of peripheral device protection occurred in an instruction fetch access. 0: Instruction fetch access information is not stored. (Initial value) 1: Instruction fetch access information is stored. **CAUTION** If you do not want to detect speculative instruction fetches (no instruction is executed in some cases), clear this bit to 0.
0	E	This bit enables or disables the peripheral devices protection function against accesses by the relevant access right. 0: The peripheral device protection function is disabled. (Initial value) 1: The peripheral device protection function is enabled.

(4) Peripheral Device Protection Setting Register 0 (IPGPMTUM0)

Bit	7	6	5	4	3	2	1	0
	—	X1	W1	R1	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R	R

Table 14.29 IPGPMTUM0 register contents

Bit Position	Bit Name	Function
7	—	Reserved. This bit is always read as 0. The write value should always be 0.
6	X1	This bit sets whether to enable instruction fetch read access to peripheral devices to be connected to the PBUS and Data Flash. 0: Instruction fetch read access to peripheral devices to be connected to the PBUS and Data Flash is treated as violation. (Initial value) 1: Instruction fetch read access to peripheral devices to be connected to the PBUS and Data Flash is not restricted.
5	W1	This bit sets whether to enable write access to peripheral devices to be connected to the PBUS and Data Flash. 0: Write access to peripheral devices to be connected to the PBUS and Data Flash is treated as violation. (Initial value) 1: Write access to peripheral devices to be connected to the PBUS and Data Flash is not restricted.
4	R1	This bit sets whether to enable read access to peripheral devices to be connected to the PBUS and Data Flash. 0: Read access to peripheral devices to be connected to the PBUS and Data Flash is treated as violation. (Initial value) 1: Read access to peripheral devices to be connected to the PBUS and Data Flash is not restricted.
3 to 0	—	Reserved. This bit is always read as 0. The write value should always be 0.

(5) Peripheral Device Protection Setting Register 2 (IPGPMTUM2)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	X0	W0	R0
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 14.30 IPGPMTUM2 register contents

Bit Position	Bit Name	Function
7 to 3	—	Reserved. This bit is always read as 0. The write value should always be 0.
2	X0	This bit sets whether to enable instruction fetch read access to INTC1. 0: Instruction fetch read access to INTC1 is treated as violation. (Initial value) 1: Instruction fetch read access to INTC1 is not restricted.
1	W0	This bit sets whether to enable write access to INTC1. 0: Write access to INTC1 is treated as violation. (Initial value) 1: Write access to INTC1 is not restricted.
0	R0	This bit sets whether to enable read access to INTC1. 0: Read access to INTC1 is treated as violation. (Initial value) 1: Read access to INTC1 is not restricted.

(6) Peripheral Device Protection Setting Register 3 (IPGPMTUM3)

Bit	7	6	5	4	3	2	1	0
	—	X1	W1	R1	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R	R

Table 14.31 IPGPMTUM3 register contents

Bit Position	Bit Name	Function
7	—	Reserved. This bit is always read as 0. The write value should always be 0.
6	X1	This bit sets whether to enable instruction fetch read access to SysErrGen. 0: Instruction fetch read access to SysErrGen is treated as violation. (Initial value) 1: Instruction fetch read access to SysErrGen is not restricted.
5	W1	This bit sets whether to enable write access to SysErrGen. 0: Write access to SysErrGen is treated as violation. (Initial value) 1: Write access to SysErrGen is not restricted.
4	R1	This bit sets whether to enable read access to SysErrGen. 0: Read access to SysErrGen is treated as violation. (Initial value) 1: Read access to SysErrGen is not restricted
3 to 0	—	Reserved. These bits are always read as 0. The write value should always be 0.

(7) Peripheral Device Protection Setting Register 4 (IPGPMTUM4)

Bit	7	6	5	4	3	2	1	0
	—	X1	W1	R1	—	—	W0	R0
Initial value	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

Table 14.32 IPGPMTUM4 register contents

Bit Position	Bit Name	Function
7	—	Reserved. These bits are always read as 0. The write value should always be 0.
6	X1	This bit sets whether to enable instruction fetch read access to peripheral devices to be connected to the cross-connect XC1. 0: Instruction fetch read access to peripheral devices to be connected to the cross-connect XC1 is treated as violation. (Initial value) 1: Instruction fetch read access to peripheral devices to be connected to the cross-connect XC1 is not restricted.
5	W1	This bit sets whether to enable write access to peripheral devices to be connected to the cross-connect XC1. 0: Write access to peripheral devices to be connected to the cross-connect XC1 is treated as violation. (Initial value) 1: Write access to peripheral devices to be connected to the cross-connect XC1 is not restricted.
4	R1	This bit sets whether to enable read access to peripheral devices to be connected to the cross-connect XC1. 0: Read access to peripheral devices to be connected to the cross-connect XC1 is treated as violation. (Initial value) 1: Read access to peripheral devices to be connected to the cross-connect XC1 is not restricted.
3, 2	—	Reserved. This bit is always read as 0. The write value should always be 0.
1	W0	This bit sets whether to enable write access to PEG. 0: Write access to PEG is treated as violation. (Initial value) 1: Write access to PEG is not restricted.
0	R0	This bit sets whether to enable read access PEG. 0: Read access to PEG is treated as violation. (Initial value) 1: Read access to PEG is not restricted

14.6.4 PBUS Guards (PBG)

The PBG module is divided into several PBG groups, each of which is provided with a maximum of 16 protection channels. A single PBG channel can designate the access against which a single peripheral circuit should be protected. Each PBG group can hold the information of the access that has been rejected.

The PBG groups are defined in Section 14.5, PBUS structure.

14.6.4.1 List of Registers

The following table lists the register provided for each PBG channel.

Module Name	Symbol	Register Name	R/W	Initial Value	Address	Access Size
PBG	FSGDxxPROTn	PBGxx protection register n	R/W	07FF FFFF$_H$	<base_addr0> + 4*n	8/16/32

NOTE

The content and the initial value of the FSGD0BPROTn registers are different to all other FSGDxxPROTn registers. For details, please refer to Section 14.6.4.2, Details of Registers.

The following table lists the registers provided for each PBG group.

Module Name	Symbol	Register Name	R/W	Initial Value	Address	Access Size
PBG	ERRSLVxxCTL	PBGxx error control register	W	0000 0000$_H$	<base_addr1> + 0$_H$	8/16/32
PBG	ERRSLVxxSTAT	PBGxx error status register	R	0000 0000$_H$	<base_addr1> + 4$_H$	8/16/32
PBG	ERRSLVxxADDR	PBGxx error address register	R	0000 0000$_H$	<base_addr1> + 8$_H$	8/16/32
PBG	ERRSLVxxTYPE	PBGxx error type register	R	0000 0000$_H$	<base_addr1> + C$_H$	8/16/32

NOTE

In the header files the names of the above registers are defined in the following format:
<ModuleName> + <Symbol>.
<ModuleName> and <Symbol> are defined in the above tables.

The availability of each PBG register depends on corresponding module availability for each product. See the overview part of each module chapter, for details.
ex) All corresponding registers for PBUS2 are not available in D1L1.

In the above tables, “xx” and “n” in the register names and symbols represents the PBG group numbers and PBG channel numbers, respectively. The table below shows the base address values <base_addr0> and <base_addr1>, which correspond to each of the PBG group numbers and PBG channel numbers.

PBG Group	PBG Group Number xx	PBG Channel Number n	<base_addr0>	<base_addr1>
PBG0A	0A	0 to 4, 6, 7	FFF9 4000_H	FFF9 5000_H
PBG0B	0B	0, 1	FFC4 $C000_H$	FFC4 $C800_H$
PBG1A	1A	0 to 2, 4 to 6	FFF9 $B000_H$	FFF9 $C000_H$
PBG10A	10A	0 to 6	FFF9 $D000_H$	FFF9 $E000_H$
PBG2A	2A	0 to 9	FFFC 7000_H	FFFC 8000_H
PBG2B	2B	1 to 5, 7, 11 to 13	FFFC 9000_H	FFFC $A000_H$
PBG3A	3A	1, 3 to 8	FFDC 0000_H	FFDC 1000_H
PBG3B	3B	0 to 8, 10, 11	FFDC 2000_H	FFDC 3000_H
PBG30A	30A	0 to 15	FFDC 8000_H	FFDC 9000_H
PBG30B	30B	0, 1	FFDC $A000_H$	FFDC $B000_H$
PBG5A	5A	2, 3, 6 to 15	FFF9 0000_H	FFF9 0400_H
PBG5B	5B	0 to 3, 5 to 10, 12	FFF9 0800_H	FFF9 $0C00_H$
PBG5C	5C	0 to 12, 14, 15	FFF9 1000_H	FFF9 1400_H
PBG32A	32A	0 to 15	FFDC 4000_H	FFDC 5000_H
PBG32B	32B	0 to 3	FFDC 6000_H	FFDC 7000_H

14.6.4.2 Details of Registers

(1) FSGDxxPROTn — PBGxx protection register n

FSGDxxPROTn designates the access to be rejected against which the peripheral circuit control registers and RAM should be protected. Any access that is disabled using any of the identifiers is rejected as an illegal access.

NOTE

The content and the initial value of the FSGD0BPROTn registers are different to all other FSGDxxPROTn registers.

(a) FSGD0BPROTn

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PROTLOCK	—	—	—	—	—	PROTUM	PROTPEID[7:0]								VCID7
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VCID[6:0]							PROTSPID[3:0]				PROTDEB	PROTRDPDEF	PROTWRPDEF	PROTRD	PROTWR
Initial value	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(b) All other FSGDxxPROTn (xx ≠ 0B):

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PROTLOCK	—	—	—	—	—	PROTUM	PROTPEID[7:0]								VCID7
Initial value	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VCID[6:0]							PROTSPID[3:0]				PROTDEB	PROTRDPDEF	PROTWRPDEF	PROTRD	PROTWR
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.33 FSGDxxPROTn register contents (1/2)

Bit Position	Bit Name	Function
31	PROTLOCK	Lock of register The PROTLOCK bit allows to block any further change of the register. 0 : Register can be re-written 1 : Any further write to this register is ignored. This bit can only be cleared by reset.
30 to 27	—	Reserved. These bits are always read as 0. The write value should also be 0.
26	—	Reserved. When read, the initial value is read. When written, write the initial value.

Table 14.33 FSGDxxPROTn register contents (2/2)

Bit Position	Bit Name	Function
25	PROTUM	User Mode Access 0: Enables access in supervisor mode. 1: Enables access in user mode and supervisor mode. **Note:** Enabling User Mode Access by FSGDxxPROTn.PROTUM is only useful, if the protected register is accessible in User Mode. For example enabled User Mode Access by PBG0B group to DMA registers may not be useful, because all DMA registers, with exception of DMACER, are not accessible in User Mode (PSW.UM=1).
24 to 17	PROTPEID[7:0]	PEID Access The PROTnPEID field is a bit list, in which one bit corresponds to one PEID value. Setting multiple bits enables the multiple ID values simultaneously. For example, setting the PROTnPEID field to 0101B enables access with PEID = 0 and PEID = 2. 0: Disables access with PEIDn. 1: Enables access with PEIDn. For the PEID assignment refer to Section 14.6.1, SPID and PEID assignment.
16 to 9	VCID[7:0]	VCID Access The PROTnVCID field is a bit list, in which one bit corresponds to one VCID value. Setting multiple bits enables the multiple ID values simultaneously. For example, setting the PROTnVCID field to 0101B enables access with VCID = 0 and VCID = 2. 0: Disables access with VCIDn. 1: Enables access with VCIDn.
8 to 5	PROTSPID[3:0]	SPID Access The PROTnSPID field is a bit list, in which one bit corresponds to one SPID value. Setting multiple bits enables the multiple SPID values simultaneously. For example, setting the PROTnSPID field to 0101B enables access with SPID = 0 and SPID = 2. 0: Disables access with SPIDn. 1: Enables access with SPIDn. For the SPID assignment refer to Section 14.6.1, SPID and PEID assignment.
4	PROTDEB	Debug Access 0: Disables access from the debug master. 1: Enables access from the debug master.
3	PROTRDPDEF	Default Read Protection 0: Enables read access from any master. 1: Only enables read access from the master having permitted access.
2	PROTWRPDEF	Default Write Protection 0: Enables write access from any master. 1: Only enables write access from the master having permitted access.
1	PROTRD	Read Permission 0: Disables read access from any master. 1: Only enables read access from the master having permitted access.
0	PROTWR	Write Permission 0: Disables write access from any master. 1: Only enables write access from the master having permitted access.

(2) ERRSLVxxCTL — PBGxx error control register

ERRSLVxxCTL clears the status in the error status register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRO	CLRE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 14.34 ERRSLVxxCTL register contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved. These bits are always read as 0. The write value should also be 0.
1	CLRO	Writing 1 to this bit clears the OVF flag in ERRSLVxxSTAT. Always read as 0 when read.
0	CLRE	Writing 1 to this bit clears the ERR flag in ERRSLVxxSTAT. Always read as 0 when read.

(3) ERRSLVxxSTAT — PBGxx error status register

ERRSLVxxSTAT holds the status of the illegal access rejected with the PBGxx.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OVF	ERR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.35 ERRSLVxxSTAT register contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved. These bits are always read as 0. The write value should also be 0.
1	OVF	0: When ERR = 1, PBG protection violation has not been detected. 1: When ERR = 1, PBG protection violation has been detected.
0	ERR	0: PBG protection violation has not been detected. 1: PBG protection violation has been detected.

(4) ERRSLVxxADDR — PBGxx error address register

ERRSLVxxADDR holds the address of the illegal access rejected with the PBGxx.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.36 ERRSLVxxADDR register contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Access Address when PBG Protection Violation is Detected ADDR[1:0] are always fixed to 0.

(5) ERRSLVxxTYPE — PBGxx error type register

ERRSLVxxTYPE holds the type of the illegal access rejected with the PBGxx.

NOTE

The content of the ERRSLV0BTYPE register is different to all other ERRSLVxxTYPE registers.

(a) ERRSLV0BTYPE

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PEID[2:0]			VCID[2:0]			SPID[1:0]		—	UM	—	STRB[3:0]				WRITE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.37 ERRSLV0BTYPE register contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved. These bits are always read as 0.
15 to 13	PEID[2:0]	PEID of the access source from which the PBG protection violation is generated.
12 to 10	VCID[2:0]	VCID of the access source from which the PBG protection violation is generated.
9 to 8	SPID[1:0]	SPID of the access source from which the PBG protection violation is generated.
7	—	Reserved. This bit is always read as 0.
6	UM	User mode of the access source from which the PBG protection violation is generated.
5	—	Reserved. This bit is always read as 0.
4 to 1	STRB[3:0]	STRB of the access source from which the PBG protection violation is generated.
0	WRITE	This bit is set to 1 when an access that has generated PBG protection violation is the write.

(b) All other ERRSLVxxTYPE (xx ≠ 0B):

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	SPID[4:0]				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PEID[2:0]			VCID[2:0]			—	—	—	UM	—	STRB[3:0]				WRITE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.38 ERRSLVxxTYPE (xx ≠ 0B) register contents

Bit Position	Bit Name	Function
31 to 21	—	Reserved. These bits are always read as 0.
20 to 16	SPID[4:0]	SPID of the access source from which the PBG protection violation is generated.
15 to 13	PEID[2:0]	PEID of the access source from which the PBG protection violation is generated.
12 to 10	VCID[2:0]	VCID of the access source from which the PBG protection violation is generated.
9 to 7	—	Reserved. These bits are always read as 0.
6	UM	User mode of the access source from which the PBG protection violation is generated.
5	—	Reserved. This bit is always read as 0.
4 to 1	STRB[3:0]	STRB of the access source from which the PBG protection violation is generated.
0	WRITE	This bit is set to 1 when an access that has generated PBG protection violation is the write.

14.6.5 Cross-connect Guards (XCG)

This microcontroller has the following number of units of the Cross-connect Guards XCGn.

Table 14.39 Units

Cross-connect Guard (XCGn)	
Units	13
Names	XCG3 to XCG15

Unit index n

Throughout this section, the individual units of the Cross-connect Guards are identified by the index "n" (n = 3 to 15), for example XCGnCTL for guard n function control register.

Window index i

Each Cross-connect Guard has eight protection windows. Each individual window is identified by the index "i" (i = 1 to 8), for example XCGnCMPi for guard n window i compare register.

Cross-connect Guard to XC slave assignment and register addresses

All Cross-connect Guards register addresses are given as address offsets from the individual base addresses <XCGn_base>.
The <XCGn_base> addresses of each XCGn and its assignment to an individual Cross-connect slave I/F are listed in the following table:

Table 14.40 Register base addresses <XCGn_base>

XCGn unit	<XCGn_base> address	Cross-connect slave
XCG3	FFFC 0000$_H$	Retention RAM (RRAM)
XCG4	FFFC 1000$_H$	Video RAM 0 (VRAM0)
XCG5	FFFC 2000$_H$	Video RAM 1 (VRAM1)
XCG6	FFFC 3000$_H$	Serial Flash Memory I/F 0 (SFMA0)
XCG7	FFFC 4000$_H$	DDR2-SDRAM Memory Controller (SDRB) port 3 (2D Drawing Engine port)
XCG8	FFFC 6000$_H$	Cross-connect XC1 module registers bridge*1
XCG9	FFFC 4400$_H$	DDR2-SDRAM Memory Controller (SDRB) port 2 (XC1 cross-connect port)*2
XCG10	FFFC 4800$_H$	DDR2-SDRAM Memory Controller (SDRB) port 0 (XC0_0 cross-connect port)
XCG11	FFFC 4C00$_H$	DDR2-SDRAM Memory Controller (SDRB) port 1 (XC0_1 cross-connect port)
XCG12	Reserved	
XCG13	FFFC 5400$_H$	Serial Flash Memory I/F 1 (SFMA1)*3
XCG14	FFFC 5800$_H$	HyperBus (HYPB0)*3
XCG15	FFFC 5C00$_H$	OctaBus (OCTA0)*3

Note 1. The configuration registers of some modules are accessed via the XC1 cross-connect.
Note 2. In D1M1H and D1M1A, this channel is used for SDR-SDRAM Memory Controller (SDRA).
Note 3. Only for D1M1-V2, D1M1A.

Cross-connect Guards error signals

Each cross-connect guard can generate two error signals:

- WINTn: guard error during write to the cross-connect slave
- RINTn: guard error during read from the cross-connect slave

Table 14.41 Cross-connect Guards error signals

XCG unit n	Slave	WINTn/RINTn connected to
XCG30 to XCG15	WINT3 to WINT15	Error Control Module INTPERIGRD*1
	RINT3 to RINT15	

Note 1. This Error Control Module input signal is a logical OR combination of various error signals. Refer to Section 48, Error Control Module (ECM) for details.

Reset sources

The Cross-connect Guards and their registers are initialized by the following reset signal:

Table 14.42 Reset sources

XCGn unit	Reset signal
XCGn	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

14.6.5.1 XC Guards function

(1) Enabling the XC Guard

The function of the entire XC Guard XCGn can be enabled via the compare control register:

- XCGnCTL.EN = 1: XCGn enabled

NOTE

Once the XCGn is enabled, it can not be disabled by the application software. Disabling the XCGn is only possible via a reset.

(2) Guard window i set-up

The XC Guard allows to define up to eight windows (i = 1 to 8).
Each guard window i is defined by its address area and access rights. These parameters can be defined separately for each guard window i.

Enabling guard window i

Each guard window i can be separately enable/disabled via the address compare register:

- XCGnCMPi.WINENi = 0: guard window i disabled
- XCGnCMPi.WINENi = 1: guard window i enabled

NOTE

If any of the window i set-up registers shall be changed, the window i must be disabled by XCGnCMPi.WINENi = 0.

CAUTION

Before enabling or disabling the a window i make sure that no master accesses are performed to the window address area.

Definition of the guard window address

An address window is defined by a base address and an address mask.

The base address is defined by the XCGnCMP[31:1] address in the compare register XCGnCMPi. The base address must be aligned to 4 KB, thus the lower 12 address bits must be set to 000_H.

The size of the window is defined by the XCGnVLDi[31:0] in the address valid register XCGnVLDi. The size must be a power of 2, i.e. the respective number of lower bits of XCGnVLDi must all be 0. The address valid register must be set up with the bit-inverted size value, i.e. XCGnVLDi = not Size.

The window addresses are calculated as follows:

- Window start address = XCGnCMPi[31:1], XCGnCMPi[0] = 0
- Window end address = (window start address) or (not XCGnVLDi[31:0])

Following an example for a 32 KB window at address F3F2 0000_H to F3F2 $7FFF_H$:

- XCGnCMPi[31:1] = F3F2 0000_H, (XCGnCMPi[0] = 0)
- XCGnVLDi = not Size = not 0000 $7FFF_H$ = FFFF 8000_H
- Window start address = F3F2 0000_H and FFFF 8000_H = F3F2 0000_H
- Window end address = F3F2 0000_H or (not FFFF 8000_H)
  = F3F2 0000_H or 0000 $7FFF_H$ = F3F2 $7FFF_H$

(3) Access rights setup

All windows i of a guard XCGn can be setup to allow access exclusively in supervisor mode via the compare control register:

- XCGnCTL.UM = 0: access allowed only in supervisor mode
- XCGnCTL.UM = 1: access allowed in supervisor and user mode

Other access rights can be defined separately for each guard window i via the window control register:

- Master access permission

  Each master is identified by its master ID MSTIDk. Refer to Table 14.2, XC0 and XC1 cross-connect master IDs for the IDs of the various masters.
  Each of the MSTIDSEL[11:0] bits in the window control register XCGnCTLWi corresponds to a certain master ID, i.e.

  - MSTIDSEL0 corresponds to MSTID0
  - MSTIDSEL1 corresponds to MSTID1
  - ...

  Each MSTIDSELx bit determines the access right of the respective master:

  - XCGnCTLWi.MSTIDSELx = 0: master with MSTIDx access prohibited
  - XCGnCTLWi.MSTIDSELx = 1: master with MSTIDx access permitted

- Write access permission

– XCGnCTLWi.WMSK = 0:
write access permitted
– XCGnCTLWi.WMSK = 1:
write access prohibited for master IDs set to prohibited by MSTIDSELx

- Read access permission
  – XCGnCTLWi.RMSK = 0:
  read access permitted
  – XCGnCTLWi.RMSK = 1:
  read access prohibited for master IDs set to prohibited by MSTIDSELx

(4) Access violation

Upon detection of an access violation interrupts can be generated:

– XCGnINTL1.WINTEN = 0/1: interrupt WINTn generation disabled/enabled upon write access violation
– XCGnINTL1.RINTEN = 0/1: interrupt RINTn generation disabled/enabled upon read access violation

Assertion of a write or read access violation interrupt can also be monitored by use of interrupt status bits:

– reading XCGnINTL1.WINTSTAT = 1: interrupt WINTn is asserted
– reading XCGnINTL1.RINTSTAT = 1: interrupt RINTn is asserted

Once asserted, an interrupt needs to be de-asserted by the application software:

– writing XCGnINTL1.WINTSTAT = 1: de-assertion of WINTn
– writing XCGnINTL1.RINTSTAT = 1: de-assertion of RINTn

The access address, that led to an access violation, is saved in a register:

– Write error address register XCGnERRWADDR holds the address of the write access violation
– Read error address register XCGnERRRADDR holds the address of the read access violation

Additional information about the access violation is provided by the registers XCGnERRRTYPE for read violation and XCGnERRWTYPE for write violation, which indicates the master, that has caused the violation, and whether it has occurred in super visor or user mode

In case a read or write access violation is detected while the respective interrupt RINTn or WINTn is already asserted (overflow condition) an overflow flag bit is set in the overflow information register XCGnOVF:

– WOVF = 1 indicates a write access violation overflow
writing XCGnOVF.WOVF = 1 clears WOVF
– ROVF = 1 indicates a read access violation overflow
writing XCGnOVF.ROVF = 1 clears ROVF

Note that the address and the additional information of the overflow access violation is not stored in the error address register.

If write access is guarded, the write data of this access will not be written to the slave, and WINTn will be asserted.

If read access is guarded, the read data from the slave will be read out, and RINTn will be asserted.

If the prohibited area overlapped by multiple windows, either one will be valid, and the access will be guarded.

NOTE

When XCGnCTL.EN = 0 the access is to any address area is not guarded.
In the case XCGnCTL.EN = 1 the access is guarded even if the address does not match to any window.

Table 14.43 Guard function matrix

XCGn CTL.EN	XCGn CTL.UM	Access mode	Comparison result of access address for window i	Access permission for window i (XCGnCTLWi. R/WMSK)	Master permission for window i (XCGnCTLWi. MSTIDSEL)	Decision
0	Don't care	Don't care	Don't care	Don't care	Don't care	Through
1	0	User mode	Don't care	Don't care	Don't care	Guard
1	0	Supervisor mode	Match to window 1	0 (permitted)	Don't care	Through
				1 (prohibited to master set for not allowed)	not allowed	Guard
					allowed	Through
1	0	Supervisor mode	Match to window 2	0 (permitted)	Don't care	Through
				1 (prohibited to master set for not allowed)	not allowed	Guard
					allowed	Through
1	0	Supervisor mode	...			
1	0	Supervisor mode	Match to window 8	0 (permitted)	Don't care	Through
				1 (prohibited to master set for not allowed)	not allowed	Guard
					allowed	Through
1	0	Supervisor mode	No match to all windows	Don't care	Don't care	Guard
1	1	User or supervisor mode	Match to window 1	0 (permitted)	Don't care	Through
				1 (prohibited to master set for not allowed)	not allowed	Guard
					allowed	Through
1	1	User or supervisor mode	Match to window 2	0 (permitted)	Don't care	Through
				1 (prohibited to master set for not allowed)	not allowed	Guard
					allowed	Through
1	1	User or supervisor mode	...			
1	1	User or supervisor mode	Match to window 8	0 (permitted)	Don't care	Through
				1 (prohibited to master set for not allowed)	not allowed	Guard
					allowed	Through
1	1	User or supervisor mode	No match to all windows	Don't care	Don't care	Guard

(5) Set-up flow example

(1) Set up the base address in XCGnCMPi.CMP[31:1].

(2) Enable the window i by XCGnCMPi.WINEN = 1.

(3) Set up the window size in the XCGnVLDi.

(4) Grant access to the masters by setting the respective XCGnCTLWi.MSTIDSEL[11:0] bits.

(5) Grant write and/or read access by setting XCGnCTLWi.WMSK = 1 and/or XCGnCTLWi.RMSK = 1.

(6) Enable the generation of write and/or read access violation interrupts WINTn and RINTn by setting XCGnINTL1.WINTEN = 1 and/or XCGnINTL1.RINTEN = 1.

(7) Select whether the XCGn shall guard accesses in only in supervisor or also in user mode by XCGnCTL.UM.

(8) Enable the XCGn by setting XCGnCTL.EN = 1.

14.6.5.2 XC Guards registers overview

The XC Guards are controlled and operated by the following registers:

Table 14.44 XC Guards registers overview

Register name	Shortcut	Address
Address compare registers	XCGnCMPi	<XCGn_base> + (i-1) x 4_H
Address valid registers	XCGnVLDi	<XCGn_base> + 20_H + (i-1) x 4_H
Compare control register	XCGnCTL	<XCGn_base> + 40_H
Window control registers	XCGnCTLWi	<XCGn_base> + 44_H + (i-1) x 4_H
Interrupt control register	XCGnINTL1	<XCGn_base> + 64_H
Overflow condition register	XCGnOVF	<XCGn_base> + 68_H
Read error address register	XCGnERRRADDR	<XCGn_base> + $6C_H$
Write error address register	XCGnERRWADDR	<XCGn_base> + 70_H
Read error type register	XCGnERRRTYPE	<XCGn_base> + 74_H
Write error type register	XCGnERRWTYPE	<XCGn_base> + 78_H

<XCGn_base>

The base addresses <XCGn_base> of the XCGn is defined in the first subsection of this section under the key word “Register addresses”.

NOTE

Memory access right after register access has possibility to cause the racing between two accesses via different routes.
Therefore, the last accessed register value must be confirmed at least before memory access. Refer to Section 3.4.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation for details.

14.6.5.3 XC Guard registers details

(1) XCGnCMPi - Address compare registers

This register enables the guard window i and defines its base address.

CAUTIONS

1. **Before enabling or disabling the a window i make sure that no master accesses are performed to the window address area.**
2. **Before changing XCGnCMPi.CMP[31:1] the window i must be disabled by XCGnCMPi.WINENi = 0.**

Access: This register can be accessed in 32-bit units.

Address: <XCGn_base> + (i-1) x 4_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CMP[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMP[15:1]															WINEN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.45 XCGnCMPi register contents

Bit position	Bit name	Function
31 to 1	CMP[31:1]	Upper 31 bit of the window i base address. **CAUTION** The window base address must be aligned to 4 KB, thus the lower 11 address bits must be set to 000_H.
0	WINEN	Enable/disable guard window i. 0: window i disabled 1: window i enabled

(2) XCGnVLDi - Address valid registers

This register defines the size of the guard window i.

CAUTION

Before changing this register the window i must be disabled by XCGnCMPi.WINENi = 0.

Access: This register can be accessed in 32-bit units.

Address: <XCGn_base> + 20_H + (i-1) x 4_H

Initial value: FFFF $FFFE_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VLD[31:16]															
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VLD[15:1]															0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 14.46 XCGnVLDi register contents

Bit position	Bit name	Function
31 to 1	VLD[31:1]	VLD[31:1] determines of the size of the guard window i. The lowest address bit VLD0 is always 0. It must be set to the bit-inversion of the size value, i.e. VLD[31:0] = not (size - 1). **CAUTION** The minimum window size is 4 KB, thus VLD[11:0] must be set to 000_H.
0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

(3) XCGnCTL - Compare control register

This register is used to enable/disable the XCGn and to define supervisor/user mode access rights.

Access: This register can be accessed in 32-bit units.

Address: <XCGn_base> + 40_H

Initial value: 8000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	UM	EN	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.47 XCGnCTL register contents

Bit position	Bit name	Function
31	UM	User and supervisor mode control 0: Access allowed only in supervisor mode 1: Access allowed in supervisor and user mode
30	EN	XC Guard enable/disable • At EN bit read: XCGn status 0: XCGn is disabled 1: XCGn is enabled • At EN write: enable XCGn 0: no function 1: enable XCGn **CAUTION** Once the XCGn is enabled, it can not be disabled by the application software. Disabling the XCGn is only possible via a reset.
29 to 0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

(4) XCGnCTLWi - Window control register

This register is used to define window specific access rights.

CAUTION

Before changing this register the window i must be disabled by XCGnCMPi.WINENi = 0.

Access: This register can be accessed in 32-bit units.

Address: <XCGn_base> + 44_H + (i-1) x 4_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	MSTIDSEL[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WMSK	RMSK
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 14.48 XCGnCTLWi register contents

Bit position	Bit name	Function
31 to 28	Reserved	When read, the value after reset is returned. When written, write the value after reset.
27 to 16	MSTIDSEL[11:0]	Master ID selection for access permission Each MSTIDSELx bit corresponds to a certain master, i.e. MSTIDSEL0: Controls access with master ID MSTID0 MSTIDSEL1: Controls access with master ID MSTID1 ... MSTIDSEL11: Controls access with master ID MSTID11 0: Access by MSTID[11:0] not allowed 1: Access by MSTID[11:0] allowed
15 to 2	Reserved	When read, the value after reset is returned. When written, write the value after reset.
1	WMSK	Write access permission 0: write permitted 1: write prohibited for master IDs set to not allowed by MSTIDSELx
0	RMSK	Read access permission 0: read permitted 1: read prohibited for master IDs set to not allowed by MSTIDSELx

(5) XCGnINTL1 - Interrupt control register

This register is used to control access violation interrupt generation and holds information flags about these interrupts.

Access: This register can be accessed in 32-bit units.

Address: <XCGn_base> + 64_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	WINT EN	WINT STAT	RINT EN	RINT STAT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 14.49 XCGnINTL1 register contents

Bit position	Bit name	Function
31 to 4	Reserved	When read, the value after reset is returned. When written, write the value after reset.
3	WINTEN	Write guard interrupt WINTn enable bit. 0: WINTn disabled 1: WINTn enabled
2	WINTSTAT	Write guard interrupt WINTn status and clear bit. • At XCGnINTL1 register read: WINTn status 0: WINTn has not occurred 1: WINTn has occurred • At XCGnINTL1 register write: WINTn clear 0: no function 1: clear this bit and WINTn de-assert interrupt
1	RINTEN	Read guard interrupt RINTn enable bit. 0: RINTn disabled 1: RINTn enabled
0	RINTSTAT	Read guard interrupt RINTn status and clear bit. • At XCGnINTL1 register read: RINTn status 0: RINTn has not occurred 1: RINTn has occurred • At XCGnINTL1 register write: RINTn clear 0: no function 1: clear this bit and RINTn de-assert interrupt

(6) XCGnOVF - Overflow information register

This register indicates occurrence of access violation overflows.

Access: This register can be accessed in 32-bit units.

Address: <XCGn_base> + 68_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WOVF	ROVF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 14.50 XCGnOVF register contents

Bit position	Bit name	Function
31 to 2	Reserved	When read, the value after reset is returned. When written, write the value after reset.
1	WOVF	Write access violation overflow flag • At read: 0: write access violation did not occur during WINTn assertion 1: write access violation occurred during WINTn assertion • At write: 0: no function 1: clear WOVF
0	ROVF	Read access violation overflow flag • At read: 0: read access violation did not occur during RINTn assertion 1: read access violation occurred during RINTn assertion • At write: 0: no function 1: clear ROVF

(7) XCGnERRRADDR - Read error address register

This register stores the address during a read access violation.

Access: This register can be accessed in 32-bit units.

Address: <XCGn_base> + $6C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ERRRADDR[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ERRRADDR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.51 XCGnERRRADDR register contents

Bit position	Bit name	Function
31 to 0	ERRRADDR[31:0]	Address of the read access violation. ERRRADDR[31:0] does not change when a read access violation overflow occurs. Thus it holds always the address of the first access violation, until the read access violation interrupt RINTn is de-asserted by setting XCGnINTL1.RINTSTAT = 1.

(8) XCGnERRWADDR - Write error address register

This register stores the address during a write access violation.

Access: This register can be accessed in 32-bit units.

Address: <XCGn_base> + 70_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ERRWADDR[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ERRWADDR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.52 XCGnERRWADDR register contents

Bit position	Bit name	Function
31 to 0	ERRWADDR[31:0]	Address of the write access violation. ERRWADDR[31:0] does not change when a write access violation overflow occurs. Thus it holds always the address of the first access violation, until the write access violation interrupt WINTn is de-asserted by setting XCGnINTL1.WINTSTAT = 1.

(9) XCGnERRRTYPE - Read error type register

This register provides addition information about a read access violation detection.

Access: This register can be accessed in 32-bit units.

Address: <XCGn_base> + 74_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	ERRRUM	ERRRMST[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.53 XCGnERRRTYPE register contents

Bit position	Bit name	Function
31 to 9	Reserved	When read, the value after reset is returned.
8	ERRRUM	This bit indicates whether the read access violation has occurred during supervisor or user mode. 0: violation occurred in supervisor mode 1: violation occurred in user mode
7 to 0	ERRRMST[7:0]	Master ID of read access violation. Refer to Table 14.54, XC Guard access violation master IDs for the correspondence between ERRRMST[7:0], the XC master and XCGnCTLWi.MSTIDSEL[11:0] bit, that sets the access right to the master. ERRRMST[7:0] does not change when a read access violation overflow occurs. Thus it holds always the address of the first access violation, until the read access violation interrupt RINTn is de-asserted by setting XCGnINTL1.RINTSTAT = 1.

Table 14.54 XC Guard access violation master IDs

Master		Master identifier in XCGnERRRTYPE and XCGnERRWTYPE registers ERRRMST[7:0]/ERRWMST[7:0]	Master ID selection bit in XCGnCTLWi register MSTIDSEL[11:0]
Debug unit		$0000\ 0000_B$	MSTIDSEL0
CPU		$0000\ 0001_B$	MSTIDSEL1
DMA Controller		$0000\ 0010_B$	MSTIDSEL2
Reserved		$0000\ 0011_B$	MSTIDSEL3
2D Graphics Processing Unit (GPU2D)		$0001\ 0000_B$	MSTIDSEL4
JPEG Codec Unit A (JCUA)		$0010\ 0000_B$	MSTIDSEL5
Ethernet AVB MAC (ETNB)		$0011\ 0000_B$	MSTIDSEL6
Media Local Bus I/F (MLBB) NAND Flash Memory I/F A (NFMA)*1		$0100\ 0000_B$	MSTIDSEL7
Sprite Engine (SPEA)	RLE units	$0101\ 0000_B$	MSTIDSEL8
	Sprite Unit 0	$0101\ 0001_B$	MSTIDSEL9
	Sprite Unit 1	$0101\ 0010_B$	MSTIDSEL10
	Sprite Unit 2	$0101\ 0011_B$	MSTIDSEL11

Note 1. Only for D1M1A.

(10) XCGnERRWTYPE - Write error type register

This register provides addition information about a write access violation detection.

Access: This register can be accessed in 32-bit units.

Address: <XCGn_base> + 78_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	ERRWUM	ERRWMST[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 14.55 XCGnERRWTYPE register contents

Bit position	Bit name	Function
31 to 9	Reserved	When read, the value after reset is returned.
8	ERRWUM	This bit indicates whether the write access violation has occurred during supervisor or user mode. 0: violation occurred in supervisor mode 1: violation occurred in user mode
7 to 0	ERRWMST[7:0]	Master ID of write access violation. Refer to Table 14.54, XC Guard access violation master IDs for the correspondence between ERRWMST[7:0], the XC master and XCGnCTLWi.MSTIDSEL[11:0] bit, that sets the access right to the master. ERRWMST[7:0] does not change when a write access violation overflow occurs. Thus it holds always the address of the first access violation, until the write access violation interrupt WINTn is de-asserted by setting XCGnINTL1.WINTSTAT = 1.

14.6.5.4 XC Guard Operation

It is recommended to setup all XC Guards to deny slave access from any master by default in order to prevent unexpected faults.

- Setup all Guards including those of unavailable slaves (e.g. SDRAM on D1M1).
- Setup of a single window for the slaves full address window is sufficient for basic setup. Several windows may be configured to comply with specific safety and security requirements.
- Deny access by any master.
- Allow those masters expected to access the specific slave.

This basic XC Guard setup is an effective measure to prevent unintended reading or manipulation of data.

14.7 Performance Monitor

This function is only available in D1M1A and D1M1-V2, or when using IE-POD (RTE7701460EPA00000R).

The Performance Monitor has the following number of units of the Performance-Monitor PM_xx for measuring the XC slave performance.

Table 14.56 Performance Monitor units

Unit	<PM_xx_base> address	Performance Monitor for
PM_R	FFFC C000$_H$	Retention RAM (RRAM)
PM_SF0	FFFC C100$_H$	Serial Flash Memory I/F 0 (SFMA0)
PM_SF1	FFFC C200$_H$	Serial Flash Memory I/F 1 (SFMA1)
PM_V0	FFFC C300$_H$	Video RAM 0 (VRAM0)
PM_V1	FFFC C400$_H$	Video RAM 1 (VRAM1)
PM_SD	FFFC C600$_H$	SDR-SDRAM Memory Controller (SDRA0)
PM_HB	FFFC CA00$_H$	HyperBus/OctaBus Controller (HYPB0/OCTA0)

The Performance Monitor can measure the following items. Also, this monitor can measure for some masters at the same time.

- request/data cycle measurement
  - read request count for a period as specified below
  - write request count for a period as specified below
  - read data count for a period as specified below
  - write data count for a period as specified below
  - clock cycle count (time elapsed) for a period as specified below
- configurable measurement periods
  - period until read request count reaches the user defined threshold
  - period until write request count reaches the user defined threshold
  - period until read data count reaches the user defined threshold
  - period until write data count reaches the user defined threshold
  - period until clock cycle count reaches the user defined threshold

14.7.1 Performance Monitor registers

The Performance Monitors are controlled and operated by the following registers:

Table 14.57 Performance Monitor registers overview (1/2)

Register name	Shortcut	Address
Measurement enable register	PMCNTEN	<PM_xx_base> + 00$_H$
Base counter select register	PMBSCNTSEL	<PM_xx_base> + 04$_H$
Stop condition threshold register	PMDCNTSET	<PM_xx_base> + 08$_H$
Read ID register	PMRIDSET	<PM_xx_base> + 0C$_H$
Write ID register	PMWIDSET	<PM_xx_base> + 10$_H$
Read request count register	PMRQRDCNT	<PM_xx_base> + 14$_H$
Write request count register	PMRQWTCNT	<PM_xx_base> + 18$_H$

Table 14.57 Performance Monitor registers overview (2/2)

Register name	Shortcut	Address
Read transferred data count register	PMDTRDCNT	<PM_xx_base> + $1C_H$
Write transferred data count register	PMDTWTCNT	<PM_xx_base> + 20_H
Clock count register	PMCKCNT	<PM_xx_base> + 24_H

14.7.1.1 PMCNTEN – Measurement enable register

Access: This register can be read/written in 32-bit units.

Address: <PM_xx_base> + 00_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	CNTEN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.58 PMCNTEN register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	CNTEN	Performance Monitor control 0: Stop measurement (stopped) 1: Start measurement (measuring) When the measurement is finished, CNTEN is set to 0. When CNTEN is set from 0 to 1 by write access, the measurement is started. When CNTEN is set from 1 to 0 by write access, the measurement is stopped. Reading CNTEN gives current status.

14.7.1.2 PMBSCNTSEL – Base counter select register

Access: This register can be read/written in 32-bit units.

Address: <PM_xx_base> + 04_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	BSCNTSEL[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.59 PMBSCNTSEL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, these bits are always read as 0. When writing, always write 0.
2 to 0	BSCNTSEL[2:0]	Base count condition for measurement stop 000_B: read request count 001_B: write request count 010_B: read data count 011_B: write data count All others: clock count

14.7.1.3 PMDCNTSET – Stop condition threshold register

Access: This register can be read/written in 32-bit units.

Address: <PM_xx_base> + 08_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCNTSET[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCNTSET[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.60 PMDCNTSET register contents

Bit Position	Bit Name	Function
31 to 0	DCNTSET[31:0]	The threshold of stop condition

CAUTIONS

1. **Don't set DCNTSET[31:0] under 20_H.**
2. **Don't set PMCNTEN.CNTEN to 1 with DCNTSET[31:0] = 0.**

14.7.1.4 PMRIDSET – Read ID register

Access: This register can be read/written in 32-bit units.

Address: <PM_xx_base> + $0C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RIDVAL	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	RIDSET[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.61 PMRIDSET register contents

Bit Position	Bit Name	Function
31	RIDVAL	The valid of Read ID comparison 0: ID comparison disable 1: ID comparison enable When this bit is 0, all master are measured.
30 to 4	Reserved	When read, these bits are always read as 0. When writing, always write 0.
3 to 0	RIDSET[3:0]	Comparison target for Read ID When RIDVAL = 1, Performance Monitor only counts events with a Read ID matching the value of RIDSET[3:0]. [When using IE-POD] 0: CPU Subsystem (DMA, OCD) 1: CPU Subsystem (CPU) 2: JCUA 3: ETNB 4: MLBB 5: RLE/Sprite Units 0 and 1 6: RLE/Sprite Units 2 and 3 7: GPU2D (D1M2(H) emulation) 8: GPU2D (other emulation) 9: NFMA [D1M1A] 0: CPU Subsystem (DMA, OCD) 1: CPU Subsystem (CPU) 2: RLE/Sprite Unit 0 and 1 3: RLE/Sprite Unit 2 and 3 4: JCUA 5: GPU2D 6: NFMA 7: ETNB [D1M1-V2] 0: CPU Subsystem (DMA, OCD) 1: CPU Subsystem (CPU) 2: RLE/Sprite Unit 0 and 1 3: RLE/Sprite Unit 2 and 3 4: JCUA 5: GPU2D 6: ETNB

14.7.1.5 PMWIDSET – Write ID register

Access: This register can be read/written in 32-bit units.

Address: <PM_xx_base> + 10_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WIDVAL	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	WIDSET[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.62 PMWIDSET register contents

Bit Position	Bit Name	Function
31	WIDVAL	The valid of Write ID comparison 0: ID comparison disable 1: ID comparison enable When this bit is 0, all master are measured.
30 to 4	Reserved	When read, these bits are always read as 0. When writing, always write 0.
3 to 0	WIDSET[3:0]	Comparison target for Write ID When WIDVAL = 1, Performance Monitor only counts events with a Write ID matching the value of WIDSET[3:0]. [When using IE-POD] 0: CPU Subsystem (DMA, OCD) 1: CPU Subsystem (CPU) 2: JCUA 3: ETNB 4: MLBB 5: RLE/Sprite Units 0 and 1 6: RLE/Sprite Units 2 and 3 7: GPU2D (D1M2(H) emulation) 8: GPU2D (other emulation) 9: NFMA [D1M1A] 0: CPU Subsystem (DMA, OCD) 1: CPU Subsystem (CPU) 2: RLE/Sprite Unit 0 and 1 3: RLE/Sprite Unit 2 and 3 4: JCUA 5: GPU2D 6: NFMA 7: ETNB [D1M1-V2] 0: CPU Subsystem (DMA, OCD) 1: CPU Subsystem (CPU) 2: RLE/Sprite Unit 0 and 1 3: RLE/Sprite Unit 2 and 3 4: JCUA 5: GPU2D 6: ETNB

14.7.1.6 PMRQRDCNT – Read request count register

Access: This register can be read/written in 32-bit units.

Address: <PM_xx_base> + 14_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RQRDCNT[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RQRDCNT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.63 PMRQRDCNT register contents

Bit Position	Bit Name	Function
31 to 0	RQRD CNT[31:0]	Read request count register This register is counted up when the measurement condition is matched. • Stop condition – Selected base count reaches the stop condition threshold – PMCNTEN.CNTEN is set from 1 to 0 – Overflow of this register (counter value is stopped at all 1) • Clear condition – PMCNTEN.CNTEN is set from 0 to 1

14.7.1.7 PMRQWTCNT – Write request count register

Access: This register can be read/written in 32-bit units.

Address: <PM_xx_base> + 18_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RQWTCNT[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RQWTCNT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.64 PMRQWTCNT register contents

Bit Position	Bit Name	Function
31 to 0	RQWT CNT[31:0]	Write request count register This register is counted up when the measurement condition is matched. • Stop condition – Selected base count reaches the stop condition threshold – PMCNTEN.CNTEN is set from 1 to 0 – Overflow of this register (counter value is stopped at all 1) • Clear condition – PMCNTEN.CNTEN is set from 0 to 1

14.7.1.8 PMDTRDCNT – Read transferred data count register

Access: This register can be read/written in 32-bit units.

Address: <PM_xx_base> + $1C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTRDCNT[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTRDCNT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.65 PMDTRDCNT register contents

Bit Position	Bit Name	Function
31 to 0	DTRDCNT[31:0]	Read data count register This register is counted up when the measurement condition is matched. • Stop condition – Selected base count reaches the stop condition threshold – PMCNTEN.CNTEN is set from 1 to 0 – Overflow of this register (counter value is stopped at all 1) • Clear condition – PMCNTEN.CNTEN is set from 0 to 1

14.7.1.9 PMDTWTCNT – Write transferred data count register

Access: This register can be read/written in 32-bit units.

Address: <PM_xx_base> + 20_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DTWTCNT[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DTWTCNT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.66 PMDTWTCNT register contents

Bit Position	Bit Name	Function
31 to 0	DTWTCNT[31:0]	Write data count register This register is counted up when the measurement condition is matched. • Stop condition – Selected base count reaches the stop condition threshold – PMCNTEN.CNTEN is set from 1 to 0 – Overflow of this register (counter value is stopped at all 1) • Clear condition – PMCNTEN.CNTEN is set from 0 to 1

14.7.1.10 PMCKCNT – Clock count register

Access: This register can be read/written in 32-bit units.

Address: <PM_xx_base> + $1C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CKCNT[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CKCNT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.67 PMCKCNT register contents

Bit Position	Bit Name	Function
31 to 0	CKCNT[31:0]	Clock count register This register is counted up with the cross-connect clock C_ISO_XCCLK. • Stop condition – Selected base count reaches the stop condition threshold – PMCNTEN.CNTEN is set from 1 to 0 – Overflow of this register (counter value is stopped at all 1) • Clear condition – PMCNTEN.CNTEN is set from 0 to 1

14.7.2 Measurement operation

14.7.2.1 Register setting

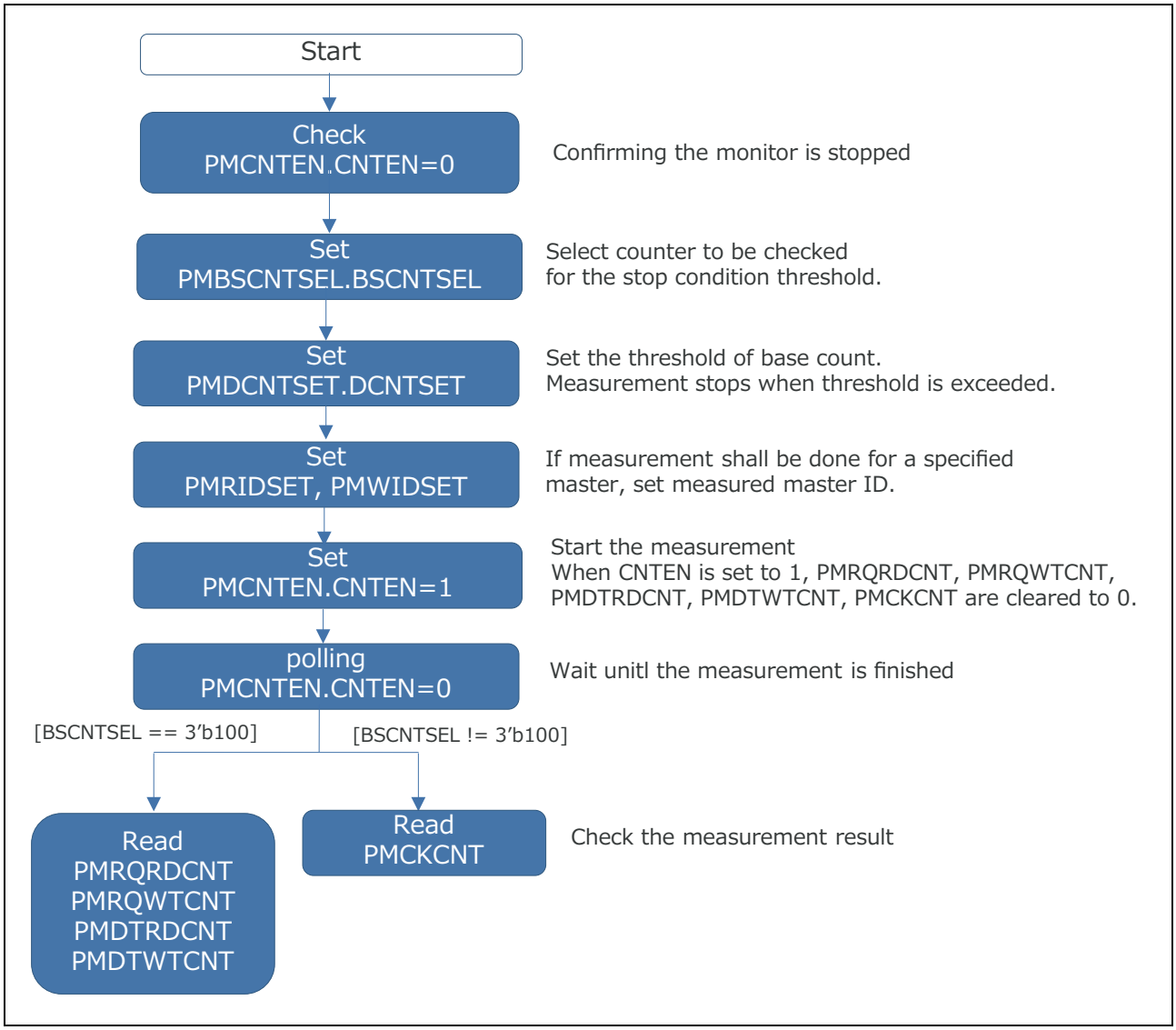


Figure 14.29 Performance Monitor register setting

14.7.2.2 Calculating the bandwidth

In order to obtain the bandwidth from the measurement result, calculation is needed from register value. The following shows an example of a bandwidth calculation.

Example

- Measurement condition
  - XC bus clock C_ISO_XCCLK: 120 MHz [= 8.33 ns]
  - Measurement type: period (PMBSCNTSEL.BSCNTSEL[2:0] = 100_B)
  - BSCNTSEL[2:0] == 100_B
    To define a measurement period of 1s:
    PMDCNTSET = C_ISO_XCCLK / t_measurement = 120MHz / 1s = 727 $0E00_H$
  - BSCNTSEL[2:0] != 100_B
    To calculate the measurement period form the clock count register:
    t_measurement = C_ISO_XCCLK / PMCKCNT
- Measurement result
  - PMDTRDCNT : 2AE $A540_H$
  - PMDTWTCNT : E4 $E1C0_H$
- Bandwidth calculation
  - Read Bandwidth: 360 MB/s ((2AE $A540_H$ x 8 byte) / 1 s))
  - Write Bandwidth: 120 MB/s ((E4 $E1C0_H$ x 8 byte / 1 s))

PMDTRDSET, PMDTWTCNT count valid data cycles on the cross-connect bus. Therefore, for calculating the amount of data, it is required to multiple by 8 byte (due to 64-bit XC bus width). This bandwidth is the view from the slave side.

NOTE

The CPU can access XC slaves with byte, half-word, word, double-word size.
In order to check the bandwidth from the view of CPU side, bandwidth calculation must be changed to respect the access size of the currently executed instructions.

14.8 Preload Buffer (PRLBn) (D1M1-V2, D1M1A only)

The Preload Buffer pre-fetches read data for XC2 masters.

14.8.1 Features

One Preload Buffer comprises 3 separate 3-entry sub-buffers, which can be assigned to different XC2 masters.

- Preloading
  - PRLB starts to issue 3 read requests (1 currently requested + 2 preload request) by the first read request from the XC2 master.
  - PRLB returns the currently requested data to the requesting XC2 master and issues 2 speculative requests in addition.
  - PRLB keeps issuing 3 request.
- Preload Conditions
  - Preloading enable by register setting
    When it is disabled, transactions go through without delay. (default setting)
  - Master assignment by register setting
    Requests of un-assigned master are not preloaded.
  - Preload is executed by 128 byte data request
    Requests shorter than 128 byte are not preloaded, even if the master is assigned.
    Because such preloading is not effective to performance.
- Preloading Cancel Condition
  When the following conditions occur, preloading is canceled.
  - Non-consecutive address
    When addresses from the master side are not consecutive, preloading is canceled.
  - Shorter data request
    When requested data from master side is shorter than 128 byte, preloading is canceled.
  - Timeout
    When the master side doesn't request the preloaded data before the timeout expires, preloading is canceled.

14.8.2 Preload Buffers registers

The Preload Buffers are controlled and operated by the following registers:

Table 14.68 Preload Buffers registers overview

Register name	Shortcut	Address
Preload buffer 0 register	PRL0REG	F200 A000$_H$
Preload buffer 1 register	PRL1REG	F200 A004$_H$
Preload buffer 2 register	PRL2REG	F200 A008$_H$
Preload buffer 3 register	PRL3REG	F200 A00C$_H$

14.8.2.1 PRLnREG – Preload buffer n register (n = 0 to 3)

Access: This register can be read/written in 32-bit units.

Address: PRL0REG: F200 A000$_H$
PRL1REG: F200 A004$_H$
PRL2REG: F200 A008$_H$
PRL3REG: F200 A00C$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	PRLTO[3:0]				–	–	–	–	–	–	PRLISS[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PRLID2[3:0]				PRLID1[3:0]				PRLID0[3:0]				–	–	–	PRLEN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14.69 PRLnREG register contents

Bit Position	Bit Name	Function
31 to 28	Reserved	When read, these bits are always read as 0. When writing, always write 0.
27 to 24	PRLTO[3:0]	Timeout cycles When the master side doesn't the request in spite of preloaded data reached to PRLB buffer,PRLB cancel preloading. 0: Timeout = 15 XC clock cycles Others: Setting prohibit
23 to 18	Reserved	When read, these bits are always read as 0. When writing, always write 0.
17 to 16	PRLISS[1:0]	Issuing request number 10_B: 3 requests Others: Setting prohibit
15 to 12	PRLID2[3:0]	Preload ID2 for assigning an XC2 master to sub-buffer 2 of Preload Buffer n About the ID assignment, refer to Section 14.8.2.2, PRLB setting.
11 to 8	PRLID1[3:0]	Preload ID1 for assigning an XC2 master to sub-buffer 1 of Preload Buffer n About the ID assignment, refer to Section 14.8.2.2, PRLB setting.
7 to 4	PRLID0[3:0]	Preload ID0 for assigning an XC2 master to sub-buffer 0 of Preload Buffer n About the ID assignment, refer to Section 14.8.2.2, PRLB setting.
3 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	PRLEN	Preload function enable 0: disable (default) 1: enable In the case of PRLEN = 0, read request and data bypass preload bridge without latency.

14.8.2.2 PRLB setting

PRLnREG.PRLEN has to be set with following initialisation:

- Set PRLnREG.PRLTO[3:0] to 0xF
- Set PRLnREG.PRLISS[1:0] to 0x2
- Set PRLnREG.PRLIDm to
  - m if you want to use PRLB.
  - 15 if you don't want to use PRLB.
  - Example:
    To use PRLB for Image Synthesizer10, set PRLB0REG.PRLID1[3:0] to 0x1
    To use PRLB for Image Synthesizer03, set PRLB3REG.PRLID0[3:0] to 0x0
- Set PRLEN to 1

PRLB has to be set before Video output start.
Also, it is prohibit to change PRLB setting while video outputs are active.

Each of the PRLB Units covers up to three specific Video Output Units.
The table below explains the assignment of the different units.
The initialization as listed above can be done with a single write access to the registers PRLnREG.

Table 14.70 PRLB units assignments

	Unit PRLB0	Unit PRLB1	Unit PRLB2	Unit PRLB3	PRLID value
for D1M1A:					
Instance 0 [PRLID0[3:0]]	Image Synthesizer 00	Image Synthesizer 01	Image Synthesizer 02	Image Synthesizer 03	Set PRLID0[3:0] to 0 to enable this instance, set to F_H to disable this instance.
Instance 1 [PRLID1[3:0]]	Image Synthesizer 10	Image Synthesizer 13	Image Synthesizer 12	Image Synthesizer 11	Set PRLID1[3:0] to 1 to enable this instance, set to F_H to disable this instance.
Instance 2 [PRLID2[3:0]]	– *1	Output Image Generator	– *1	– *1	Set PRLID2[3:0] to 2 to enable this instance, set to F_H to disable this instance.
for D1M1V2:					
Instance 0 [PRLID0[3:0]]	Image Synthesizer 00	Image Synthesizer 01	Image Synthesizer 02	Image Synthesizer 03	Set PRLID0[3:0] to 0 to enable this instance, set to F_H to disable this instance.
Instance 1 [PRLID1[3:0]]	– *1	Output Image Generator	– *1	– *1	Set PRLID1[3:0] to 1 to enable this instance, set to F_H to disable this instance.
Instance 2 [PRLID2[3:0]]	– *1	– *1	– *1	– *1	Set PRLID2[3:0] to F_H to disable this instance.

Note 1. Aways set to disable: 0xF

NOTE

PRLID value must not be the same as other PRLID except 0xF in the same unit.

14.8.3 Preload Buffers use cases

PRLB can be used for improving the performance of the XC2 masters. However, in some use cases, the preloading may be ineffective.

GPU2D

When the GPU2D is used, number of outstanding requests of XC2 masters is increased to be on level with GPU2D.

However, if more than four masters, which are reading from the same memory slave, are using preloading, the performance is almost same with and without PRLB, because the request number on the XC cross-connect is already sufficient. Therefore, preloading is better to be used for a few layers with GPU2D.

Sprite Layer

If sprite layers are used, preloading may be ineffective. By using many sprites, addressing from the XC master is sometimes not consecutive in a layer.

Also, when the boundary of sprite is not aligned to 128 byte, XC master data requests for a small amout of data are generated from sprite unit.

In both cases, preloading is canceled and already issued requests for preloading are redundant and may cause additional overhead.

Slow memory access (SFMA, HyperBus/OctaBus)

Preloading is not recommended to be used for memories with high read access latencies.
Due to the high initial overhead of a read access, preloading has only small influence.

However, many requests queued for a high latency memory may also affect access latency to other fast memories. This effect is caused by some requests waiting for completion of previously issued requests in situations of high bus load.

About other control methods for the bus traffic, refer Section 14.2.1.1, List of XC1 QoS Registers (D1M1(H), D1M1-V2, D1M1A only), Section 15.5, SDR-SDRAM Transaction Restrictor and Section 54.3, VRAM Transaction Restrictor.

14.8.4 Video frame buffer and PRLB usage

PRLB always issues 2 requests (256 byte read) as preloading with a current request to memory.

In addition, video frame buffer include additional 256 byte area at the end of frame buffer by VDCE restriction.

(Regarding VDCE restriction, refer to Section 38.6.1.2, Graphics Data Read Control **(6)**.)

Therefore, video frame buffer must include total 512 byte readable area at the end of frame buffer when PRLB is used.

Section 15 SDR-SDRAM Memory Controller (SDRA)

This section contains a generic description of the SDR-SDRAM Memory Controller.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

15.1 Overview of RH850/D1L/D1M SDR-SDRAM Memory Controller

15.1.1 Units

This microcontroller has the following number of units of the SDR-SDRAM Memory Controller.

Table 15.1 Units

SDR-SDRAM Memory Controller	D1L1	D1L2(H)	D1M1 D1M1-V2	D1M1H D1M1A	D1M2(H)
Units	–	–	–	1	–
Names	–	–	–	SDRA0	–

Units index n

Throughout this section, the individual units of a SDR-SDRAM Memory Controller is identified by the index "n" (n = 0).

15.1.2 Register addresses

All SDR-SDRAM Memory Controller register addresses are given as address offsets from the individual base addresses <SDRAn_base>.
The <SDRAn_base> addresses of each SDRAn are listed in the following table:

Table 15.2 Register base addresses <SDRAn_base>

SDRAn unit	<SDRAn_base> address
SDRA0	F2FE 0000_H

15.1.3 Clock supply

All SDR-SDRAM Memory Controller provides two clock inputs.

Table 15.3 Clock supply

SDRAn unit	SDRAn clock	Connected to
SDRA0	Cross-connect clock	Clock Controller C_ISO_XCCLK
	MEMCLK	Clock Controller SDRBCLK

NOTE

The SDRACK memory clock signal output frequency equals to the Cross-connect clock (= C_ISO_XCCLK).

15.1.4 Interrupts

The SDR-SDRAM Memory Controller can generate the following interrupts:

Table 15.4 SDRAn interrupts

ADRAn signals	Function	Connected to
SDRA0:		
INTSDRA	Underflow flag of auto refresh cycle	Interrupt Controller INTSDRA

15.1.5 Reset sources

The SDR-SDRAM Memory Controller and their registers are initialized by the following reset signal:

Table 15.5 Reset sources

SDRAn unit	Reset signal
SDRAn	• Reset Controller SYSRES • Reset Controller SDRA0RES • reset upon wake-up from DEEPSTOP mode

CAUTION

By default the SDRA0RES reset is active.
Thus before accessing this module this reset must be released via the MRSTC register of the Reset Controller. Refer to Section 9, Reset Controller for details.

15.1.6 I/O signals

The following table shows the I/O signals of the SDR-SDRAM Memory Controller.

Table 15.6 SDRAn I/O signals

SDRAn signals	Function	Connected to
SDRA0		
SDRAA[12:11], SDRAA10, SDRAA[9:0]	Address output signals	Port SDRAA[12:11], SDRAA10, SDRAA[9:0]
SDRADQ[31:0]	Data input/output signals	Port SDRADQ[31:0]
SDRABA[1:0]	Bank address output signals	Port SDRABA[1:0]
SDRACK	Memory clock signal output	Port SDRACK
SDRACKE	Clock enable output signal	Port SDRACKE
SDRADM[3:0]	Data mask input/output signals	Port SDRADM[3:0]
SDRACSB	Chip select output signal	Port SDRACSB
SDRACASB	CAS strobe output signal	Port SDRACASB
SDRARASB	RAS strobe output signal	Port SDRARASB
SDRAWEB	Write enable output signal	Port SDRAWEB

Note 1. The suffix “B” in the signal names in the above table denotes low level active signals.

15.1.7 Partial Operation of I/O Signals

Table 15.7 D1M1H, D1M1A unused SDRAM interface handling

External SDRAM connection	Handling of unused data signals	Handling of unused address signals
32 bit	• DQ[31:0]: all used • DM[3:0]: all used	Leave open
16 bit	• DQ[31:16]: leave open • DM[3:2]: leave open	
None	Refer to Section 2.8, Recommend connection of unused pins.	

15.2 Overview

The SDR-SDRAM Memory Controller gives access to external single data rate type SDRAM memory devices.

Main features

- Data bus width and maximum bandwidth:
  - 32-bit
  - max. bandwidth
    - D1M1H: 400 MB/s
    - D1M1A: 480 MB/s
- up to 128 MB external SDR-SDRAM
- Multi bank operation
  - 4 banks
  - can be interleaved between RAS and CAS on several positions
- Burst length : fix to 4
- Timing and address size freely programmable
- Early precharge in case of page miss (different bank)
- Refresh
  - Auto refresh, with early refresh on idle cycles
  - Self refresh
  - average interval and the maximum post count selectable
- Power down mode

15.3 SDR-SDRAM Memory Controller Registers

This section contains a description of all registers of the SDR-SDRAM Memory Controller.

Table 15.8 SDR-SDRAM Memory Controller registers overview

Register name	Shortcut	Address
Error status register	DBSVCR	<SDRAn_base> + 004_H
SDRAM type selection register	DBKIND	<SDRAn_base> + 008_H
SDRAM enable register	DBEN	<SDRAn_base> + 010_H
SDRAM command control register	DBCMDCNT	<SDRAn_base> + 014_H
SDRAM configuration register	DBCONF	<SDRAn_base> + 020_H
SDRAM device control register	DBDMOV	<SDRAn_base> + 024_H
SDRAM timing register 0	DBTR0	<SDRAn_base> + 030_H
SDRAM timing register 1	DBTR1	<SDRAn_base> + 034_H
SDRAM timing register 2	DBTR2	<SDRAn_base> + 038_H
SDRAM refresh/power-down control register 0	DBRFPDN0	<SDRAn_base> + 040_H
SDRAM refresh/power-down control register 1	DBRFPDN1	<SDRAn_base> + 044_H
SDRAM refresh/power-down control register 2	DBRFPDN2	<SDRAn_base> + 048_H
SDRAM refresh status register	DBRFSTS	<SDRAn_base> + $04C_H$
SDRAM mode setting register	DBMRCNT	<SDRAn_base> + 060_H
PHY control register 0	DBPDCNT0	<SDRAn_base> + 108_H

<SDRAn_base>

The base addresses <SDRAn_base> of the SDRAn is defined in the first section of this chapter under the key word “Register addresses”.

15.3.1 SDR-SDRAM Memory Controller control registers details

15.3.1.1 DBSVCR – Error status register

This register holds various error indication flags.

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 004_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	SRF PDN	INAC TIVE	0	0	BAD_ OPC	0	0	BAD_ ADDR	ERR_ SNT	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R	R	R/W	R/W	R

Table 15.9 DBSVCR register contents (1/2)

Bit position	Bit name	Function
31 to 10	Reserved	Reading returns the initial value. When written, write the initial value.
9	SRFPDN	Error cause: Data access to SDRAM during SDRAM self-refresh or power-down At read: 0: no error 1: SDRAM data was accessed while the SDRAM is in self-refresh or power-down mode, i.e. while DBRFPDN0.SRFEN = 1 or DBRFPDN0.PDN = 1. At write access: write 0: clear SRFPDN to 0 write 1: no function
8	INACTIVE	Error cause: Data access to SDRAM during SDRAM disable At read: 0: no error 1: SDRAM data was accessed while the SDRAM access is disabled, i.e. while DBEN.ACEN = 0. At write access: write 0: clear INACTIVE to 0 write 1: no function
7 to 6	Reserved	Reading returns the initial value. When written, write the initial value.
5	BAD_OPC	Error cause: Not supported transaction of the XC cross-connect At read: 0: no error 1: unsupported transaction of the XC cross-connect*1 At write access: write 0: clear BAD_OPC to 0 write 1: no function
4 to 3	Reserved	Reading returns the initial value. When written, write the initial value.
2	BAD_ADDR	Error cause: Access to undefined register of SDRA At read: 0: no error 1: access to undefined register of SDRA has occurred*2 At write access: write 0: clear BAD_ADDR to 0 write 1: no function

Table 15.9 DBSVCR register contents (2/2)

Bit position	Bit name	Function
1	ERR_SNT	Error flag: SDRA has issued an error to the cross-connect At read: 0: no error 1: SDRA error was returned to the cross-connect The error cause can be analyzed by the SRFPDN, INACTIVE and BAD_OPC error flags. At write access: write 0: clear ERR_SNT to 0 write 1: no function
0	Reserved	Reading returns the initial value. When written, write the initial value.

Note 1. The cause for unsupported transactions on XC cross-connect are unsupported data block width (bigger than 128 bit) or burst length errors (burst access to fixed address or unaligned bursts).
SDRA can detect those illegal transactions, even though they can not occur by normal cross-connect operation.

Note 2. Error cause of “undefined register access” means XC cross-connect access to register addresses of SDRA, where no SDRA register is present.

The error causes indicated by the DBSVCR register (SRFPDN, INACTIVE, BAD_OPC, BAD_ADDR and ERR_SNT) are signalled to the CPU directly by a SYSERR exception.

15.3.1.2 DBKIND – SDRAM type selection register

This register defines the SDRAM type.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBEN.ACEN = 0).
Thus write only to the DBKIND register during "Initial Sequence - 1st setting of SDRA" in Section 15.4.2, Initialization sequence.

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 008_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	DDCG[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 15.10 DBKIND register contents

Bit position	Bit name	Function
31 to 3	Reserved	Reading returns the initial value. When written, write the initial value.
2 to 0	DDCG[2:0]	SDRAM type selection 001_B: SDR-SDRAM others: setting prohibited

15.3.1.3 DBEN – SDRAM enable register

This register is used to enable or disable the SDRAM access.

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 010_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ACEN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15.11 DBEN register contents

Bit position	Bit name	Function
31 to 1	Reserved	Reading returns the initial value. When written, write the initial value.
0	ACEN	SDRAM access enable/disable 0: SDRAM access disabled 1: SDRAM access enabled

NOTE

If the SDRAM is accessed while DBEN.ACEN = 0, the error INACTIVE flag in the error status register DBSVCR is set.

15.3.1.4 DBCMDCNT – SDRAM command control register

This register is used to send commands to the SDRAM devices.

CAUTION

Writing to this register is only permitted, when auto-refresh is not performed (DBRFPDN0.ARFEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 014_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	CMD[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 15.12 DBCMDCNT register contents

Bit position	Bit name	Function
31 to 3	Reserved	Reading returns the initial value. When written, write the initial value.
2 to 0	CMD[2:0]	SDRAM command 000_B: Normal operation 010_B: Precharge all (PALL) command is issued. 011_B: SDRACKE signal goes to high level. 100_B: refresh command (REF) All others: Setting prohibited. These bits are always read as 0. Writing CMD[2:0] executes an initialization sequence and self-refresh shift/release of the SDRAM. Each write executes the respective command once. For instance, when the refresh command shall be executed twice, it is necessary to write CMD[2:0] = 100_B two times. The value set by the SDRAM timing registers is used at precharge period and minimum intervals etc. between refreshing and the next command.

15.3.1.5 DBCONF – SDRAM configuration register

This register is used to specify the SDRAM configuration.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBEN.ACEN = 0) and auto-refresh is not performed (DBRFPDN0.ARFEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 020_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	SPLIT[8:0]								
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BKADM[1:0]		BKADP[5:0]						BKADB[5:0]						BWIDTH[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.13 DBCONF register contents

Bit position	Bit name	Function
31 to 25	Reserved	Reading returns the initial value. When written, write the initial value.
24 to 16	SPLIT[8:0]	Memory configuration Refer to Section 15.4.1, Memory configuration by DBCONF.SPLIT[8:0].
15 to 14	BKADM[1:0]	Bank address definition selection*1 00_B: Consecutive addresses (BA0 and BA1 are specified by BKADP[5:0]) 01_B: Non-consecutive addresses (BA0 is specified by BKADP[5:0], BA1 is specified with BKADB[5:0]) Others: setting prohibited
13 to 8	BKADP[5:0]	Bank address position*1 00 0000_B: Top of the column address 00 1010_B: 1 KB (BA0 set to address 10) 00 1011_B: 2 KB (BA0 set to address 11) 00 1100_B: 4 KB (BA0 set to address 12) Others: setting prohibited
7 to 2	BKADB[5:0]	Upper bank address position*1 00 0000_B: setting prohibited 00 1101_B: BA1 set to address 13 00 1110_B: BA1 set to address 14 00 1111_B: BA1 set to address 15 01 0000_B: BA1 set to address 16 Others: setting prohibited BKADB[5:0] is only effective if BKADM[1:0] = 01_B.
1 to 0	BWIDTH[1:0]	External data bus width 01_B: 16 bit 10_B: 32 bit All others: Setting prohibited

Note 1. Refer to Section 15.6.2, Positioning of bank addresses for details.

15.3.1.6 DBDMOV – SDRAM device control register

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBEN.ACEN = 0) and auto-refresh is not performed (DBRFPDN0.ARFEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 024_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	TRCDC[1:0]		0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R

Table 15.14 DBDMOV register contents

Bit position	Bit name	Function
31 to 4	Reserved	Reading returns the initial value. When written, write the initial value.
3 to 2	TRCDC[1:0]	TRCD bit modification for ACT-WRITE timing 00_B: less than 1 cycle from TRCD setting for ACT-WRITE timing Others: setting prohibited This register can modify the DBTR0.TRCD[2:0] setting for ACT-WRITE timing. ACT-READ timing is using DBTR0.TRCD[2:0] setting and does thus not depend on TRCDC[2:0].
1 to 0	Reserved	Reading returns the initial value. When written, write the initial value.

15.3.1.7 DBTR0 – SDRAM timing register 0

This register is used to define SDRAM timing parameters.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBEN.ACEN = 0) and auto-refresh is not performed (DBRFPDN0.ARFEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 030_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	CL[2:0]			0	0	0	TRAS[4:0]				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	TRFC[6:0]							0	0	0	0	0	TRCD[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 15.15 DBTR0 register contents (1/2)

Bit position	Bit name	Function
31 to 27	Reserved	Reading returns the initial value. When written, write the initial value.
26 to 24	CL[2:0]	CAS latency 010_B: 2 cycles 011_B: 3 cycles Others: setting prohibited **Note:** The CL[2:0] value must be set to the same value as CAS latency, sent to the SDRAM's Mode Set Register (MSR) via the DBMRCNT register.
23 to 21	Reserved	Reading returns the initial value. When written, write the initial value.
20 to 16	TRAS[4:0]	ACT-PRE interval $0\ 0000_B$: 1 cycle $0\ 0001_B$: 2 cycles $0\ 0010_B$: 3 cycles ... $1\ 0001_B$: 18 cycles Others: setting prohibited
15	Reserved	Reading returns the initial value. When written, write the initial value.
14 to 8	TRFC[6:0]	REF-ACT/REF interval $000\ 0001_B$: 2 cycles $000\ 0010_B$: 3 cycles ... $110\ 1100_B$: 109 cycles Others: setting prohibited
7 to 3	Reserved	Reading returns the initial value. When written, write the initial value.

Table 15.15 DBTR0 register contents (2/2)

Bit position	Bit name	Function
2 to 0	TRCD[2:0]	ACT-READ interval setting bits 000_B: 1 cycle 001_B: 2 cycles 010_B: 3 cycles 011_B: 4 cycles 100_B: 5 cycles 101_B: 6 cycles Others: setting prohibited

15.3.1.8 DBTR1 – SDRAM timing register 1

This register is used to define SDRAM timing parameters.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBEN.ACEN = 0) and auto-refresh is not performed (DBRFPDN0.ARFEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 034_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	CKEH[2:0]			0	0	0	0	0	TRP[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	TRRD[2:0]			0	0	0	0	0	TWR[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 15.16 DBTR1 register contents

Bit position	Bit name	Function
31 to 27	Reserved	Reading returns the initial value. When written, write the initial value.
26 to 24	CKEH[2:0]	CKEH period 000_B: 1 cycle 001_B: 2 cycles ... 111_B: 8 cycles
23 to 19	Reserved	Reading returns the initial value. When written, write the initial value.
18 to 16	TRP[2:0]	PRE period 000_B: 1 cycle 001_B: 2 cycles ... 101_B: 6 cycles Others: setting prohibited
15 to 11	Reserved	Reading returns the initial value. When written, write the initial value.
10 to 8	TRRD[2:0]	ACT(A)-ACT(B) period 000_B: 1 cycle 001_B: 2 cycles 010_B: 3 cycles 011_B: 4 cycles Others: setting prohibited
7 to 3	Reserved	Reading returns the initial value. When written, write the initial value.
2 to 0	TWR[2:0]	Write recovery period 001_B: 1 cycle 010_B: 2 cycles ... 101_B: 5 cycles Others: setting prohibited

15.3.1.9 DBTR2 – SDRAM timing register 2

This register is used to define SDRAM timing parameters.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBEN.ACEN = 0) and auto-refresh is not performed (DBRFPDN0.ARFEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 038_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	TRTP[1:0]		0	0	0	TRC[4:0]				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	RDWR[3:0]				0	0	0	0	WRRD[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 15.17 DBTR2 register contents

Bit position	Bit name	Function
31 to 26	Reserved	Reading returns the initial value. When written, write the initial value.
25 to 24	TRTP[1:0]	READ-PRE period 01_B: 2 cycles 10_B: 3 cycles Others: setting prohibited
23 to 21	Reserved	Reading returns the initial value. When written, write the initial value.
20 to 16	TRC[4:0]	ACT-ACT/REF period $0\ 0000_B$: 1 cycle $0\ 0001_B$: 2 cycles ... $1\ 0110_B$: 23 cycles Others: setting prohibited
15 to 12	Reserved	Reading returns the initial value. When written, write the initial value.
11 to 8	RDWR[3:0]	READ-WRITE period 0011_B: 4 cycles 0100_B: 5 cycles ... 1000_B: 9 cycles Others: setting prohibited
7 to 4	Reserved	Reading returns the initial value. When written, write the initial value.
3 to 0	WRRD[3:0]	WRITE-READ period 0001_B: 2 cycles 0010_B: 3 cycles ... 1100_B: 13 cycles Others: setting prohibited

15.3.1.10 DBRFPDN0 – SDRAM refresh/power-down control register 0

This register is used to control refresh and power-down modes.

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 040_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ARFEN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	PDN	0	SRFEN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Table 15.18 DBRFPDN0 register contents

Bit position	Bit name	Function
31 to 17	Reserved	Reading returns the initial value. When written, write the initial value.
16	ARFEN	Auto-refresh control 0: auto-refresh disabled 1: auto-refresh enabled
15 to 3	Reserved	Reading returns the initial value. When written, write the initial value.
2	PDN	Power-down control 0: release power-down 1: enter power-down
1	Reserved	Reading returns the initial value. When written, write the initial value.
0	SRFEN	Self-refresh control 0: self-refresh disabled 1: self-refresh enabled

NOTE

If the SDRAM is accessed in self-refresh (SRFEN = 1) or power-down (PDN = 1) mode, the error SRFPDN flag in the error status register DBSVCR is set.

15.3.1.11 DBRFPDN1 – SDRAM refresh/power-down control register 1

This register is used to define the average refresh interval.

CAUTION

Writing to this register is only permitted, when auto-refresh is not performed (DBRFPDN0.ARFEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 044_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	TREFI[12:0]												
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.19 DBRFPDN1 register contents

Bit position	Bit name	Function
31 to 13	Reserved	Reading returns the initial value. When written, write the initial value.
12 to 0	TREFI[12:0]	Average refresh interval 0040_H: 65 cycles 0041_H: 66 cycles ... $1FFF_H$: 8192 cycles Others: setting prohibited

15.3.1.12 DBRFPDN2 – SDRAM refresh/power-down control register 2

This register is used to define the time for the refresh execution.

CAUTION

Writing to this register is only permitted, when auto-refresh is not performed (DBRFPDN0.ARFEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 048_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	LV1TH[14:0]														
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	LV0TH[9:0]									
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.20 DBRFPDN2 register contents

Bit position	Bit name	Function
31	Reserved	Reading returns the initial value. When written, write the initial value.
30 to 16	LV1TH[14:0]	Level 1 threshold Number of SDRAM clock cycles during idle bus, before executing refresh.
15 to 10	Reserved	Reading returns the initial value. When written, write the initial value.
9 to 0	LV0TH[9:0]	Level 0 threshold Number of SDRAM clock cycles during busy bus, before pausing bus transactions and executing refresh.

NOTES

1. Set the LV1TH[14:0] value within the range of the ACT-PRE command interval described in SDRAM devices' data sheet.
2. LV1TH[14:0] should be set to $t_{REFI} \times 7$. Due to SDRAM timing constraints concerning the 9th REF command timing LV1TH[14:0] must fulfil $t_{REFI} < LV1TH[14:0] \leq t_{REFI} \times 9$.
3. LV0TH[9:0] register should be set to the WRITE-WRITE command timing $(t_{RP} + t_{RCD} + t_{WR} + 3) \times 5$ in order to prevent an underflow of refresh cycle counter.

15.3.1.13 DBRFSTS – SDRAM refresh status register

This register informs about an underflow of the refresh counter.

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + $04C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RFUDF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15.21 DBRFSTS register contents

Bit position	Bit name	Function
31 to 1	Reserved	Reading returns the initial value. When written, write the initial value.
0	RFUDF	Refresh counter underflow flag 0: refresh counter did not underflow 1: refresh counter underflow occurred The RFUDF flag can be cleared by writing 0.

RFUDF is set when the refresh counter changes from 1 to 0.

That means refresh could not be executed, because the LV0TH[9:0] value is smaller than the maximum number of command execution cycles. In that case, change the LV0TH[9:0] value.

Refresh counter underflow asserts also the refresh counter underflow interrupt INTSDRA.
For de-assertion of the INTSDRA interrupt the RFUDF flag must be cleared by writing RFUDF = 0.

Please refer to Section 15.4.4, Auto-refresh for the refreshing counter.

15.3.1.14 DBMRCNT – SDRAM mode setting register

This register is used to issue “Mode Register Set (MRS)” and “Extended Mode Register Set (EMRS)” commands towards the SDRAM devices.

A write to the DBMRCNT register initiates a MRS/EMRS command to the SDRAM, which transfers several parameters to the SDRAM during the initialization process.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBEN.ACEN = 0) and auto-refresh is not performed (DBRFPDN0.ARFEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 060_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BA[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	MA[12:0]												
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.22 DBMRCNT register contents

Bit position	Bit name	Function
31 to 18	Reserved	Reading returns the initial value. When written, write the initial value.
17 to 16	BA[1:0]	BA[1:0] defines the level of the bank address during the MRS/EMRS command (selection of MRS or EMRS SDRAM register) BA[1]: SDRABA[1] BA[0]: SDRABA[0] Reading these bits always returns 0.
15 to 13	Reserved	Reading returns the initial value. When written, write the initial value.
12 to 0	MA[12:0]	MA[12:0] defines the level of the address lines during the MRS/EMRS command output MA[12:11]: SDRAA[12:11] MA[10]: SDRAA10 MA[9:0]: SDRAA[9:0] Reading these bits always returns 0.

NOTE

The CAS latency, transferred to the SDRAM, must be set to the same value, as defined in DBTR0.CL[2:0].

15.3.1.15 DBPDCNT0 – PHY control register 0

This register is used to define the write recovery period.

Access: This register can be accessed in 32-bit units.

Address: <SDRAn_base> + 108_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OFFSET[2:0]			0	ENOFFSET[3:0]				0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	STBYN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 15.23 DBPDCNT0 register contents

Bit position	Bit name	Function
31 to 29	OFFSET[2:0]	OFFSET[2:0] controls the read data capture timing of SDRAM. 001_B: for SDR-SDRAM other settings: prohibited
28 to 28	Reserved	Reading returns the initial value. When written, write the initial value.
27 to 24	ENOFFSET[3:0]	ENOFFSET[3:0] controls the I/O buffer input enable timing of SDRAM. 0011_B: for SDR-SDRAM other settings: prohibited
23 to 1	Reserved	Reading returns the initial value. When written, write the initial value.
0	STBYN	Clock standby control signal. 0: enter standby 1: exit standby

15.4 SDR-SDRAM Operation

15.4.1 Memory configuration by DBCONF.SPLIT[8:0]

Support memory devices

Following memory devices are supported:

- 16-bit data bus, that connects one 16-bit SDRAM
- 32-bit data bus, that connects one 32-bit SDRAM or two 16-bit SDRAMs in parallel

Memory configuration selection

The memory configuration is set up via the DBCONF.SPLIT[8:0] settings.

SPLIT[8:0] is a composition of different fields in form of xx_y_ppp_qqq. The meaning and the calculation formula of each numerical value are shown in the following.

- “x” specifies the width of the memory:
  - xx = 01_B: 16 bit memory width
  - xx= 10_B: 32 bit memory width
- “y” specifies the number of banks:
  - y = 0: 4 banks
- “p” specifies the width of the row address:
  - row address width = 1010_B + ppp_B
- “q" specifies the width of the column address:
  - column address width = 0111_B + $0qqq_B$

The following table shows settings for configuration with

- 16-bit (xx = 01_B) and 32-bit (xx = 10_B) SDRAM devices
- 4 banks (y = 0)
- different number of rows and columns

Table 15.24 Memory configuration settings

DBCONF.SPLIT[8:0] (xx_y_ppp_qqq)	Memory configuration	External bus width	Bank addresses	Row addresses	Column addresses
10_0_010_001	4M×32bit = 128Mbit	32-bit	2	12	8
10_0_010_010	8M×32bit = 256Mbit				9
10_0_010_011	16M×32bit = 512Mbit				10
10_0_011_001	8M×32bit = 256Mbit			13	8
10_0_011_010	16M×32bit = 512Mbit				9
10_0_011_011	32M×32bit = 1Gbit				10
01_0_010_001	4M×16bit = 64Mbit	16-bit	2	12	8
01_0_010_010	8M×16bit = 128Mbit				9
01_0_010_011	16M×16bit = 256Mbit				10
01_0_011_001	8M×16bit = 128Mbit			13	8
01_0_011_010	16M×16bit = 256Mbit				9
01_0_011_011	32M×16bit = 512Mbit				10

15.4.2 Initialization sequence

The following sequence enables SDRAM access after power-up.

NOTES

1. The sequences below are examples and may need to be changed according to the SDRAM device properties.
   Consult the SDRAM device vendor's data sheet for details.
2. Follow the values described in SDRAM device vendor's data sheet for correct waiting time between the steps in the sequence.

1. Wait until power supply for SDRA is stable.
2. Set DBPDCNT0.STBYN = 1 to release the clock supply of PHY.
3. Set DBCMDCNT.CMD[2:0] = 011_B to set CKE to high level.
4. Wait 200 µsec.
5. Set the following registers:
   – DBCONF
   – DBTR0
   – DBTR1
   – DBTR2
6. Set DBKIND.DDCG[2:0] = 001_B for SDR-SDRAM memory type
7. Set DBPDCNT0.OFFSET[2:0] = 001_B and DBPDCNT0.ENOFFSET[3:0] = 0011_B for SDR-SDRAM
8. Send the PALL (precharge all) command to the SDRAM by DBCMDCNT.CMD[2:0] = 010_B.
9. Send two times the REF (refresh) command to the SDRAM by writing two times DBCMDCNT.CMD[2:0] = 100_B.
10. Issue the MRS command towards the SDRAM by writing to the SDRAM mode setting register (DBMRCNT) in order to transfer various parameters to the SDRAM devices, in particular:
    – Set the burst length to 4.
    – The CAS latency must be set to the DBTR0.CL[2:0] value.
11. Enable SDRAM access by DBEN.ACEN = 1.
12. Set the refresh timing via the DBRFPDN1 and DBRFPDN2 registers.
13. Enable auto-refresh by DBRFPDN0.ARFEN = 1.

The SDRAM can be accessed.

15.4.3 Self-refresh

Changing to the self-refresh mode reduces power consumption, when the SDRAM is not accessed. The content of the SDRAM is retained.

NOTE

The sequences below are examples and may need to be changed according to the SDRAM device properties.
Consult the SDRAM device vendor's data sheet for details.

Self-refresh entry sequence

Follow the procedure below for entering the self-refresh mode.

(1) Stop all SDRAM accesses.

(2) Confirm that all SDRAM read transactions are completed by performing a dummy read from DBKIND register.

(3) Disable SDRAM access by DBEN.ACEN = 0.

(4) Disable auto-refresh by DBRFPDN0.ARFEN = 0.

(5) Send the PALL (precharge all) command to the SDRAM by DBCMDCNT.CMD[2:0] = 010_B.

(6) Send the REF (refresh) command to the SDRAM by DBCMDCNT.CMD[2:0] = 100_B.

(7) Enable self-refresh by DBRFPDN0.SRFEN = 1.

Now the clock can be stopped.

NOTE

If the SDRAM is accessed in self-refresh mode, the error SRFPDN flag in the error status register DBSVCR is set.

Self-refresh termination sequence

Follow the procedure below for terminating the self-refresh mode.

(1) Disable self-refresh by DBRFPDN0.SRFEN = 0.

(2) Enable SDRAM access by DBEN.ACEN = 1.

(3) Enable auto-refresh by DBRFPDN0.ARFEN = 1.

The SDRAM can be accessed.

NOTE

If the SDRAM is accessed in self-refresh mode, the error SRFPDN flag in the error status register DBSVCR is set.

15.4.4 Auto-refresh

Auto-refresh is enabled by DBRFPDN0.ARFEN = 1.
During auto-refresh the SDRAM can be accessed.

Refresh levels

In order to minimize the loss of data read/write bandwidth due to refresh cycles, the urgency for latching an auto-refresh cycles is set in different priority levels.

The auto-refresh level is determined by the current value of a 14-bit refresh counter, that counts down with the SDRAM clock. That means basically: the lower the refresh counter value, the higher the priority to perform a refresh.

The refresh levels are configurable by three register values:

- Level 0 threshold DBRFDPN2.LV0TH[9:0]
- Level 1 threshold DBRFDPN2.LV1TH[14:0]
- Average refresh interval time DBRFPDN1.TREFI[12:0]

Three refresh levels are defined:

- Level 0 - high priority refresh: 0 < refresh counter value ≤ LV0TH[9:0]
  Auto-refresh is performed during read/write transactions.
  Since the bus remains busy during continuous read/write transactions, the transactions must be paused for inserting an auto-refresh cycle. Consequently level 0 refreshing yields a loss of data transaction bandwidth.
- Level 1 - low priority refresh: LV0TH[9:0] < refresh counter value ≤ LV1TH[14:0]
  Auto-refresh is performed when the SDRAM bus is idle, i.e. no read/write transaction is ongoing.
- Level 2 - no refresh necessary: LV1TH[14:0] < refresh counter value ≤ t_{REFI} + LV1TH[14:0]
  Auto-refresh is not performed.

Refresh counter reload

After enabling auto-refresh (DBRFPDN0.ARFEN = 1), an auto-refresh cycle is immediately performed and the refresh counter is loaded with the average refresh interval DBRFPDN1.TREFI[12:0].

Each further refresh cycle adds TREFI[12:0] to the current refresh counter value, and shifts the next refresh towards lower priority.

Refresh counter underflow

If the counter underflows, i.e. no refresh has been performed until DBRFPDN1.TREFI[12:0] has count down to 0, the refresh counter underflow interrupt INTSDRA is asserted and the underflow flag DBRFSTS.RFUDF is set.

NOTE

The INTSDRA interrupt remains asserted until the underflow flag is cleared by DBRFSTS.RFUDF = 0.

After executing the precharge of all banks by using the PALL (precharge all) command, refreshing executes the auto refreshing command. Therefore, the data access to all banks enters the state of page miss after it refreshes it.

NOTE

When calculating the proper value for the refresh level 0 (DBRFDPN2.LV0TH[9:0]) take into account that launching refresh includes a 16 bytes transfer to the SDRAM and thus the actual refresh start is delayed by this.

The figure below illustrates the auto-refresh processing.

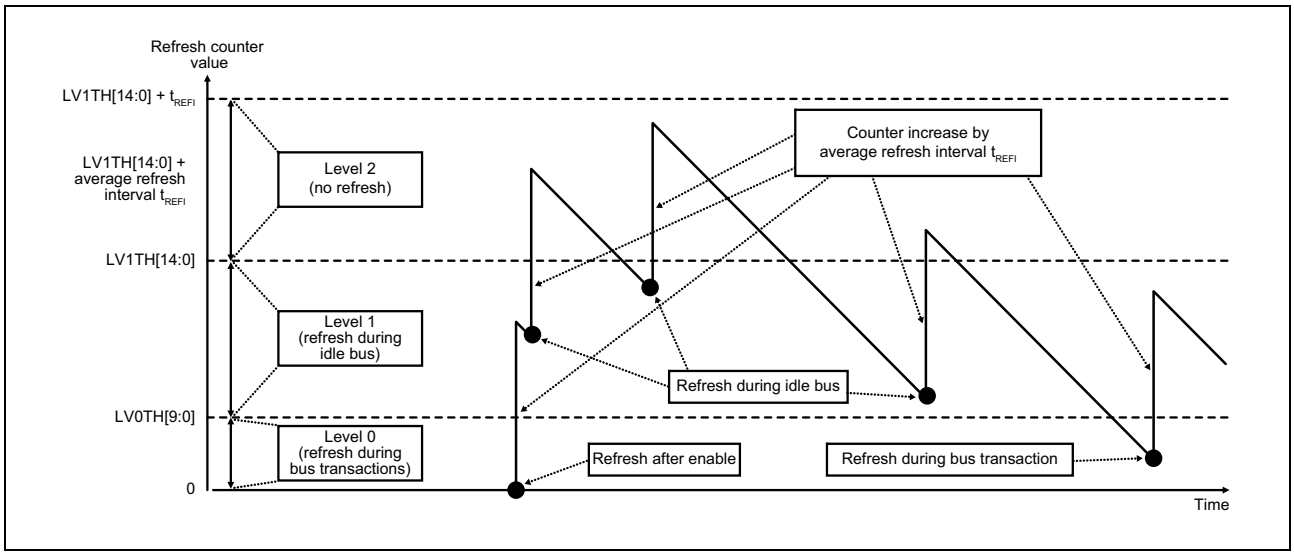


Figure 15.1 Auto-refresh processing

15.4.5 Power-down mode

Power consumption can be reduced when SDRAM access is not necessary by entering power-down mode.
In this mode the SDRAM internal clock is stopped and the SDRAM content is retained by the SDRA auto-refresh.
Power-down mode allows a fast return to normal operation, when the SDRAM must be accessed.

NOTES

1. The sequences below are examples and may need to be changed according to the SDRAM device properties.
   Consult the SDRAM device vendor's data sheet for details.
2. The here mentioned SDRAM power-down mode must not be mixed up with the microcontroller's DEEPSTOP mode.

Power-down entry sequence

Follow the procedure below for entering power-down mode.

(1) Stop all SDRAM accesses.

(2) Confirm that all SDRAM read transactions are completed by performing a dummy read from DBKIND register.

(3) Disable SDRAM access by DBEN.ACEN = 0.

(4) Send the PALL (precharge all) command to the SDRAM by DBCMDCNT.CMD[2:0] = 010_B.

(5) Enable power-down by DBRFPDN0.PDN = 1.

NOTE

If the SDRAM is accessed in power-down mode, the error SRFPDN flag in the error status register DBSVCR is set.

Power-up termination sequence

Follow the procedure below for terminating the power-down mode.

(1) Disable power-down by DBRFPDN0.PDN = 0.

(2) Enable SDRAM access by DBEN.ACEN = 1.

The SDRAM can be accessed.

In order to hold the SDRAM data in the SDRAM power-down mode, it is necessary to issue a regular refresh command as during normal operation.

Any attempt to access any SDRAM data during power-down mode results in an error and the DBSVCR.INACTIVE error flag is set.

15.5 SDR-SDRAM Transaction Restrictor

The SDR-SDRAM transaction restrictor can be used to serve cross-connect masters read request with higher priority than write request or vice versa.

It allows to contiguously process a programmable number of read (or write) transactions, while suspending any write (or read, respectively) request.

Transaction restrictor features overview

- The transaction restrictor can control the ratio between read and write transactions
- Three transaction restrictor modes:
  - write transaction restriction mode: gives priority to read requests
  - read transaction restriction mode: gives priority to write requests
  - normal mode: read and write requests are handled with same priority

This function is the same as VRAM transaction restrictor.

For the detail function, refer to Section 54.3, VRAM Transaction Restrictor.

15.5.1 SDR-SDRAM Transaction Restrictor Registers

The SDR-SDRAM transaction restrictor is controlled and operated by means of the following registers.

Table 15.25 SDR-SDRAM transaction restrictor registers overview

Address	Symbol	Register Name	R/W	Value after reset	Access Size
FFFD 3000_H	SDRATRCTL	SDRA transaction restrictor control register	R/W	0000 0000_H	32
FFFD 3004_H	SDRATRINTVL	SDRA transaction restrictor interval time register	R/W	0000 0000_H	32

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> is defined as “SDRAn” for the registers in the table above.

<Symbol> is defined in the above table.

15.5.1.1 SDRATRCTL — SDRA transaction restrictor control register (D1M1H, D1M1A only)

These registers control the transaction restrictor operation mode for SDRA.

CAUTION

This register must not be changed while SDR-SDRAM memory access is ongoing.

Access: This register can be read/written in 32-bit units.

Address: FFFD 3000_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IDLTIM[7:0]								—	—	—	—	—	—	RCTL[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTOVFL[6:0]							—	WTOVFL[6:0]							—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 15.26 SDRATRCTL register contents

Bit Position	Bit Name	Function
31 to 24	IDLTIM[7:0]	IDLTIM[7:0] defines the idle time in number of C_ISO_XCCLK clock cycles. The idle time is the time to change from write suspend (WSUSP) or read suspend (RSUSP) state to initial state (INIT).
23 to 18	Reserved	When read, the value after reset is returned. When written, write the value after reset.
17 to 16	RCTL[1:0]	Transaction restrictor mode 00_B: normal mode (transaction restrictor disabled) 01_B: write transaction restrictor mode 10_B: read transaction restrictor mode 11_B: setting prohibited
15 to 9	RTOVFL[6:0]	Read overflow count RTOVFL[6:0] determines the number of written data in read suspend state (RSUSP) before changing to read interval state (RINTVL). The actual number of written data is RTOVFL[6:0] × 2. RTOVFL[6:0] is only valid in read transaction restrictor mode, i.e. RTCL[1:0] = 10_B.
8	Reserved	When read, the value after reset is returned. When written, write the value after reset.
7 to 1	WTOVFL[6:0]	Write overflow count WTOVFL[6:0] determines the number of read data in write suspend state (WSUSP) before changing to write interval state (WINTVL). The actual number of read data is WTOVFL[6:0] × 2. WTOVFL[6:0] is only valid in write transaction restrictor mode, i.e. RTCL[1:0] = 01_B.
0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

15.5.1.2 SDRATRINTVL — SDRA transaction restrictor interval time register (D1M1H, D1M1A only)

These registers define the state change timing of from interval states to suspend states.

CAUTION

This register must not be changed while SDR-SDRAM memory access is ongoing.

Access: This register can be read/written in 32-bit units.

Address: FFFD 3004_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ITVLTIM[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 15.27 SDRATRINTVL register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned. When written, write the value after reset.
7 to 0	ITVLTIM[7:0]	ITVLTIM[7:0] defines the interval time in number of C_ISO_XCCLK clock cycles. The interval time is the time to change from write or read interval state (WINTVL or RINTVL) to write or read suspend state (WSUSP or RSUSP). ITVLTIM[7:0] is only valid in write or read transaction restrictor mode, i.e. RTCL[1:0] = 01_B or 10_B.

15.6 Appendix

In the description below

- SDRABA[1:0] is represented by BA[1:0]
- SDRAA[12:0] is represented by MA[12:0].

15.6.1 Relation between external and logical address

Table 15.28 1 x 16-bit SDRAM (DBCONF.BWIDTH[1:0] = 01_B)

Type		BA1	BA0	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
4M×16b	ROW	A10	A9	—	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
	COL	A10	A9	—	—	AP	—	—	A8	A7	A6	A5	A4	A3	A2	A1
8M×16b	ROW	A10	A9	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11
	COL	A10	A9	—	—	AP	—	—	A8	A7	A6	A5	A4	A3	A2	A1
8M×16b	ROW	A11	A10	—	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
	COL	A11	A10	—	—	AP	—	A9	A8	A7	A6	A5	A4	A3	A2	A1
16M×16b	ROW	A11	A10	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
	COL	A11	A10	—	—	AP	—	A9	A8	A7	A6	A5	A4	A3	A2	A1
16M×16b	ROW	A12	A11	—	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	A12	A11	—	—	AP	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
32M×16b	ROW	A12	A11	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	A12	A11	—	—	AP	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1

Table 15.29 1 x 32-bit or 2 x 16-bit SDRAM (DBCONF.BWIDTH[1:0] = 10_B)

Type		BA1	BA0	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
4M×32b	ROW	A11	A10	—	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
	COL	A11	A10	—	—	AP	—	—	A9	A8	A7	A6	A5	A4	A3	A2
8M×32b	ROW	A11	A10	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
	COL	A11	A10	—	—	AP	—	—	A9	A8	A7	A6	A5	A4	A3	A2
8M×32b	ROW	A12	A11	—	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	A12	A11	—	—	AP	—	A10	A9	A8	A7	A6	A5	A4	A3	A2
16M×32b	ROW	A12	A11	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	A12	A11	—	—	AP	—	A10	A9	A8	A7	A6	A5	A4	A3	A2
16M×32b	ROW	A13	A12	—	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	A13	A12	—	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2
32M×32b	ROW	A13	A12	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	A13	A12	—	—	AP	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

15.6.2 Positioning of bank addresses

The position of the bank address in the logical address can be fixed without depending on the data bus width and the number of columns of SDRAM by setting the SDRAM configuration register DBCONF. Moreover, it is possible to set each bit of the bank address to non-continuous position.

Table 15.30 32-bit interface

DBCONF register settings			Logical address lines A[26:0]*1		
BKADM[1:0]	**BKADP[5:0]**	**BKADB[5:0]**	**BA[1:0]**	**Row address**	**Column address**
00_B	$00\ 0000_B$	xx $xxxx_B$	A[13:12]	A[26:14]	A[11:2]
	$00\ 1010_B$		A[11:10]	A[26:14]	A[13:12], A[9:2]
	$00\ 1011_B$		A[12:11]	A[26:14]	A[13], A[10:2]
01_B	$00\ 1010_B$	$00\ 1101_B$	A[13], A[10]	A[26:14]	A[12:11], A[9:2]
	$00\ 1011_B$	$00\ 1101_B$	A[13], A[11]	A[26:14]	A[12], A[10:2]
	$00\ 1010_B$	$00\ 1111_B$	A[15], A[10]	A[26:16], A[14:13]	A[12:11], A[9:2]

Note 1. A[1:0] is not used because of 32-bit alignment.

Table 15.31 16-bit interface

DBCONF register settings			Logical address lines A[26:0]*1		
BKADM[1:0]	**BKADP[5:0]**	**BKADB[5:0]**	**BA[1:0]**	**Row address**	**Column address**
00_B	$00\ 0000_B$	xx $xxxx_B$	A[12:11]	A[25:13]	A[10:1]
	$00\ 1010_B$		A[11:10]	A[25:13]	A[12], A[9:1]
	$00\ 1011_B$		A[12:11]	A[25:13]	A[10:1]
01_B	$00\ 1010_B$	$00\ 1101_B$	A[13], A[10]	A[25:14], A[12]	A[11], A[9:1]
	$00\ 1011_B$	$00\ 1101_B$	A[13], A[11]	A[25:14], A[12]	A[10:1]
	$00\ 1010_B$	$00\ 1111_B$	A[15], A[10]	A[25:16], A[14:12]	A[11], A[9:1]

Note 1. A[0] is not used because of 16-bit alignment.

15.6.2.1 Bank address as consecutive addresses (DBCONF.BKADM[1:0] = 00_B)

The position of the bank address in this case is specified with DBCONF.BKADP[5:0].

- BKADP[5:0] = 00 0000_B: Bank address is set to upper position of column address.
  i.e. column address[8:0] = A[9:1], bank address[1:0] = A[11:10]
- BKADP[5:0] ≠ 00 0000_B: Bank address can set the specified address position.
  i.e. BKADP[5:0] = 001100_B, column address[8:0] = A[9:1], bank address[1:0] = A[13:12]

If BKADP[5:0] ≠ 00 0000_B, BKADP[5:0] specifies the position of BA0.

The following tables show an example with one 32 M × 16-bit SDRAM and 16-bit external data bus width.

Table 15.32 BKADP[5:0] = 00 0000_B (consecutive BKADM[1:0] = 00_B)

Type		BA1	BA0	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M×16b	ROW	A11	A10	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
	COL	A11	A10	—	—	AP	—	A9	A8	A7	A6	A5	A4	A3	A2	A1

Table 15.33 BKADP[5:0] = 00 1010_B (consecutive BKADM[1:0] = 00_B)

Type		BA1	BA0	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M×16b	ROW	A11	A10	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
	COL	A11	A10	—	—	AP	—	A9	A8	A7	A6	A5	A4	A3	A2	A1

Table 15.34 BKADP[5:0] = 00 1100_B (consecutive BKADM[1:0] = 00_B)

Type		BA1	BA0	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M×16b	ROW	A13	A12	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A11	A10
	COL	A13	A12	—	—	AP	—	A9	A8	A7	A6	A5	A4	A3	A2	A1

15.6.2.2 Bank address as non-consecutive addresses (DBCONF.BKADM[1:0] = 01_B)

If DBCONF.BKADM[1:0] = 01_B, bank address position is controlled by DBCONF.BKADP[5:0] and DBCONF.BKADB[5:0].

BKADP[5:0] specifies the physical address position to BA0.
BKADB[5:0] specifies the physical address position to BA1.

Refer to Section 15.3.1.5, DBCONF – SDRAM configuration register for details about the register setting.

The following tables show an example with one 16 M × 16-bit SDRAM and 16-bit external data bus width.

Table 15.35 BKADP[5:0] = 00 1010_B and BKADB[5:0] = 00 1101_B (non-consecutive BKADM[1:0] = 01_B)

Type		BA1	BA0	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M×16b	ROW	A13	A10	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A12	A11
	COL	A13	A10	—	—	AP	—	A9	A8	A7	A6	A5	A4	A3	A2	A1

Table 15.36 BKADP[5:0] = 00 1100_B and BKADB[5:0] = 01 0000_B (non-consecutive BKADM[1:0] = 01_B)

Type		BA1	BA0	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M×16b	ROW	A16	A12	A24	A23	A22	A21	A20	A19	A18	A17	A15	A14	A13	A11	A10
	COL	A16	A12	—	—	AP	—	A9	A8	A7	A6	A5	A4	A3	A2	A1

15.6.2.3 Bank address setting combinations

The following combinations can be set for the bank address position setting of the SDRAM composition setting register.
All other settings must not be used.

Table 15.37 Bank address setting combinations

DBCONF.BKADM[1:0]	DBCONF.BKADP[5:0]	DBCONF.BKADB[5:0]
00_B	00 0000_B	–
	00 1010_B	
	00 1011_B	
	00 1100_B	
01_B	00 1010_B	00 1101_B
		00 1110_B
		00 1111_B
		01 0000_B
	00 1011_B	00 1101_B
		00 1110_B
		00 1111_B
		01 0000_B
	00 1100_B	00 1101_B
		00 1110_B
		00 1111_B
		01 0000_B

Section 16 DDR2-SDRAM Memory Controller (SDRB)

This section contains a generic description of the DDR2-SDRAM Memory Controller.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

16.1 Overview of RH850/D1L/D1M DDR2-SDRAM Memory Controller

16.1.1 Units

This microcontroller has the following number of units of the DDR2-SDRAM Memory Controller.

Table 16.1 Units

DDR2-SDRAM Memory Controller	D1L1	D1L2(H)	D1M1 D1M1-V2	D1M1H D1M1A	D1M2(H)
Units	–	–	–	–	1
Names	–	–	–	–	SDRB0

Units index n

Throughout this section, the individual units of a DDR2-SDRAM Memory Controller is identified by the index "n" (n = 0).

16.1.2 Register addresses

All DDR2-SDRAM Memory Controller register addresses are given as address offsets from the individual base addresses <SDRBn_base>.
The <SDRBn_base> addresses of each SDRBn are listed in the following table:

Table 16.2 Register base addresses <SDRBn_base>

SDRBn unit	<SDRBn_base> address
SDRB0	FFFA 0000_H

16.1.3 Clock supply

All DDR2-SDRAM Memory Controllers provide four clock inputs.

Table 16.3 Clock supply

SDRBn unit	SDRBn clock	Connected to
SDRB0	PBUS clock	Clock Controller C_ISO_PCLK
	MEMCLK	Clock Controller SDRBCLK / 2
	MEM2CLK	Clock Controller SDRBCLK
	Cross-connect clock	Clock Controller C_ISO_XCCLK

NOTE

The SDRBCK/$\overline{\text{SDRBCK}}$ memory clock signal output (non-inverted/inverted) frequency equals to MEMCLK (= SDRBCLK/2).

16.1.4 Interrupts

The DDR2-SDRAM Memory Controllers can generate the following interrupts:

Table 16.4 SDRBn interrupts

SDRBn signals	Function	Connected to
SDRB0:		
INTSDRB0	Refresh counter underflow interrupt	not connected

16.1.5 Reset sources

The DDR2-SDRAM Memory Controllers and their registers are initialized by the following reset signal:

Table 16.5 Reset sources

SDRBn unit	Reset signal
SDRBn	• Reset Controller SYSRES • Reset Controller SDRB0RES • reset upon wake-up from DEEPSTOP mode

CAUTION

By default the SDRB0RES reset is active.
Thus before accessing this module this reset must be released via the MRSTC register of the Reset Controller. Refer to Section 9, Reset Controller for details.

16.1.6 I/O signals

The following table shows the I/O signals of the DDR2-SDRAM Memory Controller.

Table 16.6 SDRBn I/O signals (1/2)

SDRBn signals	Function	Connected to
SDRB0		
SDRBA[13:11], SDRBA10, SDRBA[9:0]	Address output signals	Port SDRBA[13:11], SDRBA10, SDRBA[9:0]
SDRBDQ[31:0]	Data input/output signals	Port • D1M2: SDRBDQ[15:0] • D1M2H: SDRBDQ[31:0]
SDRBDQS[3:0]/ SDRBDQS[3:0]B	Data strobe input/output 3 to 0 (non-inverted/inverted)	Port • D1M2: SDRBDQS[1:0]/$\overline{\text{SDRBDQS[1:0]}}$ • D1M2H: SDRBDQS[3:0]/$\overline{\text{SDRBDQS[3:0]}}$
SDRBBA[2:0]	Bank address output signals	Port SDRBBA[2:0]
SDRBCK/SDRBCKB	Memory clock signal output (non-inverted/inverted)	Port SDRBCK/$\overline{\text{SDRBCK}}$
SDRBCKE	Clock enable output signal	Port SDRBCKE
SDRBDM[3:0]	Data mask input/output signals	Port • D1M2: SDRBDM[1:0] • D1M2H: SDRBDM[3:0]
SDRBODT	ODT output signal	Port SDRBODT

Table 16.6 SDRBn I/O signals (2/2)

SDRBn signals	Function	Connected to
SDRBCSB	Chip select output signal	Port $\overline{\text{SDRBCS}}$
SDRBCASB	CAS strobe output signal	Port $\overline{\text{SDRBCAS}}$
SDRBRASB	RAS strobe output signal	Port $\overline{\text{SDRBRAS}}$
SDRBWEB	Write enable output signal	Port $\overline{\text{SDRBWE}}$

Note 1. The suffix “B” in the signal names in the above table denotes low level active signals.

16.1.7 Partial Operation of I/O Signals

Table 16.7 D1M2H unused DDR2 interface handling

External SDRAM connection	Selection of DDR2 mode	Handling of unused data signals	Handling of unused address signals
32 bit	32 bit mode	• DQ[31:0]: all used • DQS[3:0]: all used	Leave open
16 bit	16 bit mode	• DQ[31:16]: leave open • DQS[3:2]: leave open	
None	32 bit mode	Refer to Section 2.8, Recommend connection of unused pins.	

Table 16.8 D1M2 unused DDR2 interface handling

External SDRAM connection	Selection of DDR2 mode	Handling of unused data signals	Handling of unused address signals
16 bit	16 bit mode	• DQ[15:0]: all used • DQS[1:0]: all used	Leave open
None	32 bit mode	Refer to Section 2.8, Recommend connection of unused pins.	

16.2 Overview

The DDR2-SDRAM Memory Controller gives access to external double data rate type 2 SDRAM memory devices.

Main features

- Supported memory devices: DDR2-SDRAM compliant with JEDEC JESD79-2F
- Data bus width and maximum bandwidth:
  - 16-bit data bus width: max. bandwidth 960 MB/s
  - 32-bit data bus width: max. bandwidth 1920 MB/s
- Support 32-bit data bus width only 2 x 16-bit DDR2-SDRAM devices
- Up to 512 MByte external DDR2-SDRAM
- Multi bank operation (4 or 8 banks selectable)
- Timing and address size freely programmable
- Burst length
  - 16-bit DDR2-SDRAM devices: 8
  - 32-bit DDR2-SDRAM devices: 4 or 8 selectable
- Early precharge in case of page miss (different bank)
- Refresh
  average interval and the maximum post count selectable
- Auto power-down
  automatic transition to power-down after selectable inactive period
- Four ports to cross-connect systems with Dynamic Priority Generator

NOTE

The DDR2-SDRAM Memory Controller (SDRB) in 32-bit mode can only be used in D1M2H devices.
The D1M2 devices supports 16-bit mode DDR2-SDRAM only, where D1M2H supports 32-bit and 16-bit mode.

For partial operation of DDR2-SDRAM Memory Controller (e.g. usage with 16-bit DDR2-SDRAM) please refer to the Section 16.1.7, Partial Operation of I/O Signals.

16.3 DDR2-SDRAM Memory Controller Registers

This section contains a description of all registers of Dynamic Priority Generator, Arbiter and the DDR2-SDRAM Memory Controller.

Table 16.9 DDR2-SDRAM Memory Controller registers overview (1/2)

Module Name	Register Name	Symbol	Address
DDR2-SDRAM I/F control registers			
SDRBn	Error status register	DBSVCR	<SDRBn_base> + 004_H
SDRBn	Status register 0	DBSTATE0	<SDRBn_base> + 008_H
SDRBn	SDRAM access enable register	DBACEN	<SDRBn_base> + 010_H
SDRBn	Auto-refresh enable register	DBRFEN	<SDRBn_base> + 014_H
SDRBn	Manual SDRAM command register	DBCMD	<SDRBn_base> + 018_H
SDRBn	Operating completion waiting register	DBWAIT	<SDRBn_base> + $01C_H$
SDRBn	SDRAM type setting register	DBKIND	<SDRBn_base> + 020_H
SDRBn	SDRAM configuration setting register 0	DBCONF0	<SDRBn_base> + 024_H
SDRBn	DDR2-PHY type setting register	DBPHYTYPE	<SDRBn_base> + 030_H
SDRBn	SDRAM timing register 0	DBTR0	<SDRBn_base> + 040_H
SDRBn	SDRAM timing register 1	DBTR1	<SDRBn_base> + 044_H
SDRBn	SDRAM timing register 2	DBTR2	<SDRBn_base> + 048_H
SDRBn	SDRAM timing register 3	DBTR3	<SDRBn_base> + 050_H
SDRBn	SDRAM timing register 4	DBTR4	<SDRBn_base> + 054_H
SDRBn	SDRAM timing register 5	DBTR5	<SDRBn_base> + 058_H
SDRBn	SDRAM timing register 6	DBTR6	<SDRBn_base> + $05C_H$
SDRBn	SDRAM timing register 7	DBTR7	<SDRBn_base> + 060_H
SDRBn	SDRAM timing register 8	DBTR8	<SDRBn_base> + 064_H
SDRBn	SDRAM timing register 9	DBTR9	<SDRBn_base> + 068_H
SDRBn	SDRAM timing register 10	DBTR10	<SDRBn_base> + $06C_H$
SDRBn	SDRAM timing register 11	DBTR11	<SDRBn_base> + 070_H
SDRBn	SDRAM timing register 12	DBTR12	<SDRBn_base> + 074_H
SDRBn	SDRAM timing register 13	DBTR13	<SDRBn_base> + 078_H
SDRBn	SDRAM timing register 14	DBTR14	<SDRBn_base> + $07C_H$
SDRBn	SDRAM timing register 15	DBTR15	<SDRBn_base> + 080_H
SDRBn	SDRAM timing register 16	DBTR16	<SDRBn_base> + 084_H
SDRBn	SDRAM timing register 17	DBTR17	<SDRBn_base> + 088_H
SDRBn	SDRAM timing register 18	DBTR18	<SDRBn_base> + $08C_H$
SDRBn	SDRAM burst length register	DBBL	<SDRBn_base> + $0B0_H$
SDRBn	SDRB operation adjustment register 0	DBADJ0	<SDRBn_base> + $0C0_H$
SDRBn	SDRB operation adjustment register 2	DBADJ2	<SDRBn_base> + $0C8_H$
SDRBn	Refresh configuration register 0	DBRFCNF0	<SDRBn_base> + $0E0_H$
SDRBn	Refresh configuration register 1	DBRFCNF1	<SDRBn_base> + $0E4_H$
SDRBn	Refresh configuration register 2	DBRFCNF2	<SDRBn_base> + $0E8_H$
SDRBn	ODT operation configuration register0	DBRNK0	<SDRBn_base> + 100_H
SDRBn	Power down configuration register	DBPDNCNF	<SDRBn_base> + 180_H
SDRBn	PHY control register 0	DBPDCNT0	<SDRBn_base> + 200_H
SDRBn	PHY control register 1	DBPDCNT1	<SDRBn_base> + 204_H
SDRBn	PHY control register 3	DBPDCNT3	<SDRBn_base> + $20C_H$

Table 16.9 DDR2-SDRAM Memory Controller registers overview (2/2)

Module Name	Register Name	Symbol	Address
SDRAM bus control registers			
SDRBn	Bus control unit 0 control register 1	DBBS0CNT1	<SDRBn_base> + 304_H
Dynamic priority control registers port index (i = 0 to 7)			
SDRBn	Dynamic Priority Generator control register	DBLGCNTi	<SDRBn_base> + i x 100_H + 1000_H
SDRBn	QoS counter initial value register	DBTMVAL0i	<SDRBn_base> + i x 100_H + 1004_H
SDRBn	QoS counter operation state transition condition register	DBRQCTRi	<SDRBn_base> + i x 100_H + 1014_H
SDRBn	Priority level threshold register 0	DBTHRES0i	<SDRBn_base> + i x 100_H + 1018_H
SDRBn	Priority level threshold register 1	DBTHRES1i	<SDRBn_base> + i x 100_H + $101C_H$
SDRBn	Priority level threshold register 2	DBTHRES2i	<SDRBn_base> + i x 100_H + 1020_H
SDRBn	Dynamic Priority Generator status register	DBLGSTSi	<SDRBn_base> + i x 100_H + 1024_H
SDRBn	Dynamic Priority Generator enable register	DBLGQONi	<SDRBn_base> + i x 100_H + 1028_H

<SDRBn_base>

The base addresses <SDRBn_base> of the SDRBn is defined in the first section of this chapter under the key word “Register addresses”.

NOTES

1. In the header files the names of the above registers are defined in the following format: <ModuleName> + <Symbol>.
   <ModuleName> and <Symbol> are defined in the above table.
2. Memory access right after register access has possibility to cause the racing between two accesses via different routes.
   Therefore, the last accessed register value must be confirmed at least before memory access. Refer to Section 3.4.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation for details.

16.3.1 DDR2-SDRAM Memory Controller control registers details

16.3.1.1 DBSVCR – Error status register

This register holds various error indication flags.

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 004_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	INACTIVE	0	0	BAD_OPC	0	0	BAD_ADDR	ERR_SNT	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R/W	R	R	R/W	R/W	R

Table 16.10 DBSVCR register contents (1/2)

Bit position	Bit name	Function
31 to 9	Reserved	Reading returns the initial value. When written, write the initial value.
8	INACTIVE	Error cause: Data access to SDRAM during SDRAM disable At read: 0: no error 1: SDRAM data was accessed while the SDRAM access is disabled, i.e. while DBACEN.ACCEN = 0. At write access: write 0: clear INACTIVE write 1: no function
7 to 6	Reserved	Reading returns the initial value. When written, write the initial value.
5	BAD_OPC	Error cause: Not supported transaction of the XC cross-connect At read: 0: no error 1: unsupported transaction of the XC cross-connect*1 At write access: write 0: clear BAD_OPC write 1: no function
4 to 3	Reserved	Reading returns the initial value. When written, write the initial value.
2	BAD_ADDR	Error cause: Access to undefined register of SDRB At read: 0: no error 1: access to undefined register of SDRB has occurred*2 At write access: write 0: clear BAD_ADDR write 1: no function
1	ERR_SNT	Error flag: Slave error issued to cross-connect At read: 0: no error 1: Slave error was returned to the cross-connect The error cause can be analyzed by the INACTIVE and BAD_OPC error flags. At write access: write 0: clear ERR_SNT write 1: no function

Table 16.10 DBSVCR register contents (2/2)

Bit position	Bit name	Function
0	Reserved	Reading returns the initial value. When written, write the initial value.

Note 1. The cause for unsupported transactions on XC cross-connect are unsupported data block width (bigger than 128 bit) or burst length errors (burst access to fixed address or unaligned bursts).
SDRB can detect those illegal transactions, even though they can not occur by normal cross-connect operation.

Note 2. Error cause of “undefined register access” means PBUS access to register addresses of SDRB, where no SDRB register is present.

The error causes indicated by the DBSVCR register (DBSVCR.INACTIVE, DBSVCR.BAD_OPC, DBSVCR.BAD_ADDR and DBSVCR.ERR_SNT) are signalled to the CPU either directly by a SYSERR exception or INTBUSERR internal bus error interrupt respectively.

The signalling mechanism used, depends on the accessing master causing the error.

- Errors caused by CPU (master) accessing the SDRB are signalled by SYSERR exception.
- Errors caused by other masters accessing the SDRB are signalled by INTBUSERR interrupt.

16.3.1.2 DBSTATE0 – Status register 0

This register indicates whether transactions are ongoing via each port.

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 008_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	AXBSY3	AXBSY2	AXBSY1	AXBSY0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 16.11 DBSTATE0 register contents

Bit position	Bit name	Function
31 to 4	Reserved	Reading returns the initial value. When written, write the initial value.
3	AXBSY3	Port 3 busy indication 0: no transaction ongoing via the port 1: transaction ongoing via the port
2	AXBSY2	Port 2 busy indication 0: no transaction ongoing via the port 1: transaction ongoing via the port
1	AXBSY1	Port 1 busy indication 0: no transaction ongoing via the port 1: transaction ongoing via the port
0	AXBSY0	Port 0 busy indication 0: no transaction ongoing via the port 1: transaction ongoing via the port

16.3.1.3 DBACEN – SDRAM access enable register

This register is used to enable or disable the SDRAM access.

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 010_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ACCEN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.12 DBACEN register contents

Bit position	Bit name	Function
31 to 1	Reserved	Reading returns the initial value. When written, write the initial value.
0	ACCEN	SDRAM access enable/disable 0: SDRAM access disabled 1: SDRAM access enabled

NOTE

If the SDRAM is accessed while DBACEN.ACCEN = 0, the error INACTIVE flag in the error status register DBSVCR is set.

16.3.1.4 DBRFEN – Auto-refresh enable register

This register is used to enable or disable the SDRAM access.

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 014_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ARFEN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.13 DBRFEN register contents

Bit position	Bit name	Function
31 to 1	Reserved	Reading returns the initial value. When written, write the initial value.
0	ARFEN	Auto-refresh enable/disable 0: Auto-refresh stopped 1: Auto-refresh started Setting ARFEN = 1 resets the refresh counter and starts issuing auto-refresh commands in regular intervals. The refresh cycle time and other settings depend on the values held in the refresh configuration registers DBRFCNF0 to DBRFCNF2.

16.3.1.5 DBCMD – Manual SDRAM command-issuing register

This register is used to send commands to the SDRAM devices.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 018_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	OPC[5:0]						0	0	0	0	0	RANK[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ARG[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.14 DBCMD register contents

Bit position	Bit name	Function
31 to 30	Reserved	These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
29 to 24	OPC[5:0]	Operation Code Bits Specify the type of command to be issued. Refer to Table 16.15. If Wait is specified through these bits, valid SDRAM commands are not output and no processing is performed except reserving the time until the next operation. These bits are read as undefined values.
23 to 19	Reserved	These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.
18 to 16	RANK[2:0]	RANK assignment 000_B: RANK0 100_B: all RANK All others: Setting prohibited The meaning depends on the operation code OPC, refer to Table 16.15.
15 to 0	ARG[15:0]	Parameter Bits The meaning can differ according to the operation code indicated by the OPC[5:0] bits. When OPC[5:0] indicates ModeRegisterSet (MRS0 to MRS3), the ARG[15:0] bits specify the value to be issued on the address pins (MA) of SDRAM. When OPC[5:0] is any other value, the ARG[15:0] bits specify the minimum interval to issuing of the next command in SDRAM cycles. When ARG[15:0] 0, however, the values default to those given in Interval cycle column of Table 16.15. These bits are read as undefined values.

The Manual SDRAM command-issuing register (DBCMD) is used to issue the required commands for the sequence of initializing the SDRAM and of transitions to and from the self-refresh mode. The command corresponding to the OPC[5:0] bits is issued once as a result of writing to this register. For instance, issuing the refresh command twice requires that "00 1100_B" be written to the OPC[5:0] bits

twice. Do not write to this register while access to the SDRAM is enabled (DBACEN.ACCEN = 1). The timing with which an operation is complete (i.e. the timing with which the specified SDRAM command is output to the PHY unit from SDRB) may be later than the response of SDRB to writing to this register. If you wish to wait until actual output of the specified SDRAM command to SDRAM; do this by reading the DBWAIT register (described later).

The length of these periods are given in the "interval cycle" column of Table 16.15. They can also be customized by using the ARG[15:0] bits (except in cases where the OPC[5:0] bits indicate MRS0 to MRS3).

Table 16.15 Manual command function

OPC[5:0]	Symbol	Operation	Interval cycle (MCK cycle)	RANK function	ARG function
$00\ 0000_B$	Wait	Device deselected (insert the wait cycle)	4	Don't care	Customization of intervals (If ARG[15:0] = 0 when use the left column value)
$00\ 1011_B$	PreA	Precharge All	TRPA[4:0]	Assign to target RANK	
$00\ 1100_B$	Ref	Refresh	TRFC[8:0]		
$01\ 0000_B$	PDEn	Power Down Entry	4		
$01\ 0001_B$	PDXt	Power Down Exit	4		
$01\ 1000_B$	SREn	Self-Refresh Entry	4		
$01\ 1001_B$	SRXt	Self-Refresh Exit	TRFC[8:0]		
$10\ 1000_B$	MRS0	ModeRegisterSet (MRS/MR0)(DDR2)	TMOD[5:0]		Value to be set in the mode register of SDRAM is set
$10\ 1001_B$	MRS1	ModeRegisterSet (MRS/MR1)(DDR2)	TMOD[5:0]		
$10\ 1010_B$	MRS2	ModeRegisterSet (MRS/MR2)(DDR2)	TMOD[5:0]		
$10\ 1011_B$	MRS3	ModeRegisterSet (MRS/MR3)(DDR2)	TMOD[5:0]		

When "Wait" (OPC[5:0] = 00 0000B) is specified, the valid SDRAM command is not output, and only the time interval between next operation will be secured.

When OPC[5:0] is ModeRegisterSet (MRS0 to MRS3), the value that is output to Address terminal (MA) of SDRAM is set.

When OPC[5:0] is not ModeRegisterSet (MRS0 to MRS3), the minimum interval until the issue of the next command is set in the SDRAM cycle count. However, when SRG = 0, the default value which is shown in the "Interval cycle" column of Table 16.15 will be used.

Note: The value which was set in the timing register, which will be mentioned later, is used for TRPA[4:0] and TMOD[5:0] of "Interval cycle" column of Table 16.15.

Note: Write in this register only when auto refresh function has been stopped (ARFEN = 0 of DBRFEN register). However, the sequence defined in 3rd chapter SDRB function explanation does not have this limitation. When OPC = Wait is written in this register during auto refresh function operation, this may interrupt the issuing of the refresh command by auto refresh function during the secured time interval.

16.3.1.6 DBWAIT – Operating completion waiting register

This register is used to wait until command issued by DBCMD register is completed.

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + $01C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WAIT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.16 DBWAIT register contents

Bit position	Bit name	Function
31 to 1	Reserved	Reading returns the initial value. When written, write the initial value.
0	WAIT	Operation completion waiting bit.*1 This bit will always be read as 0.

Note 1. The completion of the read access to this register bit (DBWAIT.WAIT) will be delayed until the SDRB finished the current command execution. By this means the SDRB can ensure that the waiting times (e.g. after command output from the SDRB to the PHY unit) issued by DBCMD register are completed.

16.3.1.7 DBKIND – SDRAM type setting register

This register defines the SDRAM type.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 020_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	DDCG[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 16.17 DBKIND register contents

Bit position	Bit name	Function
31 to 3	Reserved	Reading returns the initial value. When written, write the initial value.
2 to 0	DDCG[2:0]	SDRAM type selection 101_B: DDR2-SDRAM All others: Setting prohibited

Only write to register DBKIND during “Initial Sequence - 1st setting of SDRB” in **Section 16.4.1, Initial sequence**.

16.3.1.8 DBCONF0 – SDRAM configuration setting register 0

This register is used to specify the SDRAM configuration.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 024_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	AWRW0[4:0]					0	0	0	AWRK0	0	0	AWBK0[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	AWCL0[3:0]				0	0	0	0	0	0	DW0[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 16.18 DBCONF0 register contents (1/2)

Bit position	Bit name	Function
31 to 29	Reserved	Reading returns the initial value. When written, write the initial value.
28 to 24	AWRW0[4:0]	Row bit width $0\ 1101_B$: 13 bit $0\ 1110_B$: 14 bit $0\ 1111_B$: 15 bit $1\ 0000_B$: 16 bit All others: Setting prohibited
23 to 21	Reserved	Reading returns the initial value. When written, write the initial value.
20	AWRK0	Number of Rank 0: 1 rank All others: Setting prohibited
19 to 18	Reserved	Reading returns the initial value. When written, write the initial value.
17 to 16	AWBK0[1:0]	Number of banks 10_B: 4 banks 11_B: 8 banks All others: Setting prohibited
15 to 12	Reserved	Reading returns the initial value. When written, write the initial value.
11 to 8	AWCL0[3:0]	Column bit width 1001_B: 9 bit 1010_B: 10 bit All others: Setting prohibited
7 to 2	Reserved	Reading returns the initial value. When written, write the initial value.
1 to 0	DW0[1:0]	External data bus width*1 01_B: 16 bit 10_B: 32 bit All others: Setting prohibited

Table 16.18 DBCONF0 register contents (2/2)

Bit position	Bit name	Function
Note 1.		If no DDR2-SDRAM device is connected to the SDRB, select DBCONF0.DW0[1:0] = 10_B: 32-bit external data bus width. Refer also to the Section 2.8, Recommend connection of unused pins for SDRB pin handling.

16.3.1.9 DBPHYTYPE – DDR2-PHY type register

This register is used to specify the type of the DDR2-PHY.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 030_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PHYTYPE[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 16.19 DBPHYTYPE register contents

Bit position	Bit name	Function
31 to 2	Reserved	Reading returns the initial value. When written, write the initial value.
1 to 0	PHYTYPE[1:0]	DDR2-PHY type 00_B: MSPAD1 All others: Setting prohibited **CAUTION** The default value of this register must not be changed.

16.3.1.10 DBTR0 – SDRAM timing register 0

This register is used to define the CAS latency.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 040_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	CL[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 16.20 DBTR0 register contents

Bit position	Bit name	Function
31 to 4	Reserved	Reading returns the initial value. When written, write the initial value.
3 to 0	CL[3:0]	CAS latency 0011_B: 3 cycles 0100_B: 4 cycles 0101_B: 5 cycles All others: Setting prohibited

16.3.1.11 DBTR1 – SDRAM timing register 1

This register is used to define the CAS write latency.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 044_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	CWL[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 16.21 DBTR1 register contents

Bit position	Bit name	Function
31 to 4	Reserved	Reading returns the initial value. When written, write the initial value.
3 to 0	CWL[3:0]	CAS write latency 0010_B: 2 cycles 0011_B: 3 cycles 0100_B: 4 cycles All others: Setting prohibited For DDR2-SDRAM types, set CAS write latency to 1 cycles less than CAS latency, i.e. CWL[3:0] = DBTR0.CL[3:0] - 1.

16.3.1.12 DBTR2 – SDRAM timing register 2

This register is used to define the additive latency.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 048_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	AL[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 16.22 DBTR2 register contents

Bit position	Bit name	Function
31 to 4	Reserved	Reading returns the initial value. When written, write the initial value.
3 to 0	AL[3:0]	Additive latency 0000_B: 0 cycles All others: Setting prohibited

16.3.1.13 DBTR3 – SDRAM timing register 3

This register is used to set a timing parameter for the SDRAM.

CAUTIONS

1. **Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).**
2. **Writing to this register is only permitted, while automatic refresh function is stopped (DBRFEN.ARFEN).**

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 050_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	TRCD[4:0]				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 16.23 DBTR3 register contents

Bit position	Bit name	Function
31 to 5	Reserved	Reading returns the initial value. When written, write the initial value.
4 to 0	TRCD[4:0]	ACT-READ/WRITE interval setting bits 0 0011_B: 3 cycles 0 0100_B: 4 cycles ... 1 1111_B: 31 cycles All others: Setting prohibited TRCD[4:0] set the minimum interval from an ACT command to a READ/WRITE command.

16.3.1.14 DBTR4 – SDRAM timing register 4

This register is used to define the precharge interval.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 054_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	TRPA[4:0]				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	TRP[4:0]				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 16.24 DBTR4 register contents

Bit position	Bit name	Function
31 to 21	Reserved	Reading returns the initial value. When written, write the initial value.
20 to 16	TRPA[4:0]	PREA interval $0\ 0011_B$: 3 cycles $0\ 0100_B$: 4 cycles ... $1\ 1111_B$: 31 cycles All others: Setting prohibited TRPA[4:0] sets the minimum interval from a PREA (precharge all banks) command to an ACT/REF command. TRPA[4:0] must be greater than or equal to TRP[4:0].
15 to 5	Reserved	Reading returns the initial value. When written, write the initial value.
4 to 0	TRP[4:0]	PRE interval $0\ 0011_B$: 3 cycles $0\ 0100_B$: 4 cycles ... $1\ 1111_B$: 31 cycles All others: Setting prohibited TRP[4:0] sets the minimum interval from a PRE (precharge) command to an ACT/REF command.

16.3.1.15 DBTR5 – SDRAM timing register 5

This register is used to define the ACT-ACT/REF interval.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 058_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	TRC[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.25 DBTR5 register contents

Bit position	Bit name	Function
31 to 6	Reserved	Reading returns the initial value. When written, write the initial value.
5 to 0	TRC[5:0]	ACT-ACT/REF interval 00 1000_B: 8 cycles 00 1001_B: 9 cycles ... 11 1111_B: 63 cycles All others: Setting prohibited TRC[5:0] sets the minimum interval from one ACT command to another ACT command (for the same bank) or to a REF command.

16.3.1.16 DBTR6 – SDRAM timing register 6

This register is used to define the ACT-PRE interval.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + $05C_H$

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	TRAS[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.26 DBTR6 register contents

Bit position	Bit name	Function
31 to 6	Reserved	Reading returns the initial value. When written, write the initial value.
5 to 0	TRAS[5:0]	ACT-PRE interval 00 0101_B: 5 cycles 00 0110_B: 6 cycles ... 11 1111_B: 63 cycles All others: Setting prohibited TRAS[5:0] sets the minimum interval from an ACT command to a PRE command.

16.3.1.17 DBTR7 – SDRAM timing register 7

This register is used to define the ACT(A)-ACT(B) interval.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 060_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	TRRD[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 16.27 DBTR7 register contents

Bit position	Bit name	Function
31 to 4	Reserved	Reading returns the initial value. When written, write the initial value.
3 to 0	TRRD[3:0]	ACT(A)-ACT(B) interval 0010_B: 2 cycles 0011_B: 3 cycles ... 1111_B: 15 cycles All others: Setting prohibited TRRD[3:0] sets the minimum interval between ACT commands issued for different banks.

16.3.1.18 DBTR8 – SDRAM timing register 8

This register is used to define the Four Activate window length.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 064_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	TFAW[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.28 DBTR8 register contents

Bit position	Bit name	Function
31 to 8	Reserved	Reading returns the initial value. When written, write the initial value.
7 to 0	TFAW[7:0]	Four Activate window length 0000 1000_B: 8 cycles 0000 1001_B: 9 cycles ... 0011 1111_B: 63 cycles All others: Setting prohibited TFAW[7:0] sets the length of the four activate window. TFAW[7:0] must be at least four times as large as the minimum interval between ACT commands issued for different banks, i.e. TFAW[7:0] ≥ 4 × DBTR7.TRRD[3:0].

16.3.1.19 DBTR9 – SDRAM timing register 9

This register is used to define the READ-PRE interval.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 068_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	TRDPR[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 16.29 DBTR9 register contents

Bit position	Bit name	Function
31 to 4	Reserved	Reading returns the initial value. When written, write the initial value.
3 to 0	TRDPR[3:0]	READ-PRE interval*1 0010_B: 2 cycles 0011_B: 3 cycles ... 1111_B: 15 cycles All others: Setting prohibited TRDPR[3:0] sets the minimum interval from a READ command to a PRE command.

Note 1. For JEDEC compliant DDR2-SDRAM the READ-PRE interval can be calculated with the following formula.
TRDPR[3:0] = BL[1:0]/2 + t_{RTP_cyc} - 2
Where the t_{RTP_cyc} need to be calculated to be
t_{RTP_cyc} = max (2, t_{RTP}/t_{CK})

16.3.1.20 DBTR10 – SDRAM timing register 10

This register is used to define the write recovery period.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + $06C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	TWR[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 16.30 DBTR10 register contents

Bit position	Bit name	Function
31 to 4	Reserved	Reading returns the initial value. When written, write the initial value.
3 to 0	TWR[3:0]	Write recovery period 0010_B: 2 cycles 0011_B: 3 cycles ... 1111_B: 15 cycles All others: Setting prohibited

16.3.1.21 DBTR11 – SDRAM timing register 11

This register is used to define the READ-WRITE interval.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 070_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	TRDWR[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.31 DBTR11 register contents

Bit position	Bit name	Function
31 to 6	Reserved	Reading returns the initial value. When written, write the initial value.
5 to 0	TRDWR[5:0]	READ-WRITE interval*1 00 0100_B: 4 cycles 00 0101_B: 5 cycles ... 00 1111_B: 15 cycles All others: Setting prohibited TRDWR[5:0] sets the minimum interval from a READ command to a WRITE command.

Note 1. The READ-WRITE interval for JEDEC standard compliant DDR2-SDRAM can be calculated with the formula:
$TRDWR[5:0] \geq BL[1:0]/2 + 3 + db\_enoffset/2$

16.3.1.22 DBTR12 – SDRAM timing register 12

This register is used to define the WRITE-READ interval.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 074_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	TWRRD[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.32 DBTR12 register contents

Bit position	Bit name	Function
31 to 6	Reserved	Reading returns the initial value. When written, write the initial value.
5 to 0	TWRRD[5:0]	WRITE-READ interval*1 00 0110_B: 6 cycles 00 0111_B: 7 cycles ... 01 1111_B: 31 cycles All others: Setting prohibited TWRRD[5:0] sets the minimum interval from a WRITE command to a READ command.

Note 1. The WRITE-READ interval for JEDEC standard compliant DDR2-SDRAM can be calculated with the formula:
TWRRD$[5:0]$ >= CWL$[3:0]$ + BL$[1:0]$/2 + t_{WTR}

16.3.1.23 DBTR13 – SDRAM timing register 13

This register is used to define the REF-ACT/REF interval.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 078_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	TRFC[8:0]								
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.33 DBTR13 register contents

Bit position	Bit name	Function
31 to 9	Reserved	Reading returns the initial value. When written, write the initial value.
8 to 0	TRFC[8:0]	REF-ACT/REF interval 008_H: 8 cycles 009_H: 9 cycles ... $1FF_H$: 511 cycles All others: Setting prohibited TRFC[8:0] sets the minimum interval from a REF (refresh) command to an ACT/REF command.

16.3.1.24 DBTR14 – SDRAM timing register 14

This register is used to define the CKEH(DLL-LOCK) and CKEH period.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + $07C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	TCKEHDLL[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	TCKEH[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.34 DBTR14 register contents

Bit position	Bit name	Function
31 to 24	Reserved	Reading returns the initial value. When written, write the initial value.
23 to 16	TCKEHDLL[7:0]	CKEH(DLL-LOCK) period 02_H: 2 cycles 03_H: 3 cycles ... $1F_H$: 31 cycles All others: Setting prohibited TCKEHDLL[7:0]] sets the minimum interval from the time the CKE signal goes high until the issuing of a further valid command that requires the DLL to be locked.
15 to 8	Reserved	Reading returns the initial value. When written, write the initial value.
7 to 0	TCKEH[7:0]	CKEH period*1 02_H: 2 cycles 03_H: 3 cycles ... $0F_H$: 15 cycles All others: Setting prohibited TCKEH[7:0] sets the minimum interval from the time the CKE signal goes high until the issuing of a further valid command.

Note 1. Please ensure that the register setup meets the condition
DBTR14.TCKEH[7:0] ≤ DBTR14.TCKEHDLL[7:0]

16.3.1.25 DBTR15 – SDRAM timing register 15

This register is used to define the CKEL period.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 080_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	TCKEL[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 16.35 DBTR15 register contents

Bit position	Bit name	Function
31 to 4	Reserved	Reading returns the initial value. When written, write the initial value.
3 to 0	TCKEL[3:0]	CKEL period 0010_B: 2 cycles 0011_B: 3 cycles ... 1111_B: 15 cycles All others: Setting prohibited TCKEL[3:0] sets the minimum time from the time the CKE signal goes low until it goes high.

16.3.1.26 DBTR16 – SDRAM timing register 16

This register is used to define the latency timing for the DDR2 PHY.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 084_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DQIENLTNCY[3:0]				0	0	DQLOFFSET[1:0]		0	0	DQL[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DQENLTNCY[3:0]				0	0	0	0	0	0	0	0	WDQL[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 16.36 DBTR16 register contents (1/2)

Bit position	Bit name	Function
31 to 28	DQIENLTNCY[3:0]	DQIENLTNCY setting 0001_B: 1 cycle Others: setting prohibited DQIENLTNCY[3:0] sets the latency from issuing of a read command to the PHY unit the dqi_en signal is output.
27 to 26	Reserved	Reading returns the initial value. When written, write the initial value.
25 to 24	DQLOFFSET[1:0]	DQLTNCY offset 10_B: 2 cycles All others: Setting prohibited DQLOFFSET[1:0] sets the additional latency necessary for SDRB internal operation when a read command is issued to the PHY unit.
23 to 22	Reserved	Reading returns the initial value. When written, write the initial value.
21 to 16	DQL[5:0]	DQLTNCY setting 0 0110_B: 6 cycles 0 0111_B: 7 cycles 0 1000_B: 8 cycles 0 1001_B: 9 cycles 0 1010_B: 10 cycles All others: Setting prohibited DQL[5:0] sets the latency from issuing of a read command to the PHY unit until the read data is returned from the PHY unit. Set the following value for DQL[5:0]: DQL[5:0] = CL[3:0] + 3 + db_offset
15 to 12	DQENLTNCY[3:0]	DQENLTNCY setting 0001_B: 1 cycle All others: Setting prohibited DQENLTNCY[3:0] sets the latency from issuing of a write command to the PHY unit until the dqi_en signal is output.
11 to 4	Reserved	Reading returns the initial value. When written, write the initial value.

Table 16.36 DBTR16 register contents (2/2)

Bit position	Bit name	Function
3 to 0	WDQL[3:0]	WDQLTNCY setting 0001_B: 1 cycle All others: Setting prohibited WDQL[3:0] sets the latency from issuing of a write command until the write data is output.

16.3.1.27 DBTR17 – SDRAM timing register 17

This register is used to define the MRS/MRW interval.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 088_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	TMOD[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 16.37 DBTR17 register contents

Bit position	Bit name	Function
31 to 22	Reserved	Reading returns the initial value. When written, write the initial value.
21 to 16	TMOD[5:0]	MRS/MRW interval 00 0010_B: 2 cycles 00 0011_B: 3 cycles ... 00 1111_B: 15 cycles All others: Setting prohibited TMOD[5:0] indicates the minimum interval from an MRS (mode register set) / MRW (mode register write) command to a subsequent command.
15 to 0	Reserved	Reading returns the initial value. When written, write the initial value.

16.3.1.28 DBTR18 – SDRAM timing register 18

This register is used to define the ODT (On-Die-Termination) terminal control timing settings.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + $08C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	RODTL[2:0]			0	0	0	0	0	RODTA[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	WODTL[2:0]			0	0	0	0	0	WODTA[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 16.38 DBTR18 register contents (1/2)

Bit position	Bit name	Function
31 to 27	Reserved	Reading returns the initial value. When written, write the initial value.
26 to 24	RODTL[2:0]	ODT assert period for read 000_B: BL[1:0]/2 cycles 001_B: BL[1:0]/2 + 1 cycles ... 111_B: BL[1:0]/2 + 7 cycles RODTL[2:0] sets the assert period of the ODT signal that is output when a read command is output.
23 to 19	Reserved	Reading returns the initial value. When written, write the initial value.
18 to 16	RODTA[2:0]	ODT assert start timing for read 000_B: Simultaneous with read command 001_B: after 1 cycle of read command 010_B: after 2 cycles of read command 011_B: after 3 cycles of read command 111_B: before 1 cycle of read command All others: Setting prohibited RODTA[2:0] sets the assert start timing for the ODT signal that is output when a read command is output.
15 to 11	Reserved	Reading returns the initial value. When written, write the initial value.
10 to 8	WODTL[2:0]	ODT assert period for write 000_B: BL[1:0]/2 cycles 001_B: BL[1:0]/2 + 1 cycles ... 111_B: BL[1:0]/2 + 7 cycles WODTL[2:0] sets the assert period of the ODT signal that is output when a write command is output.
7 to 3	Reserved	Reading returns the initial value. When written, write the initial value.

Table 16.38 DBTR18 register contents (2/2)

Bit position	Bit name	Function
2 to 0	WODTA[2:0]	ODT assert start timing for write 000_B: Simultaneous with write command 001_B: after 1 cycle of write command 010_B: after 2 cycles of write command 011_B: after 3 cycles of write command 111_B: before 1 cycle of write command All others: Setting prohibited WODTA[2:0] sets the assert start timing for the ODT signal that is output when a write command is output.

16.3.1.29 DBBL – SDRAM burst length register

This register is used to define the burst length.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + $0B0_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	BL[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 16.39 DBBL register contents

Bit position	Bit name	Function
31 to 2	Reserved	Reading returns the initial value. When written, write the initial value.
1 to 0	BL[1:0]	Burst length selection 00_B: 8 10_B: 4 All others: Setting prohibited BL[1:0] specifies the burst length of SDRAM. If DDR2-SDRAM bus width is 16 bit, then BL[1:0] should be set to 10_B.

16.3.1.30 DBADJ0 – SDRB operation adjustment register 0

This register is used to control of SDRB operation.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + $0C0_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FREQRATIO[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CAMODE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.40 DBADJ0 register contents

Bit position	Bit name	Function
31 to 18	Reserved	Reading returns the initial value. When written, write the initial value.
17 to 16	FREQ RATIO[1:0]	Frequency ratio 01_B: 1:2 SDRB: memory frequency ratio All others: Setting prohibited FREQRATIO[1:0] specifies the clock frequency ratio of PHY and SDRB.
15 to 1	Reserved	Reading returns the initial value. When written, write the initial value.
0	CAMODE	Command/Address output mode 0: 1 command issued per 1 clock 1: 1 command issued per 2 clocks CAMODE specifies the SDRAM command/address output mode. • CAMODE = 0: the SDRB outputs a single command per clock cycle. • CAMODE = 1: the SDRB outputs a single command per two clock cycles. In this case, command signals and address signals, except for the MCS[1:0] signals of SDRAM, are kept constant for two clock cycles. During this period, the CS signal becomes low only in the latter one clock cycle

16.3.1.31 DBADJ2 – SDRB operation adjustment register 2

This register is used to control of SDRB operation.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + $0C8_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ACAPC0[7:0]								0	0	0	0	ACAP0[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 16.41 DBADJ2 register contents

Bit position	Bit name	Function
31 to 16	Reserved	Reading returns the initial value. When written, write the initial value.
15 to 8	ACAPC0[7:0]	Acceptable data cell count bits by device control unit 01_H: 1 cell 02_H: 2 cells ... 20_H: 32 cells All others: Setting prohibited ACAPC0[7:0] sets the number of requests acceptable by the device control unit in the SDRB in data cell units.
7 to 4	Reserved	Reading returns the initial value. When written, write the initial value.
3 to 0	ACAP0[3:0]	Acceptable transaction count bits by device control unit 1_H: 1 transaction 2_H: 2 transactions ... 8_H: 8 transactions All others: Setting prohibited ACAP0[3:0] sets the number of requests acceptable by the device control unit in the SDRB in transaction units.

16.3.1.32 DBRFCNF0 – Refresh configuration register 0

DBRFCNF0 is used to set the timing for refreshing of the SDRAM.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + $0E0_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	REFTHF[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.42 DBRFCNF0 register contents

Bit position	Bit name	Function
31 to 12	Reserved	Reading returns the initial value. When written, write the initial value.
11 to 0	REFTHF[11:0]	Forcible Auto-Refresh threshold setting bits 080_H: 128 SDRAM operation clock cycles ... $1FF_H$: 511 SDRAM operation clock cycles All others: Setting prohibited

If you change the value of this register while auto-refresh is active (i.e. the ARFEN bit in the DBRFEN register is 1), the auto-refresh facility must be re-activated to reflect the new value in refreshing. To re-activate the auto-refresh facility, write 1 to the ARFEN bit in the DBRFEN register.

These bits set the timing for forcible refreshing regardless of bus requests.

The value represented by these bits affects the amount of jitter in the refresh interval and performance in access to memory. A smaller value means less jitter in the refresh interval but may reduce performance in access. For details on the amount of jitter in the refresh interval, see Section 16.3.1.33, DBRFCNF1 – Refresh configuration register 1.

The setting value should meet the following criteria:

REFTHF[11:0] ≥ (TCKEL[3:0] + TCKEH[7:0]) + REFTH0

(REFTH0 = max (TRDPR[3:0], CWL[3:0] + BL[1:0]/2 + TWR[3:0], TRAS[5:0], TRC[5:0] - TRP[4:0]) + RKRP * (installation rank number - 1) + TRPA[4:0] + 100)

16.3.1.33 DBRFCNF1 – Refresh configuration register 1

DBRFCNF1 is used to set the timing for refreshing of the SDRAM.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + $0E4_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	REFPMAX[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	REFINT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.43 DBRFCNF1 register contents

Bit position	Bit name	Function
31 to 16	REFPMAX[15:0]	Maximum post number of refresh commands setting bits 0002_H: 2 0003_H: 3 ... 0008_H: 8 All others: Setting prohibited
15 to 0	REFINT[15:0]	Average refresh interval setting bits 0080_H: 128 SDRAM operation clock cycles 0081_H: 129 SDRAM operation clock cycles ... $3FFF_H$: 16383 SDRAM operation clock cycles All others: Setting prohibited

If you change the value of this register while auto-refresh is active (i.e. the ARFEN bit in the DBRFEN register is 1), the auto-refresh facility must be re-activated to reflect the new value in refreshing. To re-activate the auto-refresh facility, write 1 to the ARFEN bit in the DBRFEN register.

REFPMAX[15:0] (Maximum post number of refresh commands setting)

These bits set the maximum number of refresh commands (post number) accumulated by auto-refresh. As long as the number of refresh commands that has been accumulated is smaller than REFPMAX, refresh commands are issued while there are no bus requests.

REFINT[15:0] (Average refresh interval setting)

These bits set the average interval for issuing of refresh commands. When the REFINTS bit of the DBRFCNF2 register is 0, the average interval (in cycles) is REFINT[15:0]. When the REFINTS bit of the DBRFCNF2 register is 1, on the other hand, the average interval (in cycles) is floor (REFINT[15:0]/2). This average interval is hereafter referred to as REFINT_E. Thus, REFINT_E = REFINT >> REFINTS (>>: logical right-shift operator).

Register set value and refresh issue timing

Setting example of refresh set register 1~2 and refresh issue timing at that time is explained.

a ± b indicates the range of above a-|b and below a+b.

For REFINT[15:0] set the discarded value in integer that shows an average refresh interval (t_{REFI} in Normal Operating Temperature Range) mentioned in data sheet of memory vendor by number of cycles. REFINTS sets 0 or 1 according to temperature.

In this case, considering n as right integer, time from refresh issue till refresh issue after that n refresh will be below n x REFINT_E + REFPMAX x REFINT_E cycle. However, it is assumed that there will be no writing to DBRFEN register within this period.

The setting value should meet the following criteria:

REFINT ≥ (TCALRZ + TCALZR + REFTHFx2) << REFINTS

16.3.1.34 DBRFCNF2 – Refresh configuration register 2

DBRFCNF2 is used to set the timing for refreshing of the SDRAM.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + $0E8_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	REFPMIN[3:0			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	REFINTS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.44 DBRFCNF2 register contents

Bit position	Bit name	Function
31 to 20	Reserved	Reading returns the initial value. When written, write the initial value.
19 to 16	REFPMIN[3:0]	Minimum post number of refresh commands setting 1_H: 1 All others: Setting prohibited
15 to 1	Reserved	Reading returns the initial value. When written, write the initial value.
0	REFINTS	Average refresh interval adjustment bits 0: Average refresh interval is REFINT[15:0] 1: Average refresh interval is 1/2 of REFINT[15:0]

If you change the value of this register while auto-refresh is active (i.e. the ARFEN bit in the DBRFEN register is 1), the auto-refresh facility must be re-activated to reflect the new value in refreshing. To re-activate the auto-refresh facility, write 1 to the ARFEN bit in the DBRFEN register.

REFPMIN[3:0] (minimum post number of refresh commands setting)

These bits are always read as 0. The write value should always be 0. If a value other than 0 is written, correct operation cannot be guaranteed.

REFINTS (Average refresh interval adjustment)

When this bit is 0, the average interval (in cycles) is the value set in the REFINT[15:0] bits. When this bit is 1, on the other hand, the average interval (in cycles) is floor (REFINT[15:0]/2).

16.3.1.35 DBRNK0 – ODT operation configuration register 0

This register is used to control ODT enable/disable.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 100_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RODT OUT0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WODT OUT0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.45 DBRNK0 register contents

Bit position	Bit name	Function
31 to 17	Reserved	Reading returns the initial value. When written, write the initial value.
16	RODTOUT0	The value of this bit determines the level of the SDRBODT output signal during read: 0: SDRBODT signal at low level during read 1: SDRBODT signal at high level during read
15 to 1	Reserved	Reading returns the initial value. When written, write the initial value.
0	WODTOUT0	The value of this bit determines the level of the SDRBODT output signal during write: 0: SDRBODT signal at low level during write 1: SDRBODT signal at high level during write

16.3.1.36 DBPDNCNF – Power down configuration register

This register is used to control the SDRAM power-down mode.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 180_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDWAIT[7:0]								0	0	0	PDDLL	0	0	PDMODE[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W

Table 16.46 DBPDNCNF register contents

Bit position	Bit name	Function
31 to 16	Reserved	Reading returns the initial value. When written, write the initial value.
15 to 8	PDWAIT[7:0]	SDRAM power down wait 10_H: 16 cycles 11_H: 17 cycles ... FF_H: 255 cycles All others: Setting prohibited PDWAIT[7:0] sets the number of cycles it takes to enter power-down mode after memory accesses no longer occur.
7 to 5	Reserved	Reading returns the initial value. When written, write the initial value.
4	PDDLL	SDRAM power down DLL control 0: DLL is turned off at a precharged power-down. DLL is turned on at an active power-down 1: DDR2: DLL is turned off at a power-down. PDDLL turns on or off the DLL of SDRAM when entering power-down mode.
3 to 2	Reserved	Reading returns the initial value. When written, write the initial value.
1 to 0	PDMODE[1:0]	SDRAM power down mode 00_B: Auto power-down mode is off 01_B: Auto power-down mode is on All others: Setting prohibited if PDMODE[1:0] = 01_B and there has been no memory access for a certain period, the CKE pin is set to the low level causing SDRAM to enter a power-down mode.

16.3.1.37 DBPDCNT0 – PHY control register 0

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 200_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	PHY BUS_ WIDTH	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 16.47 DBPDCNT0 register contents

Bit position	Bit name	Function
31	Reserved	Reading returns the initial value. When written, write the initial value.
30	PHYBUS_ WIDTH	Control of SDRAM interface bit width 0: 32 bit interface 1: 16 bit interface
29 to 0	Reserved	Reading returns the initial value. When written, write the initial value.

16.3.1.38 DBPDCNT1 – PHY control register 1

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 204_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DB_OFFSET[2:0]			0	DB_ENOFFSET[3:0]				0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	RNDTRIP_OFFSET1[3:0]				RNDTRIP_OFFSET0[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.48 DBPDCNT1 register contents

Bit position	Bit name	Function
31 to 29	DB_ OFFSET[2:0]	Control of Read DQ latency timing 010_B: 2 cycles other: prohibited
28	Reserved	Reading returns the initial value. When written, write the initial value.
27 to 24	DB_ ENOFFSET[3:0]	Control of Read DQS mask timing 0100_B: 4 cycles other: prohibited
23 to 8	Reserved	Reading returns the initial value. When written, write the initial value.
7 to 4	RNDTRIP_ OFFSET1[3:0]	Setting of round trip time of DQS3/2 0100_B: 4 cycles other: prohibited
3 to 0	RNDTRIP_ OFFSET0[3:0]	Setting of round trip time of DQS1/0 0100_B: 4 cycles other: prohibited

16.3.1.39 DBPDCNT3 – PHY control register 3

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + $20C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	DB_STBY	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 16.49 DBPDCNT3 register contents

Bit position	Bit name	Function
31 to 29	Reserved	Reading returns the initial value. When written, write the initial value.
28	DB_STBY	Control of PHY standby release 0: enter standby state 1: release standby state
27 to 0	Reserved	Reading returns the initial value. When written, write the initial value.

16.3.2 SDRAM bus control registers

16.3.2.1 DBBS0CNT1 – Bus control unit 0 control register 1

This register is used for bank address position.

CAUTION

Writing to this register is only permitted, when SDRAM access is disabled (DBACEN.ACCEN = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + 304_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	BKADP0[5:0]						0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 16.50 DBBS0CNT1 register contents

Bit position	Bit name	Function
31 to 8	Reserved	Reading returns the initial value. When written, write the initial value.
7 to 2	BKADP0[5:0]	Address position for bank of RANK0 Refer to Section 16.6.2, Relation between external SDRAM and logical addresses for details.
1 to 0	Reserved	Reading returns the initial value. When written, write the initial value.

16.3.3 Dynamic priority control registers

16.3.3.1 DBLGCNTi – Dynamic Priority Generator control register

This register allows to select the different operation modes of the Dynamic Priority Generator i.

CAUTION

This register must only be changed while the Dynamic Priority Generator is disabled (DBLGQONi = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + n x 100_H + 1000_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	UDF CLR	0	0	0	0	0	0	0	RGLMD
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	LDMD	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Table 16.51 DBLGCNTi register contents

Bit position	Bit name	Function
31 to 25	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
24	UDFCLR	QoS counter underflow bit clear 0: no function 1: clear bit of DBLGSTSi.UDF bit
23 to 17	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
16	RGLMD	Selection of condition for QoS counter operation state transition (Active state → Idle state) 0: State transition with number of requests 1: State transition with number of transferred bytes
15 to 9	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
8	LDMD	QoS counter reload mode selection 0: QoS counter reload without carry-over function 1: QoS counter reload with carry-over function
7 to 2	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
1	Bit 1	The default value “0” of this bit must not be changed.
0	Bit 0	The default value “0” of this bit must not be changed.

16.3.3.2 DBTMVAL0i – QoS counter initial value register

This register sets the initial value of the QoS counter of the Dynamic Priority Generator i.

CAUTION

This register must only be changed while the Dynamic Priority Generator is disabled (DBLGQONi = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + n x 100_H + 1004_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	CTSET[13:0]													
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.52 DBTMVAL0i register contents

Bit position	Bit name	Function
31 to 14	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
13 to 0	CTSET[13:0]	QoS counter initial value Allowed setting range: • CTSET[13:0] = 2001_H to $3FFF_H$ corresponds to QoS counter value 1 to 8191 • all other settings are prohibited

16.3.3.3 DBRQCTRi – QoS counter operation state transition condition register

This register defines the number of request or bytes, respectively, to change the QoS counter operation state from count to idle of the Dynamic Priority Generator i.

CAUTION

This register must only be changed while the Dynamic Priority Generator is disabled (DBLGQONi = 0).

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + n x 100_H + 1014_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	RQCTR[13:0]													
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.53 DBRQCTRi register contents

Bit position	Bit name	Function
31 to 14	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
13 to 0	RQCTR[13:0]	QoS counter operation state transition condition RQCTR[13:0] defines the number of • requests, if DBLGCNTi.RGLMD = 0 • bytes, if DBLGCNTi.RGLMD = 1 to change the QoS operation state from active to idle.

16.3.3.4 DBTHRES0i, DBTHRES1i, DBTHRES2i – Priority level threshold registers

This register defines the thresholds for changing the priority level of the Dynamic Priority Generator i.

CAUTION

This register must only be changed while the Dynamic Priority Generator is disabled (DBLGQONi = 0).

Access: This register can be accessed in 32-bit units.

Address: DBTHRES0i: <SDRBn_base> + n x 100_H + 1018_H
DBTHRES1i: <SDRBn_base> + n x 100_H + $101C_H$
DBTHRES2i: <SDRBn_base> + n x 100_H + 1020_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	CTTHRES[13:0]													
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16.54 DBTHRES0i, DBTHRES1i, DBTHRES2i register contents

Bit position	Bit name	Function
31 to 14	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
13 to 0	CTTHRES[13:0]	Priority level generation threshold CTTHRES[13:0] = 0000_H to $3FFF_H$ corresponds to threshold value -8192 to 8191

16.3.3.5 DBLGSTSi – Dynamic Priority Generator status register

This register holds various status information about the Dynamic Priority Generator i.

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + n x 100_H + 1024_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	UDF	0	0	0	0	0	0	0	REQ
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	CCT[13:0]													
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 16.55 DBLGSTSi register contents

Bit position	Bit name	Function
31 to 25	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
24	UDF	QoS counter underflow bit 0: no underflow 1: underflow UDF is set to 1 if the QoS counter reaches -8192 (that corresponds to CCT[13:0] = 0000_H). This bit can be cleared by setting DBLGCNTi.UDFCLR = 1.
23 to 17	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
16	REQ	Request detection bit 0: no request active on port i 1: request active on port i
15 to 14	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
13 to 0	CCT[13:0]	Present QoS counter value

16.3.3.6 DBLGQONi – Dynamic Priority Generator enable register

This register is used to enable or disable the Dynamic Priority Generator i.

Access: This register can be accessed in 32-bit units.

Address: <SDRBn_base> + n x 100_H + 1028_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	QON
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 16.56 DBLGQONi register contents

Bit position	Bit name	Function
31 to 1	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
0	QON	Dynamic Priority Generator enable/disable 0: Dynamic Priority Generator disabled 1: Dynamic Priority Generator enabled

16.4 DDR2-SDRAM Operation

16.4.1 Initial sequence

The following sequence enables SDRAM access after power-up.

NOTE

The sequence may vary with the specification of used memory.

1st setting of SDRB

(1) The type of memory is set through SDRAM type setting register DBKIND.

(2) Setting of the
- SDRAM configuration register DBCONF0
- SDRAM timing registers DBTR0, DBTR1 and DBTR3 to DBTR18
- SDRAM burst length register DBBL
- ODT operation configuration register 0 DBRNK0
- SDRB operation adjustment register 0 DBADJ0
- SDRB operation adjustment register 2 DBADJ2
- Dynamic Priority Generators related registers

is performed.

(3) Setting of the
- PHY control registers DBPDCNT0, DBPDCNT1, DBPDCNT3

(4) Release the PHY macro reset by MRSTC.SDRB0RES

DDR2-SDRAM setting

(1) The CKE terminal of all the ranks of the SDRAM is set to low by the manual SDRAM command register DBCMD with:
- OPC[5:0] = PDEn
- ARG[15:0] = 100 µs.

(2) Insert the wait time using the manual SDRAM command register DBCMD with:
- OPC[5:0] = Wait
- ARG[15:0] = 100 µs.

(3) The CKE terminal of the total rank of SDRAM is set to high by the manual SDRAM command register DBCMD with:
- OPC[5:0] = PDXt
- ARG[15:0] = 400 ns.

(4) PREA (pre-charge All) command is issued to RANK0 by manual SDRAM command register DBCMD with:

– OPC[5:0] = PreA
– ARG[15:0] = 0.

(5) The EMRS (MR2) command is issued to RANK0 by the manual SDRAM command register DBCMD.

(6) The EMRS (MR3) command is issued to RANK0 by the manual SDRAM command register DBCMD.

(7) The EMRS (MR1) command is issued to RANK0 by the manual SDRAM command register DBCMD. Additive Latency is set to 0, and DLL Enable is set to enable.

(8) The MRS (MR0) command is issued to RANK0 by the manual SDRAM command register DBCMD. In this case, the operation mode is set to Normal, DLL reset to reset, and burst type to sequential.
Perform following settings:
– burst length DBBL.BL[1:0]
– write recovery period DBTR10.TWR[3:0]
– power-down DLL control DBPDNCNF.PDDLL

(9) PREA (pre-charge All) command is issued to RANK0 by the manual SDRAM command register DBCMD with:
– OPC[5:0] = PreA
– ARG[15:0] = 0.

(10) REF (refresh) command is issued to RANK0 by the manual SDRAM command register DBCMD with:
– OPC[5:0] = Ref
– ARG[15:0] = 0.

(11) REF (refresh) command is issued to RANK0 by the manual SDRAM command register DBCMD with:
– OPC[5:0] = Ref
– ARG[15:0] = 0.

(12) The MRS (MR0) command is issued to RANK0 by the manual SDRAM command register DBCMD. In this case, perform the settings without the DLL reset.

(13) Insert the wait time using the manual SDRAM command register DBCMD with:
– OPC[5:0] = Wait
– ARG[15:0] = 200 cycles.

(14) The EMRS (MR1) command is issued to RANK0 by the manual SDRAM command register DBCMD. At this time, in addition, for the OCD calibration Program, set the value same as **(7)** to the OCD Calibration default.

(15) The EMRS (MR1) command is issued to RANK0 by the manual SDRAM command register DBCMD. At this time, in addition, for the OCD calibration Program, set a value same as **(7)** to the OCD Calibration mode exit.

2nd setting of SDRB

(1) If necessary, perform setting of bus control register 1 (DBBS0CNT1) and power-down configuration register (DBPDNCNF).

(2) Perform the setting of the refresh configuration registers DBRFCNF0 and DBRFCNF2).

(3) Enable SDRAM access by DBACEN.ACCEN = 1.

(4) Read the operation completion waiting register DBWAIT, and wait until a response is returned.

16.4.2 Power-down and -up sequence

If there is no need to access the SDRAM, it is possible to deactivate the internal clock of the SDRAM device and thereby reduce the power consumption of the device by changing to SDRAM power-down mode. Moreover, even in SDRAM power-down mode, the memory clock (SDRBCK/SDRBCKB) from the SDRB and the power to the SDRB and the SDRAM device must be supplied.

NOTE

The here mentioned SDRAM power-down mode must not be mixed up with the microcontroller's DEEPSTOP mode.

Power-down sequence

(1) Disable SDRAM access by DBACEN.ACCEN = 0.

(2) The Power Down Entry command is issued by the SDRAM manual command register DBCMD with:
- OPC[5:0] = PDEn
- ARG[15:0] = 0.

Power-up sequence

(1) The Power Down Exit command is issued by the SDRAM manual command register DBCMD with:
- OPC[5:0] = PDXt
- ARG[15:0] = 0.

(2) Enable SDRAM access by DBACEN.ACCEN = 1.

In order to hold the SDRAM data in the SDRAM power-down mode, it is necessary to issue a regular refresh command as during normal operation.

Any attempt to access any SDRAM data during power-down mode results in an error and the DBSVCR.INACTIVE error flag is set.

16.5 Dynamic Priority Generator and Arbitration

The Priority Generators and the Arbiter rule the access of the cross-connect masters to the DDR2-SDRAM Memory Controller.

The Arbiter finally grants access to the DDR2-SDRAM, based on various criteria, refer to Section 16.5.7, Arbiter operation for details.

The main criterion for the Arbiter is the priority level.
The Dynamic Priority Generators generate this priority level. It provides facilities to change the priority level dynamically in order to assign sufficient SDRAM bandwidth for the various master.

Each of the four ports has two Priority Generators, one handles the read, the other the write requests. Throughout this section the individual Priority Generators are identified by the index i (i = 0 to 7) with following assignment:

- i = 0: port 0 read (Port 0R Priority Generator)
- i = 1: port 1 read (Port 1R Priority Generator)
- i = 2: port 2 read (Port 2R Priority Generator)
- i = 3: port 3 read (Port 3R Priority Generator)
- i = 4: port 0 write (Port 0W Priority Generator)
- i = 5: port 1 write (Port 1W Priority Generator)
- i = 6: port 2 write (Port 2W Priority Generator)
- i = 7: port 3 write (Port 3W Priority Generator)

The following diagram shows the connections between the cross-connects, Priority Generators, Arbiter and DDR2-SDRAM Memory Controller.

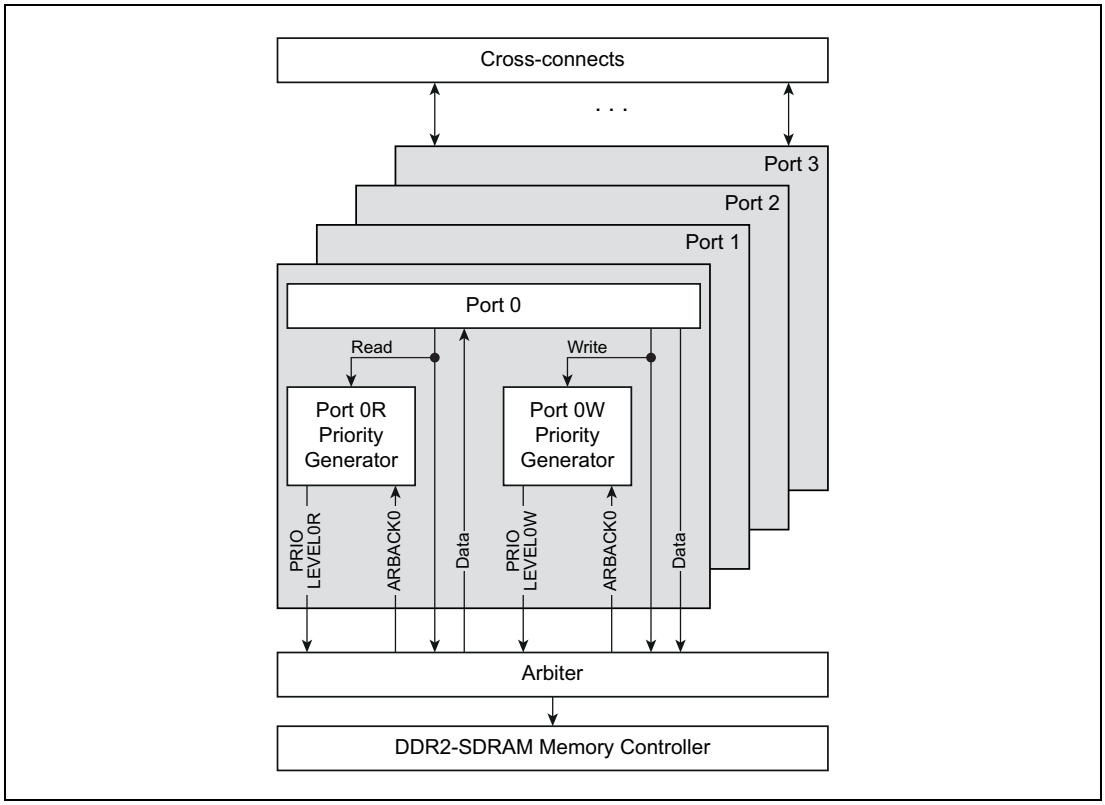


Figure 16.1 Priority Generators connection

The following diagram shows the main functional blocks of the Dynamic Priority Generators.

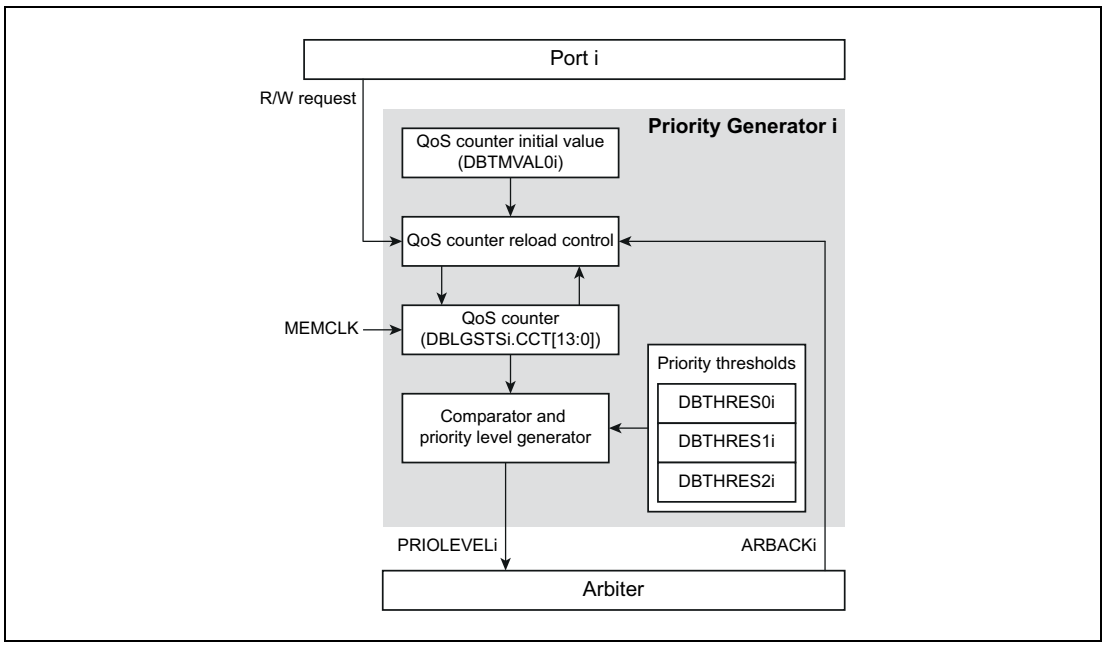


Figure 16.2 Dynamic Priority Generator block diagram

16.5.1 Dynamic priority control principle

The priority level of a port is dynamically changed over the time. The QoS counter DBLGSTi.CCT[13:0] is used as the time base. It starts counting down from a port specific start value with a data transfer request applied to its port.
At each point in time the counter value DBLGSTSi.CCT[13:0] is compared with three threshold values DBTHRES[2:0]i.CTTHRES[13:0]. The compare result defines the priority level PRIOLEVELi:

- DBLGSTSi.CCT[13:0] ≥ DBTHRES0i.CTTHRES[13:0]: PRIOLEVELi = 0
- DBTHRES0i.CTTHRES[13:0] > DBLGSTSi.CCT[13:0] ≥ DBTHRES1i.CTTHRES[13:0]: PRIOLEVELi = 1
- DBTHRES1i.CTTHRES[13:0] > DBLGSTSi.CCT[13:0] ≥ DBTHRES2i.CTTHRES[13:0]: PRIOLEVELi = 2
- DBTHRES2i.CTTHRES[13:0] > DBLGSTSi.CCT[13:0]: PRIOLEVELi = 3

The following diagram illustrates this process by using port 0 and 1 as examples.

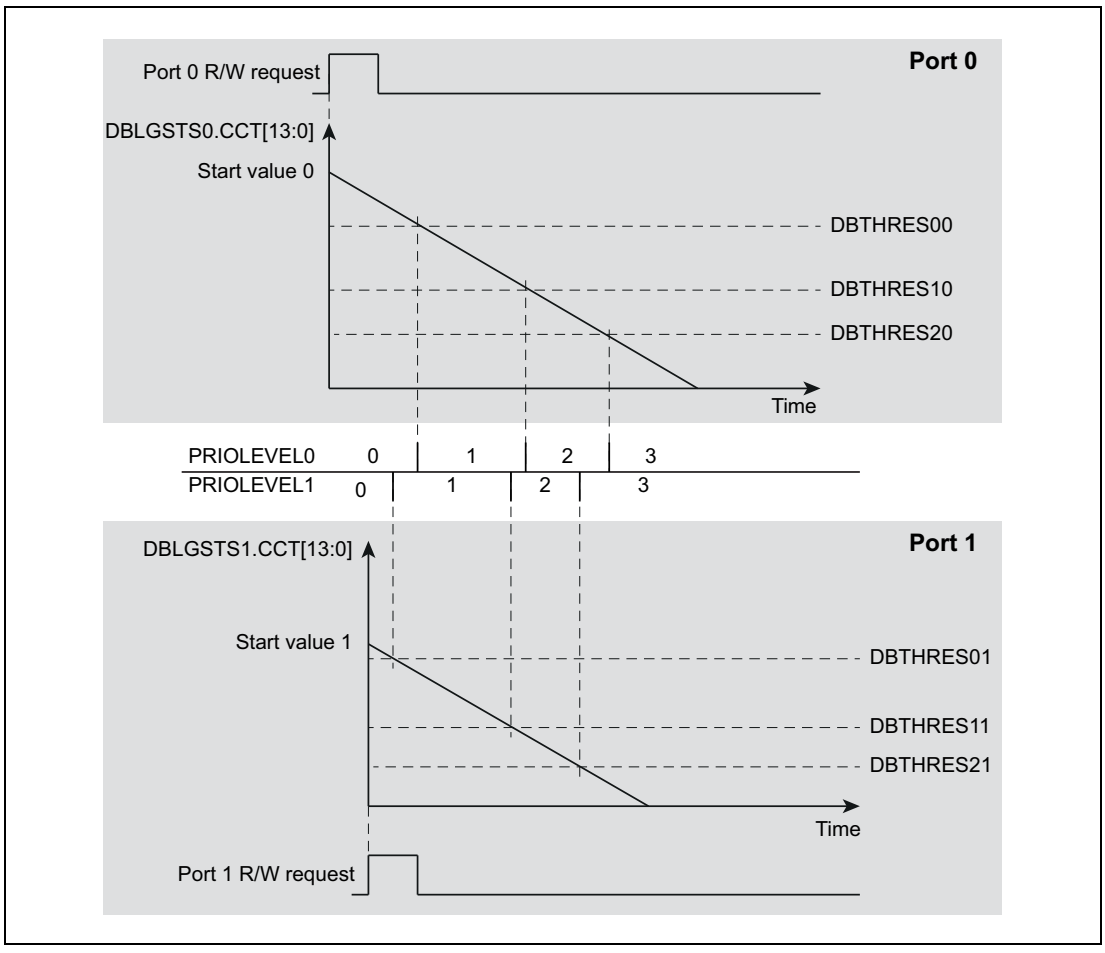


Figure 16.3 Dynamic priority control principle

The priority level of each port's R/W request at each point in time is determined by the

- start value of the QoS counter DBTMVAL0i.CTSET[13:0]
- priority threshold values DBTHRES[2:0]i.CTTHRES[13:0]

Thus adapting the start value and the thresholds means adjusting the priority as a function in time, and thus the waiting time for R/W request to be served.

Basically,

- higher thresholds mean higher priority, and thus lower waiting time for service
- higher start value means lower priority, and thus higher waiting time for service

16.5.2 Priority control by threshold adaptation

The threshold registers DBTHRES[2:0]i.CTTHRES[13:0] must only be changed by the application software, when the Priority Generator is disabled ((DBLGQONi.QON = 0).

The values of the thresholds CTTHRES[13:0] ranges
from -8192 (CTTHRES[13:0] = 0000_H) to +8191 (CTTHRES[13:0] = $3FFF_H$).

16.5.3 Enabling the Priority Generator

By default the Priority Generator is disabled (DBLGQONi.QON = 0) issues priority level PRIOLEVELi = 0.
The Priority Generator i is enabled by DBLGQONi.QON = 1.

CAUTION

The control bits RGLMD and LDMD in the dynamic priority level generation register DBLGCNTi must not be changed while the Priority Generator is enabled (DBLGQONi.QON = 1).

16.5.4 QoS counter states

The QoS counter has two operation states (provided it is enabled by DBLGQONi.QON = 1):

- Active state: In active state the QoS counter counts down.
- Idle state: In idle state the QoS counter holds the value 0 and does not count down.

The transition from idle state to active state is initiated upon a R/W request from the port.

The transition from active to idle state is performed when a certain number of requests have been served or a certain number of bytes have been transferred consecutively via one port.

The number of R/W requests or transferred bytes can be set for each Priority Generator i:

- DBLGCNTi.RGLMD = 0: QoS active-to-idle transition after DBRQCTRi.RQCTR[15:0] number of served requests
- DBLGCNTi.RGLMD = 1: QoS active-to-idle transition after DBRQCTRi.RQCTR[15:0] number of transferred bytes

16.5.5 Dynamic Priority Generator status information

The DBLGSTSi register holds various status information:

DBLGSTSi.UDF: QoS counter underflow

The count values of the QoS counter DBLGSTi.CCT[13:0] ranges
from -8192 (CCT[13:0] = 0000_H)
via 0 (CCT[13:0] = 2000_H)
to +8191 (CCT[13:0] = $3FFF_H$).

If a R/W request is received from the port, while the counter underflows (counter value ≤ 0, i.e. CCT[13:0] ≤ 2000_H), the QoS counter underflow bit DBLGSTSi.UDF is set.

Once set, this bit is not automatically cleared, but can be cleared by the application program by setting DBLGCNTi.UDFCLR = 1.

DBLGSTSi.REQ: R/W request status

REQ = 1 indicates, that a R/W request has been received from the port and pending or under process.

DBLGSTSi.CCT[13:0]: QoS counter value

CCT[13:0] holds the current value of the QoS counter.

16.5.6 Priority control by QoS counter start value adaptation

16.5.6.1 QoS counter reload modes

The reload of the start value of the QoS counter DBLGSTi.CCT[13:0] can operate in two different modes:

- DBLGCNTi.LDMD = 0: reload without carry-over function
- DBLGCNTi.LDMD =1: reload with carry-over function

QoS counter initial value

In both modes the QoS counter is set to an initial value at the first R/W request after enabling the Dynamic Priority Generator.

The initial value is defined by the CTSET[13:0] bits in the DBTMVAL0i register.
The initial value must not define a negative counter value, thus the allowed range is
from 1 (CTSET[13:0] = 2001_H)
to +8191 (CTSET[13:0] = $3FFF_H$).

16.5.6.2 QoS counter reload without carry-over function (DBLGCNTi.LDMD = 0)

In this mode the QoS counter is always reloaded with its
initial value DBTMVALi.CTSET[13:0]
upon R/W request.

In normal case, all data transfers via the DDR2-SDRAM Memory Controller, related to the R/W request, are completed within the QoS count time from it's initial value DBTMVALi.CTSET[13:0] down to 0. If these data transfers could not be completed within this time, the QoS counter underflow bit DBLGSTSi.UDF is set.

The following diagram explains the QoS counter behaviour under different circumstances.

The example in the diagram assumes, that each R/W request generates two data transfers (1, 2) via the DDR2-SDRAM Memory Controller and thus the Memory Controller arbiter replies two acknowledges (ARBACK) upon completion of each transfer.

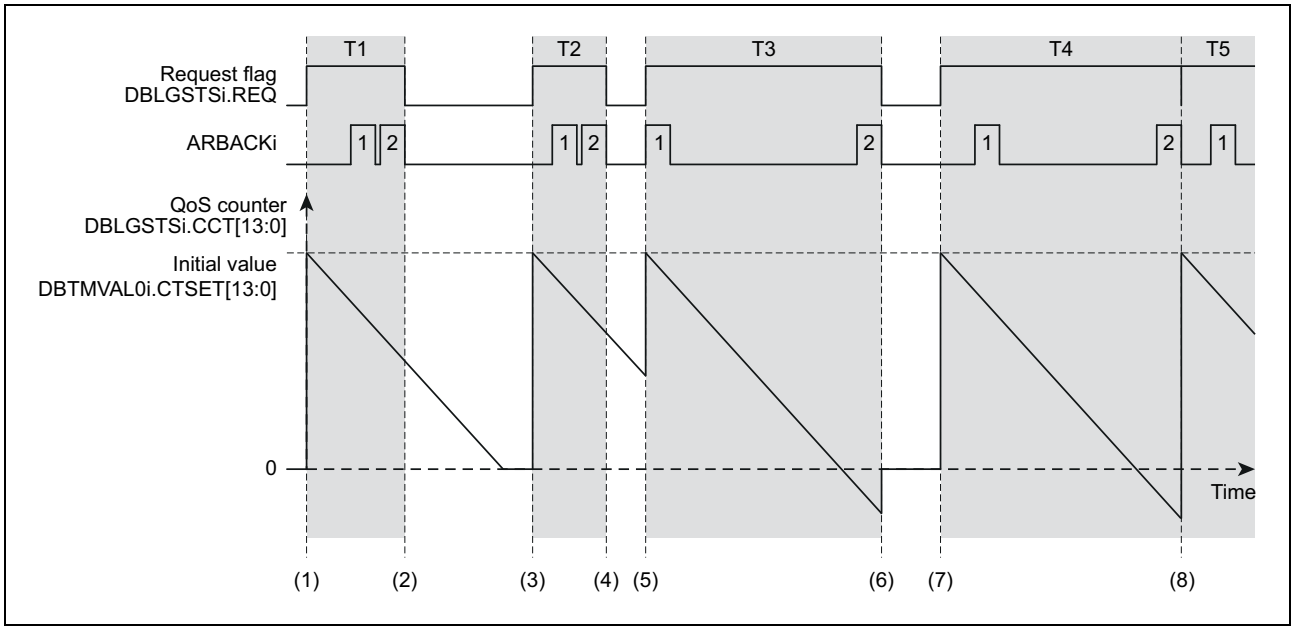


Figure 16.4 QoS counter reload without carry-over function examples

(1) The first R/W request starts the T1 transfer.
The QoS counter DBLGSTi.CCT[13:0] is loaded with the initial value DBTMVALi.CTSET[13:0] and starts counting down.

(2) After the arbiter has acknowledged both transfers (1, 2) T1 is completed.
The QoS counter counts down to 0 and remains there until the next R/W request, i.e. stays in idle state.

(3) The next R/W request initiates the T2 transfer.
The QoS counter DBLGSTi.CCT[13:0] is loaded with the initial value DBTMVALi.CTSET[13:0] and starts counting down.

(4) T2 is completed.

(5) Before the QoS counter reaches 0 a new T3 transfer is initiated.
The QoS counter is reloaded with the initial value and counts down.

(6) T3 is completed while the QoS counter underflows, i.e. the count value is < 0.

(7) The next R/W request initiates the T4 transfer.
The QoS counter DBLGSTi.CCT[13:0] is loaded with the initial value DBTMVALi.CTSET[13:0] and starts counting down.

(8) T4 is completed during QoS counter underflow and the next transfer T5 is started immediately.
The QoS counter is reloaded with the initial value and starts count-down.

16.5.6.3 QoS counter reload with carry-over function (DBLGCNTi.LDMD = 1)

In this mode the QoS counter is reloaded with the
DBTMVALi.CTSET[13:0] plus the current QoS counter value
upon R/W request.

Thus a R/W request, that occurs

- while the QoS counter has not reached 0 after the previous transfer, results in a higher QoS counter start value, i.e. lower priority and longer service waiting time, see change from T2 to T3 transfer in **Figure 16.5**
- during an underflow of the QoS counter results in a lower QoS counter start value, i.e. higher priority and shorter service waiting time, see change from T4 to T5 transfer in **Figure 16.5**

The following diagram explains the QoS counter behaviour under different circumstances.

The example in the diagram assumes, that each R/W request generates two data transfers (1, 2) via the DDR2-SDRAM Memory Controller and thus the Memory Controller arbiter replies two acknowledges (ARBACK) upon completion of each transfer.

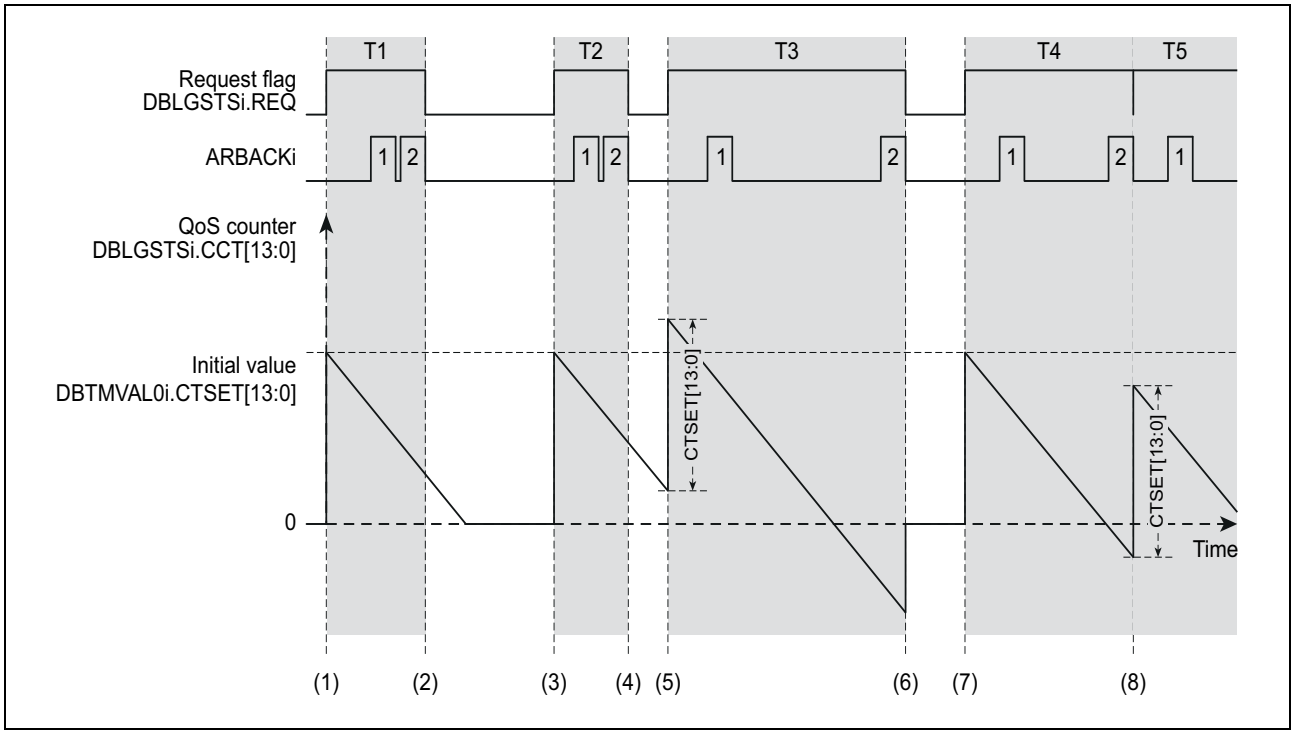


Figure 16.5 QoS counter reload with carry-over function examples

(1) The first R/W request starts the T1 transfer.
The QoS counter DBLGSTi.CCT[13:0] is loaded with the initial value DBTMVALi.CTSET[13:0] and starts counting down.

(2) After the arbiter has acknowledged both transfers (1, 2) T1 is completed.
The QoS counter continues to count down.

(3) The next R/W request initiates the T2 transfer.
The QoS counter is reloaded with the remaining counter value (here 0) plus the DBTMVALi.CTSET[13:0] value and starts count-down.

(4) T2 is completed.

(5) Before the QoS counter reaches 0 a new T3 transfer is initiated.
The QoS counter is reloaded with the remaining counter value plus the DBTMVALi.CTSET[13:0] value and starts count-down.

(6) T3 is completed while the QoS counter underflows, i.e. the count value is < 0.
The counter value is set to 0.

(7) The next R/W request initiates the T4 transfer.
The QoS counter is reloaded with the remaining counter value (here 0) plus the DBTMVALi.CTSET[13:0] value and starts count-down.

(8) T4 is completed during QoS counter underflow and the next transfer T5 is started immediately.
The QoS counter is reloaded with the remaining counter value (here a negative value) plus the DBTMVALi.CTSET[13:0] value and starts count-down.

16.5.7 Arbiter operation

The arbiter selects the port of the next transaction along the follow list of criteria:

(1) Serve the request with the largest priority level PRIOLEVELi.

(2) If multiple requests fulfil above criterion, serve the request with the same R/W direction as the previously served request.

(3) If multiple requests fulfil all above criteria, serve the request with access to the same SDRAM bank as the previously served request.

(4) If multiple requests fulfil all above criteria, serve the request with different access size than with the previous access. I.e. the request, that has changed its access size, gets higher priority and will be served.

(5) If multiple requests fulfil all above criteria, serve the request with the smaller port number i.

16.6 Appendix

16.6.1 Setting methods for SDRAM configuration setting register

16.6.1.1 DDR2-SDRAM (16 bit external bus)

Capability	Memory configuration	RANK	BANK [piece]	ROW [bit]	COL [bit]	AWRW	AWRK	AWBK	AWCL	DW
32MB	16M×16bit “256Mbit” (1 piece)	1	4	13	9	01101	0	10	1001	01
64MB	32M×16bit “512Mbit” (1 piece)	1	4	13	10	01101	0	10	1010	01
128MB	64M×16bit ”1Gbit” (1 piece)	1	8	13	10	01101	0	11	1010	01
256MB	128M×16bit ”2Gbit” (1 piece)	1	8	14	10	01110	0	11	1010	01

16.6.1.2 DDR2-SDRAM (32 bit external bus)

Capability	Memory configuration	RANK	BANK [piece]	ROW [bit]	COL [bit]	AWRW	AWRK	AWBK	AWCL	DW
64MB	16M×16bit “256Mbit” (2 pieces)	1	4	13	9	01101	0	10	1001	10
128MB	32M×16bit “512Mbit” (2 pieces)	1	4	13	10	01101	0	10	1010	10
256MB	64M×16bit ”1Gbit” (2 pieces)	1	8	13	10	01101	0	11	1010	10
512MB	128M×16bit ”2Gbit” (2 pieces)	1	8	14	10	01110	0	11	1010	10

16.6.2 Relation between external SDRAM and logical addresses

This section shows the relation between the external SDRAM pins and the logical address of the cross-connect transaction.

16.6.2.1 DDR2-SDRAM 16 bit external bus

Table 16.57 Relation between external and logical address (BKADP = 000000_B) with one 16-bit width SDRAM

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A11	A10	0	0	0	A24	A23	A22	A21	A20	A29	A28	A21	A16	A15	A14	A13	A12
	COL	0	A11	A10	0	0	0	0	0	AP	0	A9	A8	A7	A6	A5	A4	0	0	0
32M x16b	ROW	0	A12	A11	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	0	A12	A11	0	0	0	0	0	AP	A10	A9	A8	A7	A6	A5	A4	0	0	0
64M x16b	ROW	A13	A12	A11	0	0	0	A26	A25	A24	A23	A22	A21	A20	A29	A28	A21	A16	A15	A14
	COL	A13	A12	A11	0	0	0	0	0	AP	A10	A9	A8	A7	A6	A5	A4	0	0	0
128M x16b	ROW	A13	A12	A11	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A29	A28	A21	A16	A15	A14
	COL	A13	A12	A11	0	0	0	0	0	AP	A10	A9	A8	A7	A6	A5	A4	0	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

Table 16.58 Relation between external and logical address (BKADP = 000111_B) with one 16-bit width SDRAM

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A8	A7	0	0	0	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
	COL	0	A8	A7	0	0	0	0	0	AP	0	A11	A10	A9	A6	A5	A4	0	0	0
32M x16b	ROW	0	A8	A7	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	0	A8	A7	0	0	0	0	0	AP	A12	A11	A10	A9	A6	A5	A4	0	0	0
64M x16b	ROW	A9	A8	A7	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	A9	A8	A7	0	0	0	0	0	AP	A13	A12	A11	A10	A6	A5	A4	0	0	0
128M x16b	ROW	A9	A8	A7	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	A9	A8	A7	0	0	0	0	0	AP	A13	A12	A11	A10	A6	A5	A4	0	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

Table 16.59 Relation between external and logical address (BKADP = 001000_B) with one 16-bit width SDRAM

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A9	A8	0	0	0	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
	COL	0	A9	A8	0	0	0	0	0	AP	0	A11	A10	A7	A6	A5	A4	0	0	0
32M x16b	ROW	0	A9	A8	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	0	A9	A8	0	0	0	0	0	AP	A12	A11	A10	A7	A6	A5	A4	0	0	0
64M x16b	ROW	A10	A9	A8	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	A10	A9	A8	0	0	0	0	0	AP	A13	A12	A11	A7	A6	A5	A4	0	0	0
128M x16b	ROW	A10	A9	A8	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	A10	A9	A8	0	0	0	0	0	AP	A13	A12	A11	A7	A6	A5	A4	0	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

Table 16.60 Relation between external and logical address (BKADP = 001001_B) with one 16-bit width SDRAM

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A10	A9	0	0	0	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
	COL	0	A10	A9	0	0	0	0	0	AP	0	A11	A8	A7	A6	A5	A4	0	0	0
32M x16b	ROW	0	A10	A9	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	0	A10	A9	0	0	0	0	0	AP	A12	A11	A8	A7	A6	A5	A4	0	0	0
64M x16b	ROW	A11	A10	A9	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	A11	A10	A9	0	0	0	0	0	AP	A13	A12	A8	A7	A6	A5	A4	0	0	0
128M x16b	ROW	A11	A10	A9	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	A11	A10	A9	0	0	0	0	0	AP	A13	A12	A8	A7	A6	A5	A4	0	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

Table 16.61 Relation between external and logical address (BKADP = 001010_B) with one 16-bit width SDRAM

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A11	A10	0	0	0	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
	COL	0	A11	A10	0	0	0	0	0	AP	0	A9	A8	A7	A6	A5	A4	0	0	0
32M x16b	ROW	0	A11	A10	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	0	A11	A10	0	0	0	0	0	AP	A12	A9	A8	A7	A6	A5	A4	0	0	0
64M x16b	ROW	A12	A11	A10	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	A12	A11	A10	0	0	0	0	0	AP	A13	A9	A8	A7	A6	A5	A4	0	0	0
128M x16b	ROW	A12	A11	A10	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	A12	A11	A10	0	0	0	0	0	AP	A13	A9	A8	A7	A6	A5	A4	0	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

Table 16.62 Relation between external and logical address (BKADP = 001011_B) with one 16-bit width SDRAM

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A12	A11	0	0	0	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A10
	COL	0	A12	A11	0	0	0	0	0	AP	0	A9	A8	A7	A6	A5	A4	0	0	0
32M x16b	ROW	0	A12	A11	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	0	A12	A11	0	0	0	0	0	AP	A10	A9	A8	A7	A6	A5	A4	0	0	0
64M x16b	ROW	A13	A12	A11	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	A13	A12	A11	0	0	0	0	0	AP	A10	A9	A8	A7	A6	A5	A4	0	0	0
128M x16b	ROW	A13	A12	A11	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	A13	A12	A11	0	0	0	0	0	AP	A10	A9	A8	A7	A6	A5	A4	0	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

Table 16.63 Relation between external and logical address (BKADP = 001100_B) with one 16-bit width SDRAM

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A13	A12	0	0	0	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A11	A10
	COL	0	A13	A12	0	0	0	0	0	AP	0	A9	A8	A7	A6	A5	A4	0	0	0
32M x16b	ROW	0	A13	A12	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A11
	COL	0	A13	A12	0	0	0	0	0	AP	A10	A9	A8	A7	A6	A5	A4	0	0	0
64M x16b	ROW	A14	A13	A12	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A11
	COL	A14	A13	A12	0	0	0	0	0	AP	A10	A9	A8	A7	A6	A5	A4	0	0	0
128M x16b	ROW	A14	A13	A12	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A11
	COL	A14	A13	A12	0	0	0	0	0	AP	A10	A9	A8	A7	A6	A5	A4	0	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

16.6.2.2 DDR2-SDRAM (32 bit external bus)

Table 16.64 Relation between external and logical address (BKADP = 000000_B) with two 16-bit width SDRAMs

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A12	A11	0	0	0	A25	A24	A23	A22	A21	A20	A29	A28	A21	A16	A15	A14	A13
	COL	0	A12	A11	0	0	0	0	0	AP	0	A10	A9	A8	A7	A6	A5	A4	0	0
32M x16b	ROW	0	A13	A12	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	0	A13	A12	0	0	0	0	0	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
64M x16b	ROW	A14	A13	A12	0	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A14	A13	A12	0	0	0	0	0	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
128M x16b	ROW	A14	A13	A12	0	0	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A14	A13	A12	0	0	0	0	0	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

Table 16.65 Relation between external and logical address (BKADP = 000111_B) with two 16-bit width SDRAMs

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A8	A7	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	0	A8	A7	0	0	0	0	0	AP	0	A12	A11	A10	A9	A6	A5	A4	0	0
32M x16b	ROW	0	A8	A7	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	0	A8	A7	0	0	0	0	0	AP	A13	A12	A11	A10	A9	A6	A5	A4	0	0
64M x16b	ROW	A9	A8	A7	0	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A9	A8	A7	0	0	0	0	0	AP	A14	A13	A12	A11	A10	A6	A5	A4	0	0
128M x16b	ROW	A9	A8	A7	0	0	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A9	A8	A7	0	0	0	0	0	AP	A14	A13	A12	A11	A10	A6	A5	A4	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. 2 AP indicates the auto pre-charge option.

Table 16.66 Relation between external and logical address (BKADP = 001000_B) with two 16-bit width SDRAMs

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A9	A8	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	0	A9	A8	0	0	0	0	0	AP	0	A12	A11	A10	A7	A6	A5	A4	0	0
32M x16b	ROW	0	A9	A8	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	0	A9	A8	0	0	0	0	0	AP	A13	A12	A11	A10	A7	A6	A5	A4	0	0
64M x16b	ROW	A10	A9	A8	0	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A10	A9	A8	0	0	0	0	0	AP	A14	A13	A12	A11	A7	A6	A5	A4	0	0
128M x16b	ROW	A10	A9	A8	0	0	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A10	A9	A8	0	0	0	0	0	AP	A14	A13	A12	A11	A7	A6	A5	A4	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

Table 16.67 Relation between external and logical address (BKADP = 001001_B) with two 16-bit width SDRAMs

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A10	A9	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	0	A10	A9	0	0	0	0	0	AP	0	A12	A11	A8	A7	A6	A5	A4	0	0
32M x16b	ROW	0	A10	A9	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	0	A10	A9	0	0	0	0	0	AP	A13	A12	A11	A8	A7	A6	A5	A4	0	0
64M x16b	ROW	A11	A10	A9	0	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A11	A10	A9	0	0	0	0	0	AP	A14	A13	A12	A8	A7	A6	A5	A4	0	0
128M x16b	ROW	A11	A10	A9	0	0	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A11	A10	A9	0	0	0	0	0	AP	A14	A13	A12	A8	A7	A6	A5	A4	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

Table 16.68 Relation between external and logical address (BKADP = 001010_B) with two 16-bit width SDRAMs

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A11	A10	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	0	A11	A10	0	0	0	0	0	AP	0	A12	A9	A8	A7	A6	A5	A4	0	0
32M x16b	ROW	0	A11	A10	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	0	A11	A10	0	0	0	0	0	AP	A13	A12	A9	A8	A7	A6	A5	A4	0	0
64M x16b	ROW	A12	A11	A10	0	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A12	A11	A10	0	0	0	0	0	AP	A14	A13	A9	A8	A7	A6	A5	A4	0	0
128M x16b	ROW	A12	A11	A10	0	0	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A12	A11	A10	0	0	0	0	0	AP	A14	A13	A9	A8	A7	A6	A5	A4	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

Table 16.69 Relation between external and logical address (BKADP = 001011_B) with two 16-bit width SDRAMs

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A12	A11	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13
	COL	0	A12	A11	0	0	0	0	0	AP	0	A10	A9	A8	A7	A6	A5	A4	0	0
32M x16b	ROW	0	A12	A11	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	0	A12	A11	0	0	0	0	0	AP	A13	A10	A9	A8	A7	A6	A5	A4	0	0
64M x16b	ROW	A13	A12	A11	0	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A13	A12	A11	0	0	0	0	0	AP	A14	A10	A9	A8	A7	A6	A5	A4	0	0
128M x16b	ROW	A13	A12	A11	0	0	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A13	A12	A11	0	0	0	0	0	AP	A14	A10	A9	A8	A7	A6	A5	A4	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.
Note 2. AP indicates the auto pre-charge option.

Table 16.70 Relation between external and logical address (BKADP = 001100_B) with two 16-bit width SDRAMs

Type		BA2	BA1	BA0	MA15	MA14	MA13	MA12	MA11	MA10	MA9	MA8	MA7	MA6	MA5	MA4	MA3	MA2	MA1	MA0
16M x16b	ROW	0	A13	A12	0	0	0	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A11
	COL	0	A13	A12	0	0	0	0	0	AP	0	A10	A9	A8	A7	A6	A5	A4	0	0
32M x16b	ROW	0	A13	A12	0	0	0	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14
	COL	0	A13	A12	0	0	0	0	0	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
64M x16b	ROW	A14	A13	A12	0	0	0	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A14	A13	A12	0	0	0	0	0	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0
128M x16b	ROW	A14	A13	A12	0	0	A28	A27	A26	A25	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15
	COL	A14	A13	A12	0	0	0	0	0	AP	A11	A10	A9	A8	A7	A6	A5	A4	0	0

Note 1. A31-A0 are logical address bits of byte unit, and A31 indicates the MSB side and A0 indicates the LSB side.

Note 2. AP indicates the auto pre-charge option.

Section 17 Serial Flash Memory Interface A (SFMA)

This section contains a generic description of the Serial Flash Memory Interface A.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

17.1 Overview of RH850/D1L/D1M Serial Flash Memory Interface A

17.1.1 Units

This microcontroller has the following number of units of the Serial Flash Memory Interface A.

Table 17.1 Units

Serial Flash Memory Interface A (SFMA)	D1L1 D1L2(H)	D1M1(H)	D1M1-V2	D1M1A	D1M2(H)
Units	1	1	2	3	1
Names	SFMA0	SFMA0	SFMA0, SFMA1	SFMA0, SFMA1, SFMA2	SFMA0

Units index n

Throughout this section, the individual units of a Serial Flash Memory Interface A is identified by the index "n" (n = 0), for example SFMAnCMNCR for the SFMAn common control register.

17.1.2 Register addresses

All Serial Flash Memory Interfaces A register addresses are given as address offsets from the individual base addresses <SFMAn_base>.
The <SFMAn_base> addresses of each SFMAn are listed in the following table:

Table 17.2 Register base addresses <SFMAn_base>

SFMAn unit	<SFMAn_base> address
SFMA0	F2FF 0000_H
SFMA1*1	F2FF 1000_H
SFMA2*1	

Note 1. SFMA1 and SFMA2 cannot be used concurrently. One of them must be selected by XCRAMCFG0.SFMASEL. Refer to Section 14.4, Bus Switch for external memory interfaces (D1M1-V2, D1M1A only) for details.

17.1.3 Clock supply

All Serial Flash Memory Interfaces A provide two clock inputs.

Table 17.3 Clock supply

SFMAn unit	SFMAn clock	Connected to
SFMA0 SFMA2	BΦ	Clock Controller CKSC_ISFMAD_CTL clock divider output
	PHCLK	Clock Controller CKSC_ISFMAS_CTL clock selector output
SFMA1	BΦ	Clock Controller CKSC_ISFMAD_CTL clock divider output divided by 2
	PHCLK	Clock Controller CKSC_ISFMAS_CTL clock selector output divided by 2

17.1.4 Reset sources

The Serial Flash Memory Interfaces A and their registers are initialized by the following reset signal:

Table 17.4 Reset sources

SFMAn unit	Reset signal
SFMAn	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

17.1.5 I/O signals

The following table shows the I/O signals of the Serial Flash Memory Interfaces A.

Table 17.5 SFMAn I/O signals (1/2)

SFMAn signals	Function	Connected to
SFMA0		
SPBCLK	Clock output	Port SFMA0CLK
SPBSSL	Slave select signal output	Port SFMA0SSL
SPBMO0/ SPBIO00	Port 0 master transmit data/data 0	Port SFMA0I00
SPBMI0/ SPBIO10	Port 0 master input data/data 1	Port SFMA0I10
SPBIO20	Port 0 data 2	Port SFMA0I20
SPBIO30	Port 0 data 3	Port SFMA0I30
SPBMO1/ SPBIO01	Port 1 master transmit data/data 0	Port SFMA0I01
SPBMI1/ SPBIO11	Port 1 master input data/data 1	Port SFMA0I11
SPBIO21	Port 1 data 2	Port SFMA0I21
SPBIO31	Port 1 data 3	Port SFMA0I31
SFMA1		
SPBCLK	Clock output	Port SFMA1CLK
SPBSSL	Slave select signal output	Port SFMA1SSL
SPBMO0/ SPBIO00	Port 0 master transmit data/data 0	Port SFMA1I00
SPBMI0/ SPBIO10	Port 0 master input data/data 1	Port SFMA1I10
SPBIO20	Port 0 data 2	Port SFMA1I20
SPBIO30	Port 0 data 3	Port SFMA1I30
SPBMO1/ SPBIO01	Port 1 master transmit data/data 0	Port SFMA1I01
SPBMI1/ SPBIO11	Port 1 master input data/data 1	Port SFMA1I11
SPBIO21	Port 1 data 2	Port SFMA1I21
SPBIO31	Port 1 data 3	Port SFMA1I31
SFMA2		
SPBCLK	Clock output	Port SFMA2CLK
SPBSSL	Slave select signal output	Port SFMA2SSL
SPBMO0/ SPBIO00	Port 0 master transmit data/data 0	Port SFMA2I00

Table 17.5 SFMAn I/O signals (2/2)

SFMAn signals	Function	Connected to
SPBMI0/ SPBIO10	Port 0 master input data/data 1	Port SFMA2I10
SPBIO20	Port 0 data 2	Port SFMA2I20
SPBIO30	Port 0 data 3	Port SFMA2I30

17.2 Features

The Serial Flash Memory Interface outputs control signals to the serial flash memory connected to the SPI multi I/O bus space, thus enabling direct connection of the serial flash memory.

This module allows the connected serial flash memory to be accessed by directly reading the SPI multi I/O bus space, or using SPI operating mode to transmit and receive data.

- Serial Flash Memory Interface
  One serial flash memory device can be connected to each port
  A data bus size of 1 bit, 2 bits, or 4 bits can be selected for one serial flash memory device.
  Serial flash memory for DDR transfer can be directly connected.
- External Address Space Read Mode
  A maximum of 8-Gbyte address space is supported (when two serial flash memories are connected)
  The SPBSSL pin can be automatically controlled through access address monitoring
  Efficient data reception due to built-in read cache (64-bit line × 16 entries)
- SPI Operating Mode
  Desired read/write access to serial flash memory possible
- Bit rate
  SPBCLK is generated by frequency division of Bϕ by internal baud rate generator
  SPBCLK frequency division ratio can be set from 2 to 4080
- SPBSSL Pin Control
  Delay from SPBSSL signal assertion to SPBCLK operation (clock delay) can be set
  Range: 1 to 8 SPBCLK cycles (set in SPBCLK-cycle units)
  Delay from SPBCLK stop to SPBSSL output negation (SPBSSL negation delay) can be set
  Range: 1.5 to 8.5 SPBCLK cycles (set in SPBCLK-cycle units)
  SPBSSL output assertion wait before next access (next access delay) can be set
  Range: 1 to 8 SPBCLK cycles (set in SPBCLK-cycle units)
  SPBSSL polarity can be changed

17.3 Block Diagram

The following figure shows a block diagram of this module for one channel.

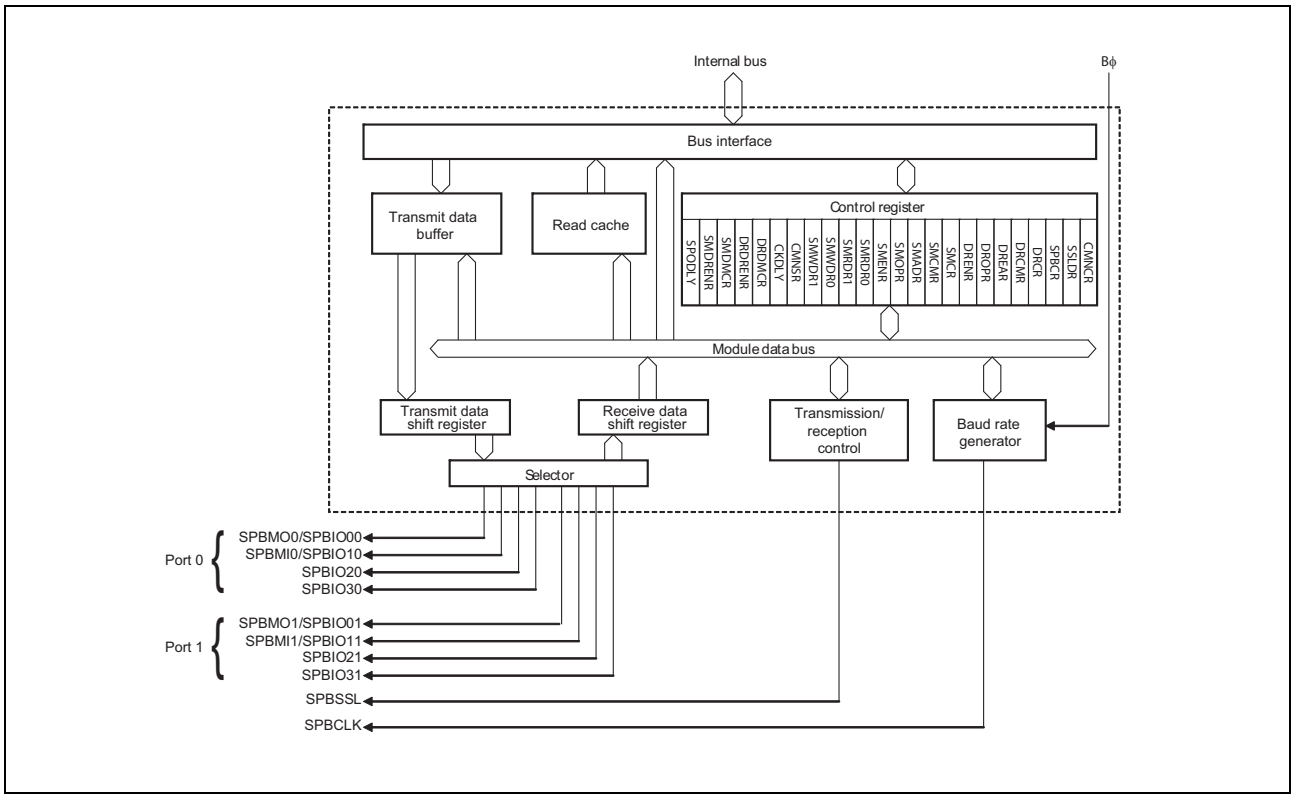


Figure 17.1 Block Diagram

17.4 Register Descriptions

The following table shows the register configuration.

Table 17.6 Register Configuration

Module Name	Register Name	Symbol	Initial Value	R/W	Address	Access Size
SFMAn	Common control register	CMNCR	H'01AA4000	R/W	<SFMAn_base> + 00_H	32
SFMAn	SSL delay register	SSLDR	H'00070707	R/W	<SFMAn_base> + 04_H	32
SFMAn	Bit rate register	SPBCR	H'00000003	R/W	<SFMAn_base> + 08_H	32
SFMAn	Data read control register	DRCR	H'00000000	R/W	<SFMAn_base> + $0C_H$	32
SFMAn	Data read command setting register	DRCMR	H'00030000	R/W	<SFMAn_base> + 10_H	32
SFMAn	Data read extended address setting register	DREAR	H'00000000	R/W	<SFMAn_base> + 14_H	32
SFMAn	Data read option setting register	DROPR	H'00000000	R/W	<SFMAn_base> + 18_H	32
SFMAn	Data read enable setting register	DRENR	H'00004700	R/W	<SFMAn_base> + $1C_H$	32
SFMAn	SPI mode control register	SMCR	H'00000000	R/W	<SFMAn_base> + 20_H	32
SFMAn	SPI mode command setting register	SMCMR	H'00000000	R/W	<SFMAn_base> + 24_H	32
SFMAn	SPI mode address setting register	SMADR	H'00000000	R/W	<SFMAn_base> + 28_H	32
SFMAn	SPI mode option setting register	SMOPR	H'00000000	R/W	<SFMAn_base> + $2C_H$	32
SFMAn	SPI mode enable setting register	SMENR	H'00004000	R/W	<SFMAn_base> + 30_H	32
SFMAn	SPI mode read data register 0	SMRDR0	Undefined	R	<SFMAn_base> + 38_H	8, 16, 32
SFMAn	SPI mode read data register 1	SMRDR1	Undefined	R	<SFMAn_base> + $3C_H$	8, 16, 32
SFMAn	SPI mode write data register 0	SMWDR0	H'00000000	R/W	<SFMAn_base> + 40_H	32
SFMAn	SPI mode write data register 1	SMWDR1	H'00000000	R/W	<SFMAn_base> + 44_H	32
SFMAn	Common status register	CMNSR	H'00000001	R	<SFMAn_base> + 48_H	32
SFMAn	Clock phase adjust register	CKDLY	H'00000000	R/W	<SFMAn_base> + 50_H	32
SFMAn	Data read dummy cycle setting register	DRDMCR	H'00000000	R/W	<SFMAn_base> + 58_H	32
SFMAn	Data read DDR enable register	DRDRENR	H'00000000	R/W	<SFMAn_base> + $5C_H$	32
SFMAn	SPI mode dummy cycle setting register	SMDMCR	H'00000000	R/W	<SFMAn_base> + 60_H	32
SFMAn	SPI mode DDR enable register	SMDRENR	H'00000000	R/W	<SFMAn_base> + 64_H	32
SFMAn	Output data delay register	SPODLY	H'00001111	R/W	<SFMAn_base> + 68_H	32

<SFMAn_base>

The base addresses <SFMAn_base> of the SFMAn is defined in the first subsection of this section under the key word “Register addresses”.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

17.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the SPI multi I/O bus controller. The settings of this register are reflected both in external address space read mode and SPI operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD	-	-	-	-	-	-	SFDE	MOIIO3[1:0]		MOIIO2[1:0]		MOIIO1[1:0]		MOIIO0[1:0]	
Initial value:	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0
R/W:	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IO3FV[1:0]		IO2FV[1:0]		-	-	IO0FV[1:0]		-	CPHAT	CPHAR	SSLP	CPOL	-	BSZ[1:0]	
Initial value:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	MD	0	R/W	Operating Mode Switch Switches the operating modes. 0: External address space read mode 1: SPI operating mode
30 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	SFDE	1	R/W	Data Swap Setting for Serial Flash Memory Specifies whether or not swapping of data in serial flash memory is performed. 0: Swapping is not performed. 1: Swapping is performed in 8-bit units. For details, see Section 17.5.4, Data Alignment.
23, 22	MOIIO3[1:0]	10	R/W	SPBSSL Output Idle Value Fix SPBIO30, SPBIO31 Fixes output values of SPBIO30 and SPBIO31 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z
21, 20	MOIIO2[1:0]	10	R/W	SPBSSL Output Idle Value Fix SPBIO20, SPBIO21 Fixes output values of SPBIO20 and SPBIO21 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z
19, 18	MOIIO1[1:0]	10	R/W	SPBSSL Output Idle Value Fix SPBIO10, SPBIO11 Fixes output values of SPBIO10 and SPBIO11 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z

Bit	Bit Name	Initial Value	R/W	Description
17, 16	MOIIO0[1:0]	10	R/W	SPBSSL Output Idle Value Fix SPBIO00, SPBIO01 Fixes output values of SPBIO00 and SPBIO01 in SPBSSL negation period. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z
15, 14	IO3FV[1:0]	01	R/W	SPBIO30, SPBIO31 Fixed Value for 1-bit/2-bit Size Fixes the output value of SPBIO30 and SPBIO31 pins for 1-bit/2-bit size. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z
13, 12	IO2FV[1:0]	00	R/W	SPBIO20, SPBIO21 Fixed Value for 1-bit/2-bit Size Fixes the output value of SPBIO20 and SPBIO21 pins for 1-bit/2-bit size. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	IO0FV[1:0]	00	R/W	SPBIO00, SPBIO01 Fixed Value for 1-bit Size Input Fixes the output value of SPBIO00 and SPBIO01 pins for 1-bit size input. 00: Output value 0 01: Output value 1 10: Output value is the value of the immediately previous bit (or the pin is Hi-Z, if Hi-Z was the state in the immediately previous bit period). 11: Output value Hi-Z
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	CPHAT	0	R/W	Output Shift Sets the SPBCLK edge of the output data. CPHAT and CPHAR should be set according to the table in the description of CPHAR. 0: Data transmission at even edge during SDR transfer Data transmission starts at even edge during DDR transfer. 1: Data transmission at odd edge during SDR transfer Data transmission starts at odd edge during DDR transfer.
5	CPHAR	0	R/W	Input Latch Sets the SPBCLK edge of the reception data. CPHAT and CPHAR should be set according to the following table. 0: Data reception at odd edge during SDR transfer Data reception starts at odd edge during DDR transfer. 1: Data reception at even edge during SDR transfer Data reception starts at even edge during DDR transfer

CPHAT	CPHAR	
0	0	Setting enabled
0	1	Setting enabled during SDR transfer
1	0	Setting prohibited
1	1	Setting enabled

Note: To set DDR transfer, set both the CPHAT and CPHAR bits to 0 or 1.

Bit	Bit Name	Initial Value	R/W	Description
4	SSLP	0	R/W	SPBSSL Signal Polarity Sets the polarity of SPBSSL signal. 0: Active low SPBSSL signal 1: Active high SPBSSL signal
3	CPOL	0	R/W	SPBSSL Negation Period SPBCLK Output Direction Sets the SPBCLK output direction during SPBSSL negation period. 0: SPBCLK output is 0 during SPBSSL negation period. 1: SPBCLK output is 1 during SPBSSL negation period.
2	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
1, 0	BSZ[1:0]	00	R/W	Data Bus Size Specifies the number of serial flash memories to be connected. 00: 1 memory 01: 2 memories 1X: Setting prohibited Note: After changing (the value of) this bit, all the entries in the read cache must be cleared by setting the RCF bit in DRCR to 1.

17.4.2 SSL Delay Register (SSLDR)

SSLDR is a 32-bit register that adjusts the timing between the SPBSSL signal and the SPBCLK signal.

The settings of this register are reflected both in external address space read mode and SPI operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	SPNDL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	SLNDL[2:0]			-	-	-	-	-	SCKDL[2:0]		
Initial value:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18 to 16	SPNDL[2:0]	111	R/W	Next Access Delay Sets the period from transfer end to next transfer start (next access). 000: 1 SPBCLK cycle 001: 2 SPBCLK cycles 010: 3 SPBCLK cycles 011: 4 SPBCLK cycles 100: 5 SPBCLK cycles 101: 6 SPBCLK cycles 110: 7 SPBCLK cycles 111: 8 SPBCLK cycles
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 8	SLNDL[2:0]	111	R/W	SPBSSL Negation Delay Sets the period from the time the last SPBCLK edge is sent of a transfer to SPBSSL pin negation (SPBSSL negation delay). 000: 1.5 SPBCLK cycles 001: 2.5 SPBCLK cycles 010: 3.5 SPBCLK cycles 011: 4.5 SPBCLK cycles 100: 5.5 SPBCLK cycles 101: 6.5 SPBCLK cycles 110: 7.5 SPBCLK cycles 111: 8.5 SPBCLK cycles
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	SCKDL[2:0]	111	R/W	Clock Delay Sets the period from SPBSSL pin assertion to SPBCLK oscillation (clock delay). 000: 1 SPBCLK cycle 001: 2 SPBCLK cycles 010: 3 SPBCLK cycles 011: 4 SPBCLK cycles 100: 5 SPBCLK cycles 101: 6 SPBCLK cycles 110: 7 SPBCLK cycles 111: 8 SPBCLK cycles

17.4.3 Bit Rate Register (SPBCR)

SPBCR is a 32-bit register that sets the bit rate.

The settings of this register are reflected both in external address space read mode and SPI operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPBR[7:0]								-	-	-	-	-	-	BRDV[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 8	SPBR[7:0]	All 0	R/W	Bit Rate Sets the bit rate. The bit rate is determined by a combination of these bits with the BRDV[1:0] bits. For details, see Section 17.4.3.1, Bit Rate .
7 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	BRDV[1:0]	11	R/W	Bit Rate Frequency Division Sets the bit rate. The bit rate is determined by a combination of these bits with the SPBR[7:0] bits. The SPBR[7:0] value is used to set the base bit rate. The BRDV[1:0] value is used to select a division ratio of the base bit rate from among no division, 2, 4, and 8. 00: Base bit rate 01: Base bit rate divided by 2 10: Base bit rate divided by 4 11: Base bit rate divided by 8

17.4.3.1 Bit Rate

SPBR[7:0] and BRDV[1:0] are used for setting the bit rate.

The following formula is used to calculate the bit rate when SPBR[7:0] ≠ 0.

$$\text{Bit rate} = B\phi / (2 \times n \times 2^N)$$

n: SPBR[7:0] setting (1, …, 255)
N: BRDV[1:0] setting (0 to 3)

17.4.4 Data Read Control Register (DRCR)

DRCR is a 32-bit register that sets the operation in external address space read mode.

The bits except the SSLN bit should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	BFM[2:0]			-	-	-	-	SSLN	-	-	-	-	RBURST[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R	W	R	R	R	R	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	RCF	RBE	-	-	-	-	-	-	-	SSLE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	W	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	BFM [2:0]	000	R/W	Enable SFMA cache 000: Burst length depends on the access size. Further the cache is flushed at each cycle. 111: Burst length of Serial Flash depends on values of RBE and RBURST[3:0]. The read cache is enabled when the RBE is set to 1. All others: Setting prohibited **Note:** If the SFMA is accessed by the CPU via XC Cache, make sure to also enable the SFMA cache as described in Section 17.6.3, Notes on using the SFMA Cache.
28 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	SSLN	0	W	SPBSSL Negation Asserted SPBSSL can be negated by writing 1 to this bit when both the RBE and SSLE bits are 1. This bit is always read as 0. Note: To start next access after SPBSSL negation using this bit, read SSLF in CMNSR = 0 to confirm that the SPBSSL has been negated.
23 to 20	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
19 to 16	RBURST [3:0]	0000	R/W	Read Data Burst Length Sets the burst length (data unit count) when reading. This bit is enabled when the RBE bit is set to 1. 0000: 1 data unit 0001: 2 continuous data units : 1110: 15 continuous data units 1111: 16 continuous data units One data unit is 64 bits long.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9	RCF	0	W	Read cache Flush When 1 is written to this bit, all the entries in the read cache are cleared. This bit is always read as 0. Note: After flushing the read cache by writing 1 to the RCF bit, read the DRCR before proceeding to read from the external address space.

Bit	Bit Name	Initial Value	R/W	Description
8	RBE	0	R/W	Read Burst Turns burst ON or OFF when reading. 0: Data is read according to the access size. 1: read cache is enabled, and as many data units as the burst count specified in RBURST[3:0] bits is read. Please set "111" to BFM[2:0] bit when this bit is set to 0.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SSLE	0	R/W	SPBSSL Negation Sets the conditions for SPBSSL negation during read burst. SPBSSL is negated for each access during normal read. 0: SPBSSL is negated after transfer of data set in burst length. 1: SPBSSL is negated when the accessed address is not continuous with the previously transferred address.

17.4.5 Data Read Command Setting Register (DRCMR)

DRCMR is a 32-bit register that sets the commands issued in external address space read mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	CMD[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	OCMD[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	CMD[7:0]	H'03	R/W	Command Sets the command.
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	OCMD[7:0]	H'00	R/W	Optional Command Sets the optional command.

17.4.6 Data Read Extended Address Setting Register (DREAR)

DREAR is a 32-bit register that sets the address when the serial flash address is output in 32-bit mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	EAV[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	EAC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	EAV[7:0]	H'00	R/W	32-Bit Extended Upper Address Fixed Value Sets the upper address bit values of the external address specified by the EAC[2:0] bits when the serial flash address is output in 32-bit mode. Bit 0 corresponds to the serial flash address bit [25], and bit 7 corresponds to the bit [32]. This setting is valid when the ADE[3] bit in DRENR is 1. When EAC[2:0] are 000, serial flash address [32:25] fixed values should be set to EAV[7:0]. When EAC[2:0] are 001, serial flash address [32:26] fixed values should set to EAV[7:1]. (1) When BSZ[1:0] in CMNCR = 00 (one serial flash memory connected) Serial flash addresses [31:0] are used for accessing. (2) When BSZ[1:0] in CMNCR = 01 (two serial flash memories connected) Serial flash addresses [32:1] are used for accessing.
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	EAC[2:0]	000	R/W	32-Bit Extended External Address Valid Range Sets the range of the external address to be used as serial flash address when the serial flash address is output in 32-bit mode. This setting is valid when the ADE[3] bit in DRENR is 1. 000: External address bits [24:0] enabled 001: External address bits [25:0] enabled 010: External address bits [26:0] enabled 011: External address bits [27:0] enabled 100: External address bits [28:0] enabled Other than above: Setting prohibited

17.4.7 Data Read Option Setting Register (DROPR)

DROPR is a 32-bit register that sets the option data in external address space read mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OPD3[7:0]								OPD2[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPD1[7:0]								OPD0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	OPD3[7:0]	H'00	R/W	Option Data 3 Sets the option data 3.
23 to 16	OPD2[7:0]	H'00	R/W	Option Data 2 Sets the option data 2.
15 to 8	OPD1[7:0]	H'00	R/W	Option Data 1 Sets the option data 1.
7 to 0	OPD0[7:0]	H'00	R/W	Option Data 0 Sets the option data 0.

Note: OPD3, OPD2, OPD1, and OPD0 are output in this order.

17.4.8 Data Read Enable Setting Register (DRENR)

DRENR is a 32-bit register that sets the bit size of the command, optional command, address, option data, and read data in external address space read mode and enables outputting them other than read data.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDB[1:0]		OCDB[1:0]		-	-	ADB[1:0]		-	-	OPDB[1:0]		-	-	DRDB[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DME	CDE	-	OCDE	ADE[3:0]				OPDE[3:0]				-	-	-	-
Initial value:	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31, 30	CDB[1:0]	00	R/W	Command Bit Size Sets the command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
29, 28	OCDB[1:0]	00	R/W	Optional Command Bit Size Sets the optional command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
27, 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25, 24	ADB[1:0]	00	R/W	Address Bit Size Sets the address size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	OPDB[1:0]	00	R/W	Option Data Bit Size Sets the option data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	DRDB[1:0]	00	R/W	Data Read Bit Size Sets the data read size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
15	DME	0	R/W	Dummy Cycle Enable Enables insertion of the dummy cycle before the read data. Note: A setting is prohibited for a transfer starting with a dummy cycle. 0: Dummy cycle insertion disabled 1: Dummy cycle insertion enabled
14	CDE	1	R/W	Command Enable Sets the command to be output. 0: Command output disabled 1: Command output enabled
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	OCDE	0	R/W	Optional Command Enable Sets the optional command to be output. 0: Optional command output disabled 1: Optional command output enabled
11 to 8	ADE[3:0]	0111	R/W	Address Enable Sets the address to be output. Be sure to use the following setting; otherwise, the operation is not guaranteed. (1) BSZ[1:0] in CMNCR = 00 (one serial flash memory connected) 0000: Output disabled 0111: Address[23:0] 1111: Address[31:0] Other than above: Setting prohibited (2) BSZ[1:0] in CMNCR = 01 (two serial flash memories connected) 0000: Output disabled 0111: Address[24:1] 1111: Address[32:1] Other than above: Setting prohibited
7 to 4	OPDE[3:0]	0000	R/W	Option Data Enable Sets the option data to be output. Use only the settings given below. Otherwise, the operation cannot be guaranteed. 0000: Output disabled 1000: OPD3 1100: OPD3, OPD2 1110: OPD3, OPD2, OPD1 1111: OPD3, OPD2, OPD1, OPD0 Other than above: Setting prohibited
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

17.4.9 SPI Mode Control Register (SMCR)

SMCR is a 32-bit register that sets the operation in SPI operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	SSLKP	-	-	-	-	-	SPIRE	SPIWE	SPIE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SSLKP	0	R/W	SPBSSL Signal Level Determines the SPBSSL status after the end of transfer. 0: SPBSSL signal is negated at the end of transfer. 1: SPBSSL signal level is maintained from the end of transfer to the start of next access.
7 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2	SPIRE	0	R/W	Data Read Enable Enables reading in SPI operating mode. 0: Data reading disabled 1: Data reading enabled Note: When the transfer data bit size is set to 2 bits or 4 bits with the SPIDB[1:0] bits, the SPIRE and SPIWE bits should not be set to 1 at the same time.
1	SPIWE	0	R/W	Data Write Enable Enables writing in SPI operating mode. 0: Data writing disabled 1: Data writing enabled Note: When the transfer data bit size is set to 2 bits or 4 bits with the SPIDB[1:0] bits, the SPIRE and SPIWE bits should not be set to 1 at the same time.
0	SPIE	0	W	SPI Data Transfer Enable Data is transferred by setting this bit to 1. This bit is enabled only when the TEND bit in CMNSR is set to 1. The operation cannot be guaranteed when this bit is set to 1 with the TEND bit set to 0. This bit is always read as 0. Note: When the SPBSSL pin is de-asserted, the command, optional command, address, and option data that are output enabled are output even if the SPIRE and SPIWE bits are set to 0. When the SPBSSL pin is asserted, follow the notes described in Section 17.6.2, Notes on Starting Transfer from the SPBSSL Retained State in SPI Operating Mode.

17.4.10 SPI Mode Command Setting Register (SMCMR)

SMCMR is a 32-bit register that sets the commands issued in SPI operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	CMD[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	OCMD[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	CMD[7:0]	H'00	R/W	Command Sets the command.
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	OCMD[7:0]	H'00	R/W	Optional Command Sets the optional command.

17.4.11 SPI Mode Address Setting Register (SMADR)

SMADR is a 32-bit register that sets the addresses in SPI operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR[31:24]								ADR[23:16]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	ADR[31:24]	H'00	R/W	Address Sets the value of bits 31 to 24 when the serial flash address is output in 32-bit units. This setting is valid when ADE[3] in SMENR is 1.
23 to 0	ADR[23:0]	H'000000	R/W	Address Sets the address.

17.4.12 SPI Mode Option Setting Register (SMOPR)

SMOPR is a 32-bit register that sets the option data in SPI operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OPD3[7:0]								OPD2[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPD1[7:0]								OPD0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	OPD3[7:0]	H'00	R/W	Option Data 3 Sets the option data 3.
23 to 16	OPD2[7:0]	H'00	R/W	Option Data 2 Sets the option data 2.
15 to 8	OPD1[7:0]	H'00	R/W	Option Data 1 Sets the option data 1.
7 to 0	OPD0[7:0]	H'00	R/W	Option Data 0 Sets the option data 0.

Note: OPD3, OPD2, OPD1, and OPD0 are output in this order.

17.4.13 SPI Mode Enable Setting Register (SMENR)

SMENR is a 32-bit register that sets the bit size of the command, optional command, address, option data, and transfer data in SPI operating mode and enables their output. SMENR also enables dummy cycle insertion. Disabling all of the command, optional command, address, option data, dummy cycle, and transfer data is prohibited. At least one of them except dummy cycle must be enabled.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDB[1:0]		OCDB[1:0]		-	-	ADB[1:0]		-	-	OPDB[1:0]		-	-	SPIDB[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DME	CDE	-	OCDE	ADE[3:0]				OPDE[3:0]				SPIDE[3:0]			
Initial value:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	CDB[1:0]	00	R/W	Command Bit Size Sets the command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
29, 28	OCDB[1:0]	00	R/W	Optional Command Bit Size Sets the optional command size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
27, 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25, 24	ADB[1:0]	00	R/W	Address Bit Size Sets the address size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
23, 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	OPDB[1:0]	00	R/W	Option Data Bit Size Sets the option data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	SPIDB[1:0]	00	R/W	Transfer Data Bit Size Sets the transfer data size in bit units. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited

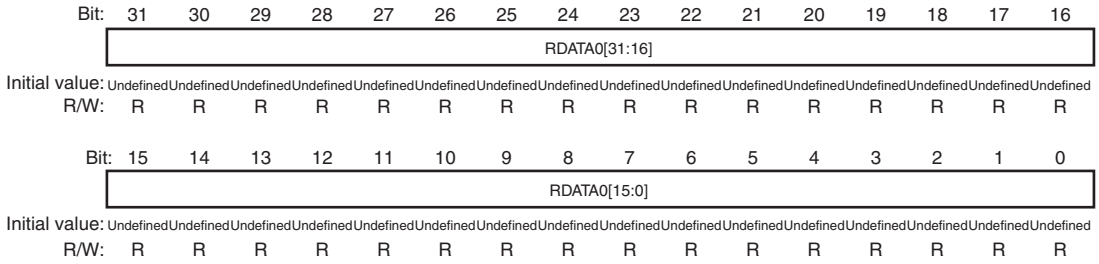
Bit	Bit Name	Initial Value	R/W	Description
15	DME	0	R/W	Dummy Cycle Enable Enables insertion of the dummy cycle before the read data. Note: Dummy cycle insertion is prohibited for write in SPI operating mode including the case in which a transfer ends with a dummy cycle. Note: A setting is prohibited for a transfer starting with a dummy cycle. 0: Dummy cycle insertion disabled 1: Dummy cycle insertion enabled
14	CDE	1	R/W	Command Enable Sets the command to be output. 0: Command output disabled 1: Command output enabled
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	OCDE	0	R/W	Optional Command Enable Sets the optional command to be output. 0: Optional command output disabled 1: Optional command output enabled
11 to 8	ADE[3:0]	0000	R/W	Address Enable Sets the address to be output. Use only the settings given below. Otherwise, the operation cannot be guaranteed. 0000: Output disabled 0100: ADR[23:16] 0110: ADR[23:8] 0111: ADR[23:0] 1111: ADR[31:0] Other than above: Setting prohibited
7 to 4	OPDE[3:0]	0000	R/W	Option Data Enable Sets the option data to be output. Use only the settings given below. Otherwise, the operation cannot be guaranteed. 0000: Output disabled 1000: OPD3 1100: OPD3, OPD2 1110: OPD3, OPD2, OPD1 1111: OPD3, OPD2, OPD1, OPD0 Other than above: Setting prohibited
3 to 0	SPIDE[3:0]	0000	R/W	Transfer Data Enable Sets valid transfer data. Valid data differs depending on the BSZ[1:0] bit setting in CMNCR. The following settings must be used. Otherwise, the operation is not guaranteed. (1) BSZ[1:0] bits in CMNCR = 00 (one serial flash memory connected) 0000: Not transferred 1000: 8 bits transferred (enables data at address 0 of the SPI mode read/write data registers 0) 1100: 16 bits transferred (enables data at addresses 0 and 1 of the SPI mode read/write data registers 0) 1111: 32 bits transferred (enables data at addresses 0 to 3 of the SPI mode read/write data registers 0) Other than above: Setting prohibited (2) BSZ[1:0] bits in CMNCR = 01 (two serial flash memories connected) 0000: Not transferred 1000: 16 bits transferred (enables data at addresses 0 and 1 of the SPI mode read/write data registers 0) 1100: 32 bits transferred (enables data at addresses 0 to 3 of the SPI mode read/write data registers 0) 1111: 64 bits transferred (enables data at addresses 0 to 3 of the SPI mode read/write data registers 0 and data at addresses 0 to 3 of the SPI mode read/write data registers 1) Other than above: Setting prohibited

17.4.14 SPI Mode Read Data Register 0 (SMRDR0)

SMRDR0 is a 32-bit register that stores the read data in SPI operating mode.

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the SPI mode enable setting register (SMENR). Be sure to access from address 0.

The settings of this register should be read when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDATA0 [31:0]	Undefined	R	Read Data Holds the data read in SPI operating mode. Data bits differ depending on the settings of SFDE and BSZ[1:0] bits in CMNCR and SPIDE[3:0] bits in SMENR. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 1: Read data[63:32]. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 0: Read data[31:0]. Other than the above: Read data[31:0].

Note: The contents of this register and SMRDR1 are modified upon completion of reception in SPI operating mode. Be sure to read data when reception in SPI operating mode is completed.

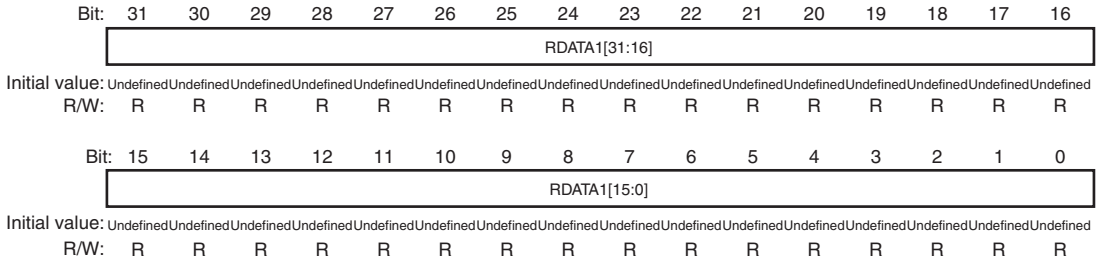
17.4.15 SPI Mode Read Data Register 1 (SMRDR1)

SMRDR1 is a 32-bit register that stores the read data in SPI operating mode.

This register is enabled when the BSZ[1:0] bits in CMNCR are set to 01 (two serial flash memories connected) and disabled when the BSZ[1:0] bits in CMNCR are set to 00 (one serial flash memory connected).

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the SPI mode enable setting register (SMENR). Be sure to access from address 0.

The settings of this register should be read when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.



Bit	Bit Name	Initial Value	R/W	Description
31 to 0	RDATA1 [31:0]	Undefined	R	Read Data Holds the data read in SPI operating mode. Data bits differ depending on the settings of SFDE and BSZ[1:0] bits in CMNCR and SPIDE[3:0] bits in SMENR. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 1: Read data[31:0]. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 0: Read data[63:32]. Other than the above: Bits in this register are disabled.

17.4.16 SPI Mode Write Data Register 0 (SMWDR0)

SMWDR0 is a 32-bit register that sets the write data in SPI operating mode.

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the SPI mode enable setting register (SMENR). Be sure to access from address 0.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDATA0[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDATA0[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	WDATA0 [31:0]	All 0	R/W	Write Data Holds the data to be written in SPI operating mode. Data bits differ depending on the settings of SFDE and BSZ[1:0] bits in CMNCR and SPIDE[3:0] bits in SMENR. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 1: Write data[63:32]. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 0: Write data[31:0]. Other than the above: Write data[31:0].

17.4.17 SPI Mode Write Data Register 1 (SMWDR1)

SMWDR1 is a 32-bit register that sets the write data in SPI operating mode.

This register is enabled when the BSZ[1:0] bits in CMNCR are set to 01 (two serial flash memories connected) and disabled when the BSZ[1:0] bits in CMNCR are set to 00 (one serial flash memory connected).

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the SPI mode enable setting register (SMENR). Be sure to access from address 0.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDATA1[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDATA1[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	WDATA1 [31:0]	All 0	R/W	Write Data Holds the data to be written in SPI operating mode. Data bits differ depending on the settings of SFDE and BSZ[1:0] bits in CMNCR and SPIDE[3:0] bits in SMENR. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 1: Write data[31:0]. BSZ[1:0] = 01, SPIDE[3:0] = 1111, SFDE = 0: Write data[63:32]. Other than the above: Bits in this register are disabled.

17.4.18 Common Status Register (CMNSR)

CMNSR is a 32-bit register that holds flags indicating the operating state.

The settings of this register are reflected both in external address space read mode and SPI operating mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	SSLF	TEND
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1	SSLF	0	R	SPBSSL Pin Monitor 0: SPBSSL pin is negated 1: SPBSSL pin is asserted
0	TEND	1	R	Transfer End Flag Indicates whether the data transfer has ended. 0: Indicates that data transfer is in progress 1: Indicates that data transfer has ended

17.4.19 Clock phase adjust register (CKDLY)

This register defines phase shifts between the SPBCLK clock, sampling point and input/output data.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

For detailed information about how to set the register values, refer to Section 17.5.14, Adjustment between input/output data and sampling clocks.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PROT_CKDLY[7:0]								-	-	CKDLY_TS[1:0]*1		-	CKDLYOC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	CKDLYRX[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Note 1. These bits are only available in SFMA1 of D1M1A and D1M1-V2.

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	PROT_ CKDLY[7:0]	All 0	W	CKDLY write protection While changing the settings of this register, set PROT_CKDLY[7:0] = $A5_H$. If this register is written with a different PROT_CKDLY[7:0] value, writing is ignored.
23 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	CKDLY_TS[1:0]	All 0	R/W	Output timing adjustment 00_B: No shift 01_B: 0.5 f_{PHCLK} cycle delay from SPBCLK 10_B: 1.0 f_{PHCLK} cycle delay from SPBCLK 11_B: 1.5 f_{PHCLK} cycle delay from SPBCLK *1 Note 1. This selection is only available when CKSC_ISFMAD_CTL = 011_B or 100_B.
19	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	CKDLYOC[2:0]	0	R/W	SPBCLK clock phase shift CKDLYOC[2:0] defines a phase shift between the SPBCLK clock and data output towards the external flash memory devices. The phase shift value delays the data outputs with respect to the SPBCLK clock by the defined value. The phase shift is set in units of $1/(2xf_{PHCLK})$, i.e. in half of the phase shift clock PHCLK cycle time. 000_B: no phase shift 010_B: phase shift = 2 x $[1/(2xf_{PHCLK})]$ - $1/f_{B\Phi}$ 011_B: phase shift = 3 x $[1/(2xf_{PHCLK})]$ - $1/f_{B\Phi}$ 100_B: phase shift = 4 x $[1/(2xf_{PHCLK})]$ - $1/f_{B\Phi}$ 101_B: phase shift = 5 x $[1/(2xf_{PHCLK})]$ - $1/f_{B\Phi}$ 110_B: phase shift = 6 x $[1/(2xf_{PHCLK})]$ - $1/f_{B\Phi}$ 111_B: phase shift = 7 x $[1/(2xf_{PHCLK})]$ - $1/f_{B\Phi}$ others: setting prohibited CAUTION The phase shift must not exceed the cycle time of the BΦ clock.
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	CKDLYRX[2:0]	0	R/W	Sampling point phase shift CKDLYRX[2:0] defines a phase shift between the input data from the external flash memory devices and the sampling point. The sampling point shift is set in units of $1/2xf_{PHCLK}$, i.e. in half of the phase shift clock PHCLK cycle time. 000_B: no sampling point shift 001_B: sampling point shift = 1 x $[1/(2xf_{PHCLK})]$ 010_B: sampling point shift = 2 x $[1/(2xf_{PHCLK})]$ 011_B: sampling point shift = 3 x $[1/(2xf_{PHCLK})]$ 100_B: sampling point shift = 4 x $[1/(2xf_{PHCLK})]$ 101_B: sampling point shift = 5 x $[1/(2xf_{PHCLK})]$ others: setting prohibited*1 Note 1. For calibration usage only, – 110_B (sampling point shift = 6 x $[1/(2xf_{PHCLK})]$) and – 111_B (sampling point shift = 7 x $[1/(2xf_{PHCLK})]$) can be also used.

17.4.20 Data Read Dummy Cycle Setting Register (DRDMCR)

DRDMCR is a 32-bit register that sets the size and number of dummy cycles to be inserted in external address space read mode.

The settings of this register are enabled when the DME bit in the data read enable setting register (DRENR) is 1.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DMDB[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	DMCYC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	DMDB [1:0]	00	R/W	Dummy Cycle Bit Size Sets the dummy cycle size in bit units. The setting of these bits is combined with the setting of the IO0FV[1:0], IO2FV[1:0], and IO3FV[1:0] bits in the common control register (CMNCR) to determine the state of the unused pins during the dummy cycles. The state of the used pins is Hi-Z. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	DMCYC [2:0]	000	R/W	Number of Dummy Cycles Sets the number of dummy cycles to be inserted when the DME bit in the data read enable setting register (DRENR) is 1. 000: 1 cycle 001: 2 cycles 010: 3 cycles 011: 4 cycles 100: 5 cycles 101: 6 cycles 110: 7 cycles 111: 8 cycles

17.4.21 Data Read DDR Enable Register (DRDRENR)

DRDRENR is a 32-bit register that sets SDR or DDR transfer of the address, option data, and read data in external address space read mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ADDRE	-	-	-	OPDRE	-	-	-	DRDRE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ADDRE	0	R/W	Address DDR Enable Sets SDR or DDR transfer of the address. 0: SDR transfer 1: DDR transfer
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	OPDRE	0	R/W	Option Data DDR Enable Sets SDR or DDR transfer of the option data. 0: SDR transfer 1: DDR transfer
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DRDRE	0	R/W	Data Read DDR Enable Sets SDR or DDR transfer of the read data. 0: SDR transfer 1: DDR transfer

17.4.22 SPI Mode Dummy Cycle Setting Register (SMDMCR)

SMDMCR is a 32-bit register that sets the size and number of dummy cycles to be inserted in SPI operating mode.

The settings of this register are enabled when the DME bit in the SPI mode enable setting register (SMENR) is 1.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DMDB[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	DMCYC[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	DMDB [1:0]	00	R/W	Dummy Cycle Bit Size Sets the dummy cycle size in bit units. The setting of these bits is combined with the setting of the IO0FV[1:0], IO2FV[1:0], and IO3FV[1:0] bits in the common control register (CMNCR) to determine the state of the unused pins during the dummy cycles. The state of the used pins is Hi-Z. 00: 1 bit 01: 2 bits 10: 4 bits 11: Setting prohibited
15 to 3	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
2 to 0	DMCYC[2:0]	000	R/W	Number of Dummy Cycles Sets the number of dummy cycles to be inserted when the DME bit in the SPI mode enable setting register (SMENR) is 1. 000: 1 cycle 001: 2 cycles 010: 3 cycles 011: 4 cycles 100: 5 cycles 101: 6 cycles 110: 7 cycles 111: 8 cycles

17.4.23 SPI Mode DDR Enable Register (SMDRENR)

SMDRENR is a 32-bit register that sets SDR or DDR transfer of the address, option data, and transfer data in SPI operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	ADDRE	-	-	-	OPDRE	-	-	-	SPIDRE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	ADDRE	0	R/W	Address DDR Enable Sets SDR or DDR transfer of the address. 0: SDR transfer 1: DDR transfer
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	OPDRE	0	R/W	Option Data DDR Enable Sets SDR or DDR transfer of the option data. 0: SDR transfer 1: DDR transfer
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SPIDRE	0	R/W	Transfer Data DDR Enable Sets SDR or DDR transfer of the transfer data. 0: SDR transfer 1: DDR transfer

17.4.24 Output data delay register (SPODLY)

This register defines a fixed delay between the SPBCLK clock and the data output signals SPIO[3:0]i (i = 0, 1).

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

For detailed information about how to set the register values, refer to Section 17.5.14, Adjustment between input/output data and sampling clocks.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PROT_SPODLY[7:0]								-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	W	W	W	W	W	W	W	W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	DLYOE1	-	-	-	DLYO1	-	-	-	DLYOE0	-	-	-	DLYO0
Initial value:	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	PROT_ SPODLY[7:0]	All 0	W	SPODLY write protection While changing the settings of this register, set PROT_SPODLY[7:0] = $A5_H$. If this register is written with a different PROT_SPODLY[7:0] value, writing is ignored.
23 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12 8	DLYOE1 DLYO1	1 1	R/W	Enable port 1 output delay 1: Output delay minimum 2 ns*1 0: No output delay **CAUTION** Both bits DLYOE1 and DLYO1 must be changed together to the same value.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4 0	DLYOE0 DLYO0	1 1	R/W	Enable port 0 output delay 1: Output delay minimum 2 ns*1 0: No output delay **CAUTION** Both bits DLYOE0 and DLYO0 must be changed together to the same value.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note 1. Refer to the Data Sheet for the range of the output delay.

17.5 Operation

17.5.1 System Configuration

With this module, one or two serial flash memories can be directly connected per channel (data size of 1, 2, and 4 bits). The number of connected memories can be selected using the BSZ[1:0] bits in CMNCR.

Examples of system configuration with one serial flash memory connected and two serial flash memories connected are shown in **Figure 17.2** and **Figure 17.3**, respectively.

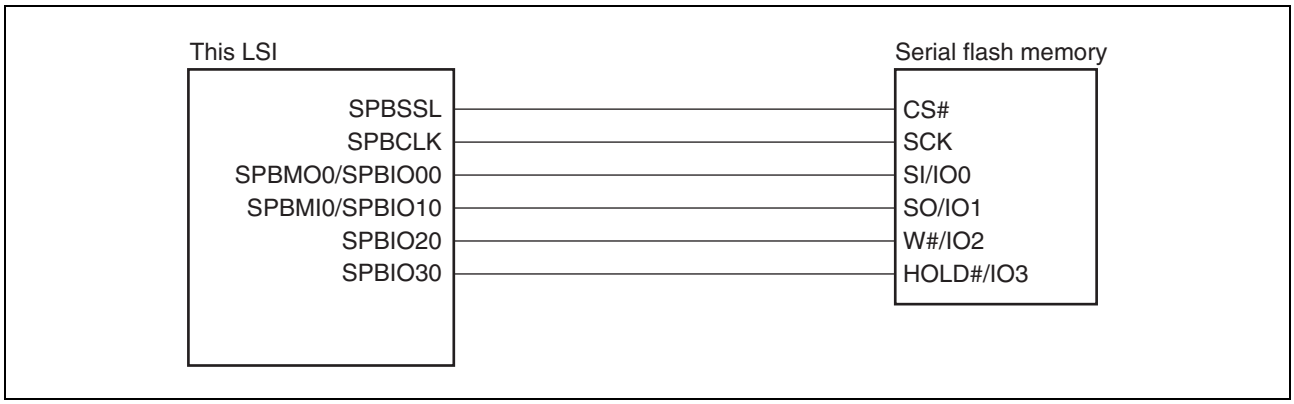


Figure 17.2 System Configuration Example with 4-Bit Data Size and One Serial Flash Memory Connected (BSZ[1:0] Bits in CMNCR = 00)

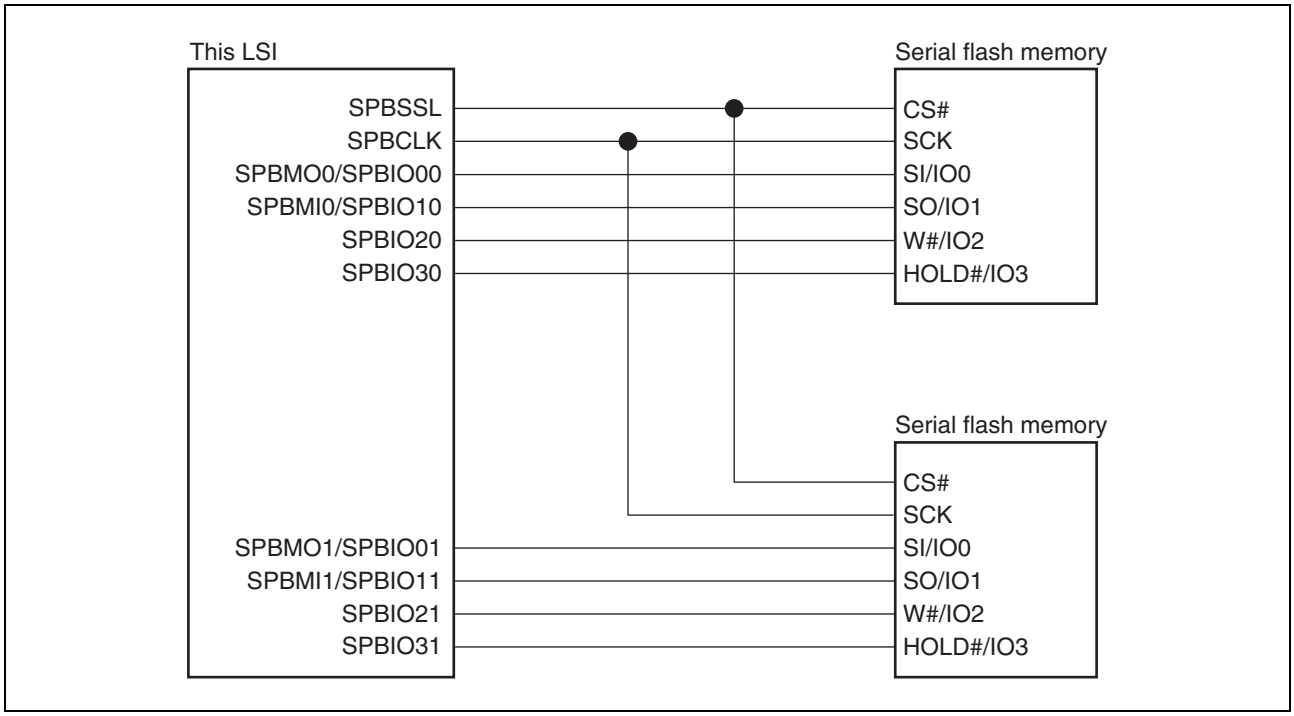


Figure 17.3 System Configuration Example with 4-Bit Data Size and Two Serial Flash Memories Connected (BSZ[1:0] Bits in CMNCR = 01)

17.5.2 Address Map

In external address space read mode, the serial flash connected is assigned in the SPI multi I/O bus space. A maximum accessible address space differs depending on the number of serial flash memories connected. In combination with DREAR, a maximum of 4 Gbytes can be accessed when one serial flash memory is connected, and a maximum of 8 Gbytes can be accessed when two memories are connected.

Table 17.7 Address Map

Number of Serial Flash Memories Connected	Internal Address	Max. Access Area
1	H'1000 0000 to H'2FFF FFFF	4 Gbytes
2		8 Gbytes

17.5.3 32-bit Serial Flash Addresses

Since the SPI multi I/O bus space is 512 Mbytes, only a part of the 32-bit serial flash address area can be directly accessed. Here, the fixed value set in the pertinent register is used as the upper bit value of a 32-bit address.

To output serial flash addresses in 32 bits, set the ADE[3] bit in DRENR to 1, set the range of the external addresses used as the serial flash addresses to the EAC[2:0] bits in DREAR, and set the upper bit value of the 32-bit address as the fixed value to the EAV[7:0] bits in DREAR.

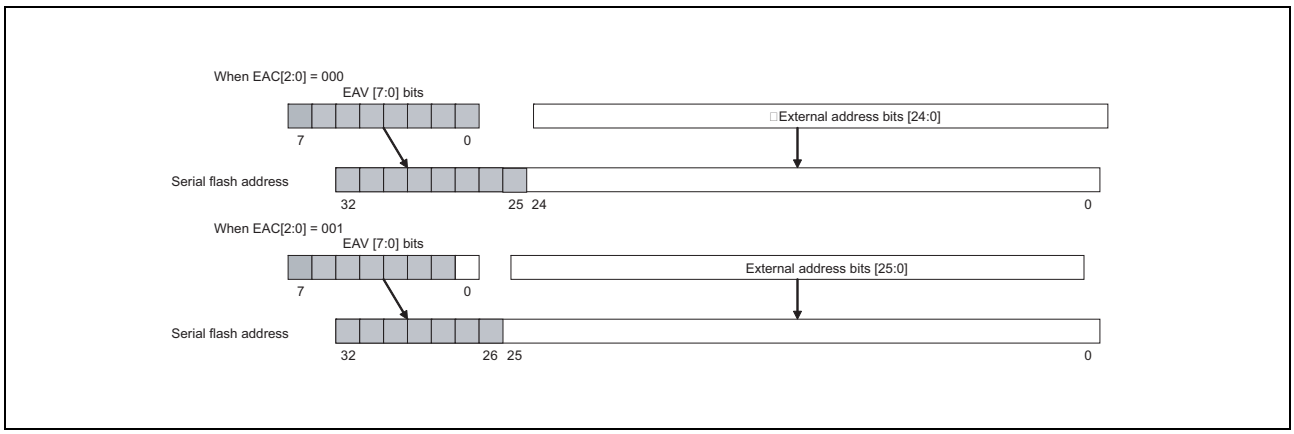


Figure 17.4 32-Bit Address Setting (example for EAC[2:0] = 000 and 001)

Setting the ADE[3] bit in DRENR to 1 allows the serial flash address to be output using [31:0] bits.
When EAC[2:0] = 000, external address bits [24:0] are valid; set the value for [32:25] bits to EAV[7:0].
When EAC[2:0] = 001, external address bits [25:0] are valid; set the value for [32:26] bits to EAV[7:1].
When EAC[2:0] = 010, external address bits [26:0] are valid; set the value for [32:27] bits to EAV[7:2].
When EAC[2:0] = 011, external address bits [27:0] are valid; set the value for [32:28] bits to EAV[7:3].
When EAC[2:0] = 100, external address bits [28:0] are valid; set the value for [32:29] bits to EAV[7:4].

The address bits actually used for access depend on the number of serial flash memories connected. When one serial flash memory is connected, address bits [31:0] are used and when two memories are connected, address bits [32:1] are used.

NOTE

When the capacity of the serial flash memory used is smaller than 4 Gbytes, keep the following point in mind.
If an access spreads over the last address of the serial flash in burst mode (RBE bit in DRCR = 1), the access address does not agree with the internal address of the serial flash. To prevent this, software should appropriately manage the accessible address areas for the serial flash memory used according to the memory capacity.

17.5.4 Data Alignment

Data alignment can be set by using the SFDE bit in the common control register (CMNCR). Data alignment in data read mode and in SPI mode are shown in **Figure 17.5** and **Figure 17.6**, respectively.

When two serial flash memories are connected, the serial flash memory connected to the pin SPBIO30-SPBIO00 has the address 2n and the serial flash memory connected to the pin SPBIO31-SPBIO01 has the address 2n + 1. The data must be accessed in word or larger units. It cannot be accessed in byte units.

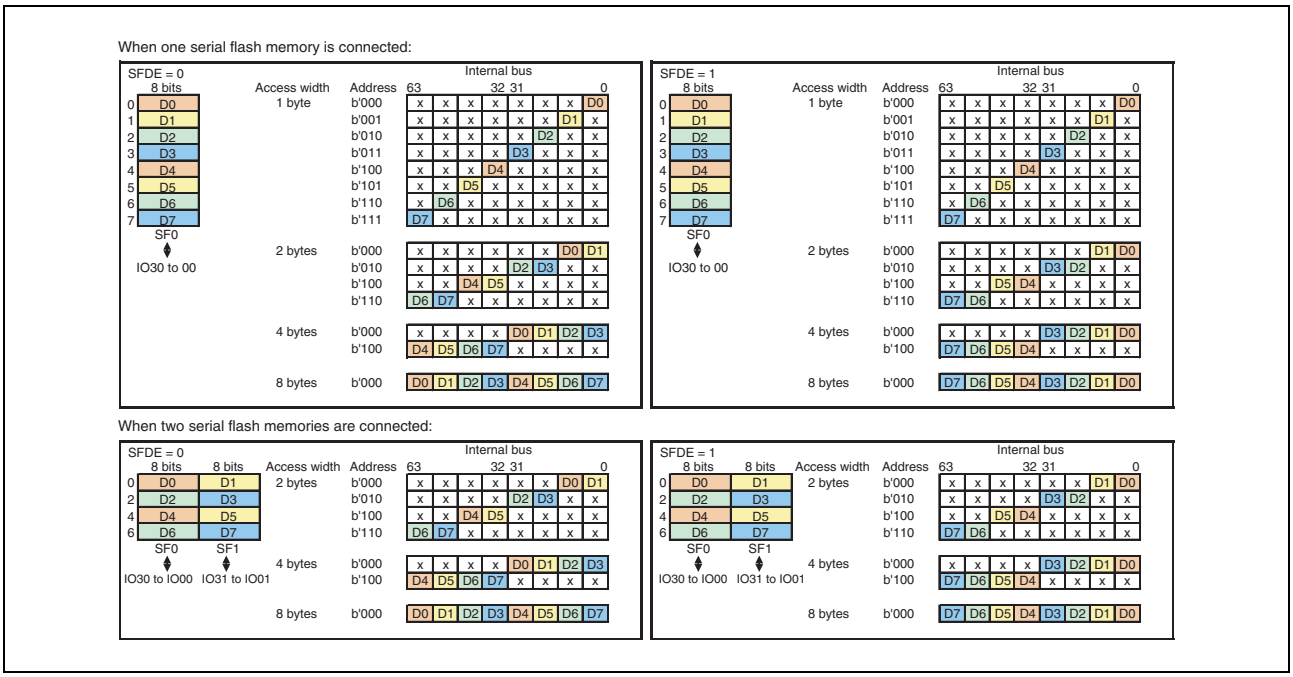


Figure 17.5 Data Alignment in Data Read Mode

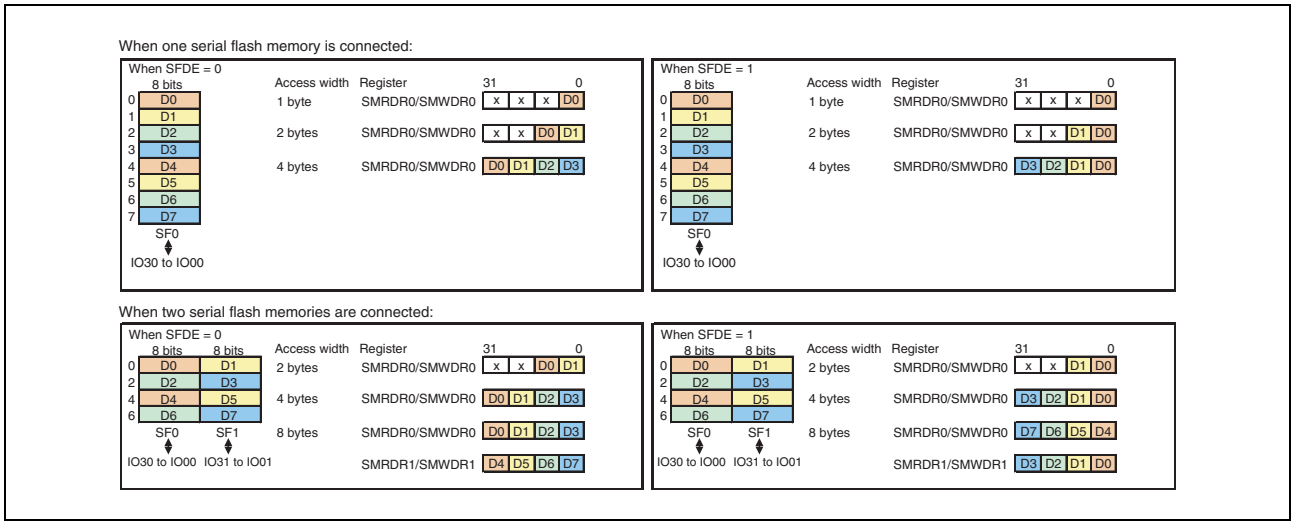


Figure 17.6 Data Alignment in SPI Mode

17.5.5 Operating Modes

This module has two operating modes: external address space read mode and SPI operating mode.

In external address space read mode, a read access to the SPI multi I/O bus space is converted into SPI communication and data is received. After data acquisition, data is returned to the bus master that is the issuing source. For details, see Section 17.5.6, External Address Space Read Mode.

In SPI operating mode, arbitrary SPI communication is carried out using register settings. For details, see Section 17.5.8, SPI Operating Mode.

17.5.6 External Address Space Read Mode

A read access to the SPI multi I/O bus space can be converted into SPI communication in external address space read mode. Further, the commands, optional commands, option data, and dummy cycle issued for reading can be modified using registers. For the address, option data, and read data, either SDR or DDR transfer can be selected using the appropriate register when the SPBCLK frequency division ratio is two or larger.

In external address space read mode, either normal read operation or burst read operation can be selected. The transfer format is determined based on the common control register (CMNCR), SSL delay register (SSLDR), bit rate setting register (SPBCR), data read control register (DRCR), data read command setting register (DRCMR), data read extended address setting register (DREAR), data read option setting register (DROPR), data read enable setting register (DRENR), data read dummy cycle setting register (DRDMCR), and data read DDR enable register (DRDRENR).

17.5.6.1 Normal Read Operation

When the RBE bit in DRCR is set to 0, normal read operation is performed.

In the normal read operation, the data of 8 bits, 16 bits, 32 bits, and 64 bits are read for respectively a byte, a word, and a longword read access. Here, a byte access is enabled only when one serial flash memory is connected. After reading, the SPBSSL pin is negated.

The normal read operation timing is shown in **Figure 17.7**.

t1 is the time period from SPBSSL pin assertion to SPBCLK oscillation (clock delay), t2 is the time period from transmission of the last SPBCLK edge of a transfer to SPBSSL pin negation (SPBSSL

negation delay), and t3 is the time period from one transfer end to the next transfer start (next access). For details of t1, t2, and t3, see Section 17.5.9, Transfer Format.

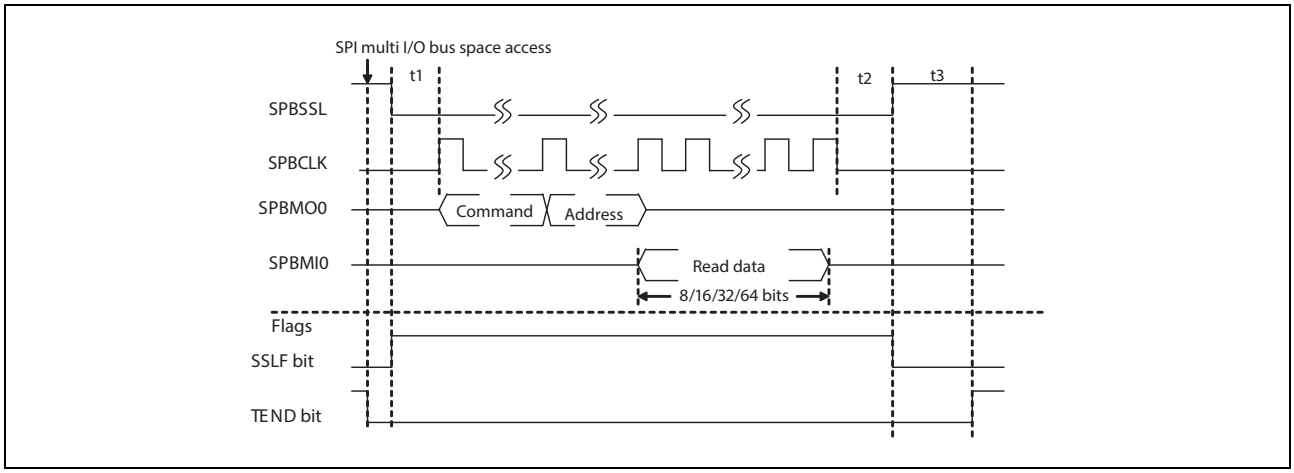


Figure 17.7 Normal Read Operation Timing

17.5.6.2 Burst Read Operation

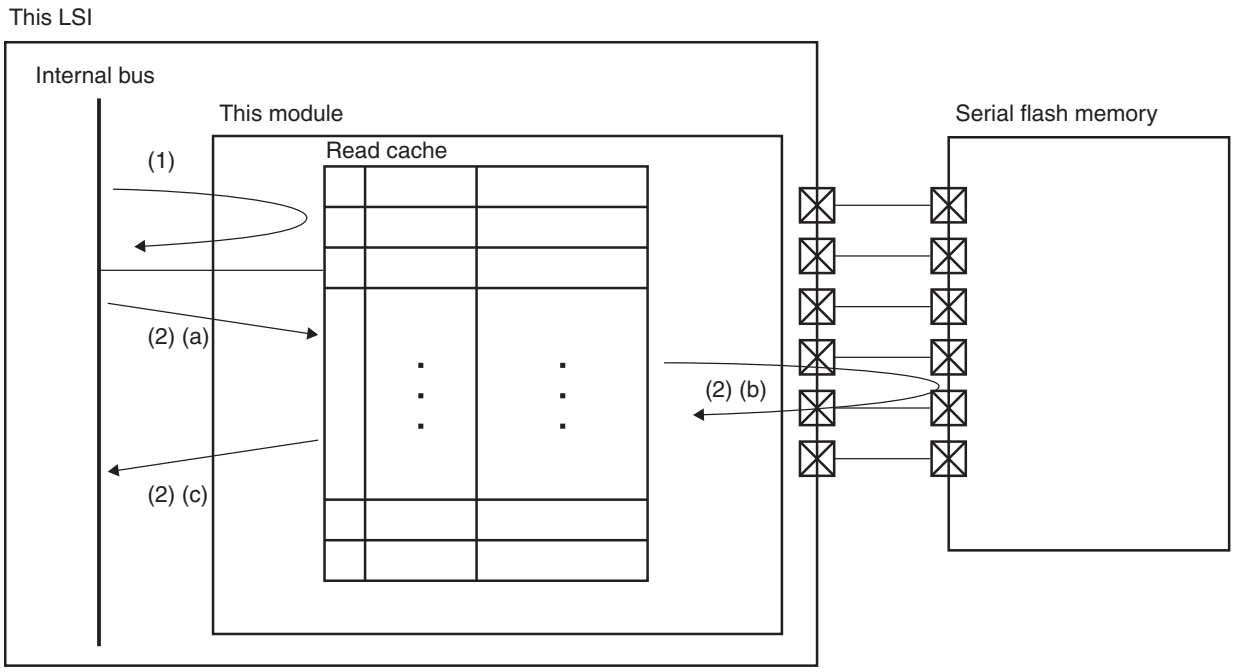
When the RBE bit in DRCR is set to 1, burst read operation is performed.

read cache is enabled in the burst read operation. For read cache operation, see Section 17.5.7, Read Cache.

For reading bytes, words, or longwords, the read cache is first referred to for the data. When the read cache contains the data, the data is read from the read cache without accessing the serial flash memory. When the read cache does not contain the data, burst read operation is performed in the serial flash memory and the read data is stored in the read cache. The data transfer length at that time is 64 bits × (RBURST[3:0] + 1) bits and the data is always read from the 64-bit boundary.

The SPBSSL pin status after data transfer can be selected by using the SSLE bit in DRCR. When the SSLE bit is set to 0, the SPBSSL pin is negated after data transfer. For an operation performed when the SSLE bit is set to 1, see Section 17.5.6.3, Burst Read Operation with Automatic SPBSSL Negation, just below.

A pattern diagram of this operation and a burst read operation timing diagram when SSLE bit is set to 0 are shown in **Figure 17.8** and **Figure 17.9**.



(1) When the read cache contains the data
The data is read from the read cache without performing SPI communication.

(2) When the read cache does not contain the data
(a) The read cache is accessed to confirm that the data is not in the read cache.
(b) The data is read from the serial flash memory and the read data is stored in the read cache.
(c) The data is read from the read cache.

Figure 17.8 Burst Read Operation

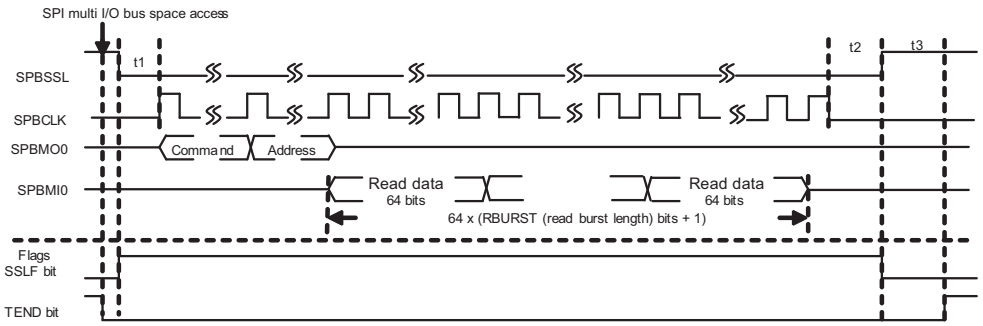


Figure 17.9 Burst Read Operation Timing (SSLE Bit = 0)

17.5.6.3 Burst Read Operation with Automatic SPBSSL Negation

When SSLE bit in DRCR is set to 1, this module does not negate the SPBSSL pin after the burst read transfer. When accessing the next time, if the address is continuous with the previous read address, the burst read operation is performed without issuing the command, optional command, address, option data, or dummy cycle. If the address is not continuous with the previous read address, the SPBSSL pin is once negated and the burst read operation is performed after issuing the command, optional command, address, option data, or dummy cycle.

Burst read timing diagrams for continuous address and non-continuous address are shown in **Figure 17.10** and **Figure 17.11**.

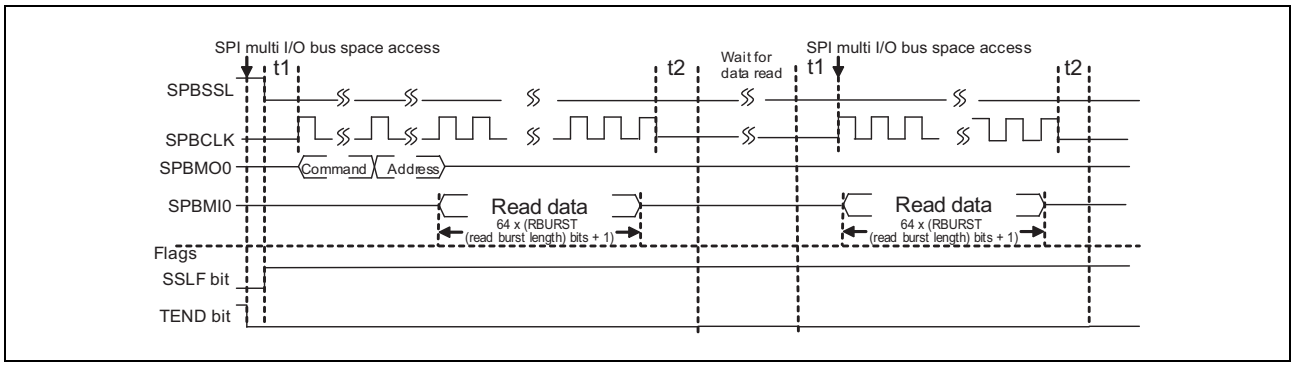


Figure 17.10 Burst Read Timing for Continuous Address (SSLE Bit = 1)

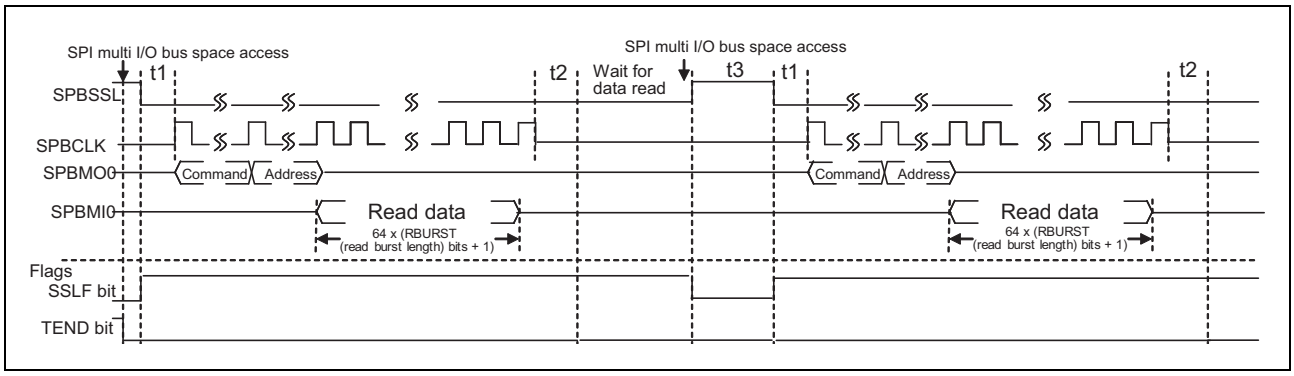


Figure 17.11 Burst Read Timing for Non-Continuous Address (SSLE Bit = 1)

For the next access after negation of the SPBSSL with the SSLN bit in DRCR with this operation, read SSLF = 0 in CMNSR to confirm that the SPBSSL has been negated.

17.5.6.4 Initial Setting Flow

An example of an initial setting flow in external address space read mode is shown in **Figure 17.12**.

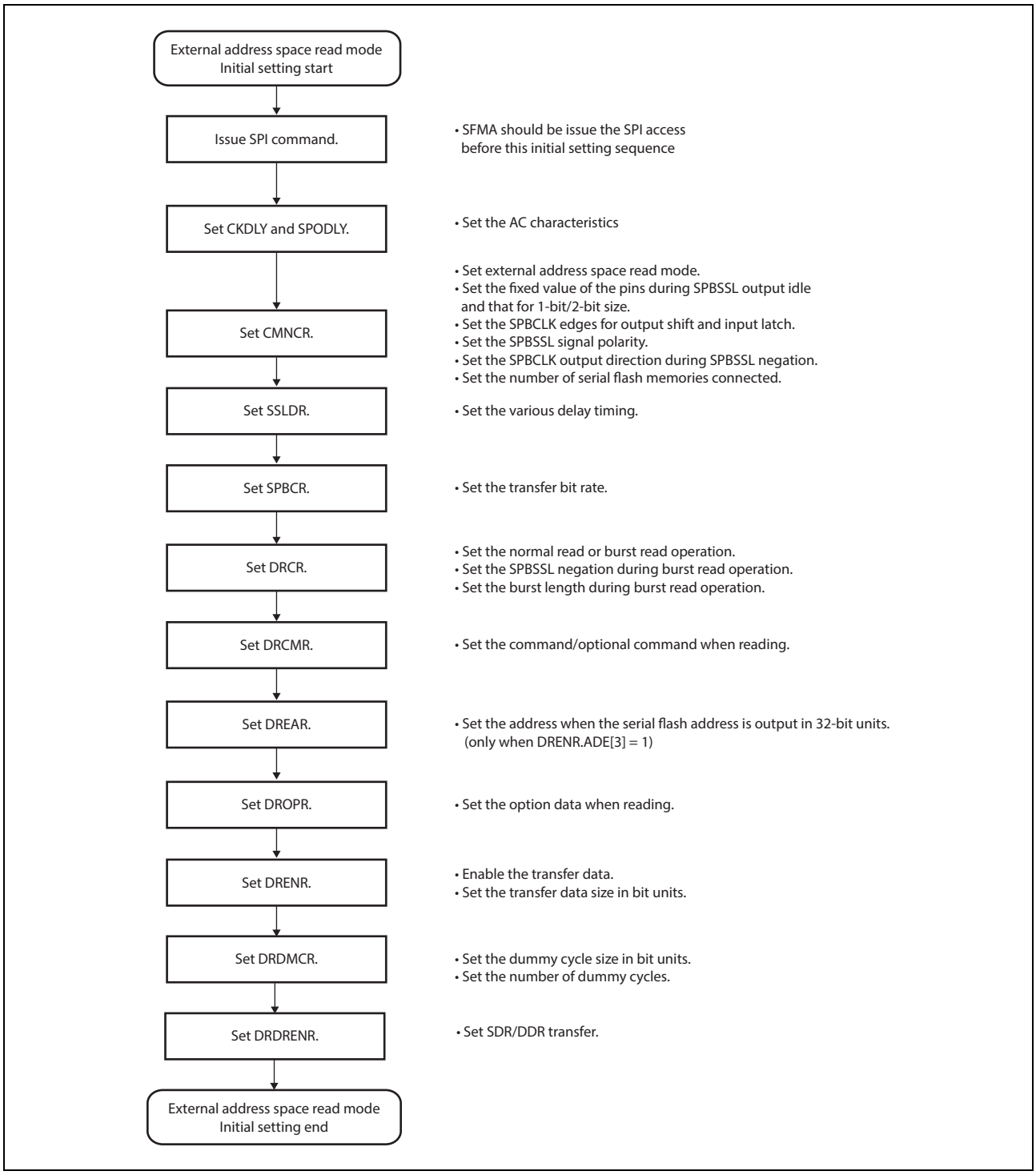


Figure 17.12 Example of Initial Setting Flow in External Address Space Read Mode

17.5.7 Read Cache

This module has a simple built-in read cache. The read cache can be used during external address space read mode and burst read operation. The read cache is configured with a line size of 64 bits and 16 entries.

Read cache configuration is shown in **Figure 17.13**.

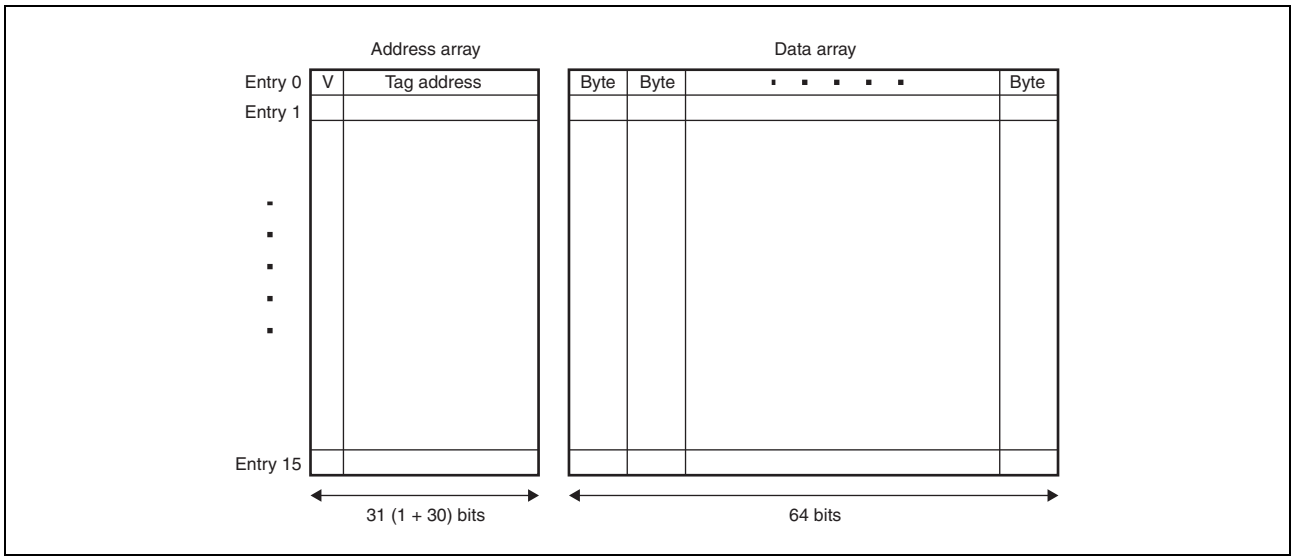


Figure 17.13 Read Cache Configuration

NOTE

If the SFMA is accessed by the CPU via XC Cache, make sure to also enable the SFMA cache as described in Section 17.6.3, Notes on using the SFMA Cache.

17.5.7.1 Address Array

The V bit indicates whether the entry data is valid. When the V bit is 1, the data is valid and when V bit is 0, the data is invalid.

The tag address bits hold the address used for the serial flash memory. Address bits 32 to 3 are used for the tag address.

Address bits 23 to 3 are enabled when address output is 24 bits and one serial flash memory is connected; and address bits 24 to 3 are enabled when two serial flash memories are connected.

Address bits 31 to 3 are enabled when address output is 32 bits and one serial flash memory is connected; and address bits 32 to 3 are enabled when two serial flash memories are connected.

17.5.7.2 Data Array

It retains the 64-bit read data. Registration in the read cache is performed in line units.

17.5.7.3 Read Operation

In case of read-hit, data is read from the read cache. In case of read-miss, after the 64 × (RBURST[3:0] (read burst length) + 1) data is read from the serial flash memory and the read cache is updated, the data is returned to the bus master.

17.5.7.4 Data Replacement

The write pointer is used to update data. In case of read-miss, the RBURST[3:0] (read burst length) portion data is replaced starting at the entry specified by the write pointer. In other words, the data is replaced in the storage order of the data. Whether data is referred to or not will not affect the replacement order of data.

17.5.8 SPI Operating Mode

This module can carry out an arbitrary SPI operation by using the register settings.

The transfer format is determined based on the common control register (CMNCR), SSL delay register (SSLDR), bit rate setting register (SPBCR), SPI mode control register (SMCR), SPI mode command setting register (SMCMR), SPI mode address setting register (SMADR), SPI mode option setting register (SMOPR), and SPI mode enable setting register (SMENR), SPI mode read data register (SMRDR), SPI mode write data register (SMWDR), SPI mode dummy cycle setting register (SMDMCR), and SPI mode DDR enable register (SMDRENR). For the address, option data, and transfer data, either SDR or DDR transfer can be selected using the appropriate register when the SPBCLK frequency division ratio is two or larger.

SPI operating mode can be used for reading the status of the serial flash memory and writing to the serial flash memory.

In this mode, one transfer refers to the operation from when the SPIE bit in SMCR is set to 1 when the TEND bit in CMNSR is set to 1.

17.5.8.1 Transfer Start

The transfer of data is started in the set transfer format by setting the SPIE bit in SMCR to 1. When write operation is enabled, the SPI mode write data register is transmitted to the serial flash memory. When read operation is enabled, data read from the serial flash memory is stored into the SPI mode read data register.

The SPI operation timing is shown in **Figure 17.14**.

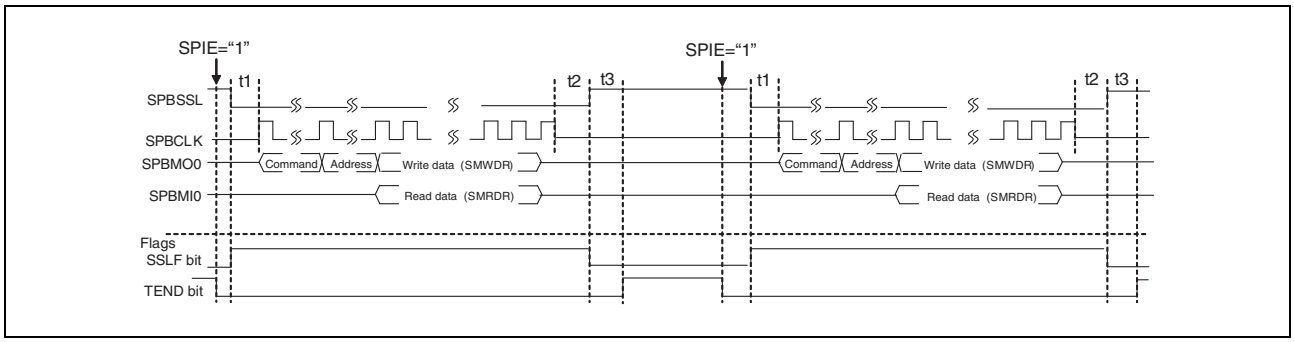


Figure 17.14 SPI Operation Timing

17.5.8.2 Read/Write Enable

- Read operation: Data can be read by setting the SPIRE bit in SMCR to 1. The read data is stored into SMRDR.
- Write operation: Data can be written by setting the SPIWE bit in SMCR to 1. The data stored in SMWDR is output.

When the data size is set to 1 bit using the SPIDB[1:0] bits in SMENR, data can be transmitted and received by setting the SPIRE and SPIWE bits to 1. However, when the data size is set to 2 or 4 bits by using the SPIDB[1:0] bits, only one of the SPIRE and SPIWE bits should be enabled. The operation is not guaranteed if both the bits are enabled.

17.5.8.3 Retention of SPBSSL Pin Assertion

By setting the SSLKP bit in SMCR to 1, assertion of the SPBSSL pin can be continued till the next transfer. With this function, the transfer can be carried out continuously with the SPBSSL kept in the asserted state.

The data transfer timing using the SSLKP bit is shown in **Figure 17.15**.

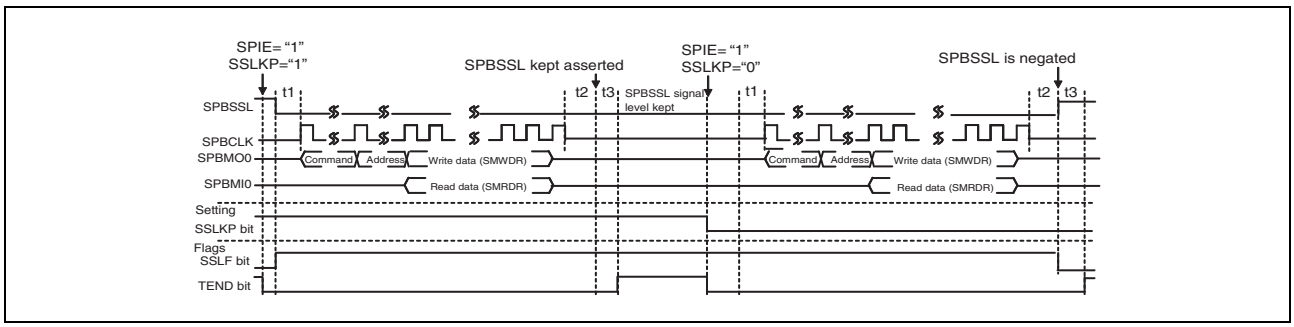


Figure 17.15 Data Transfer Timing using the SSLKP Bit

17.5.8.4 Initial Setting Flow

An example of an initial setting flow in SPI operating mode is shown in **Figure 17.16**.

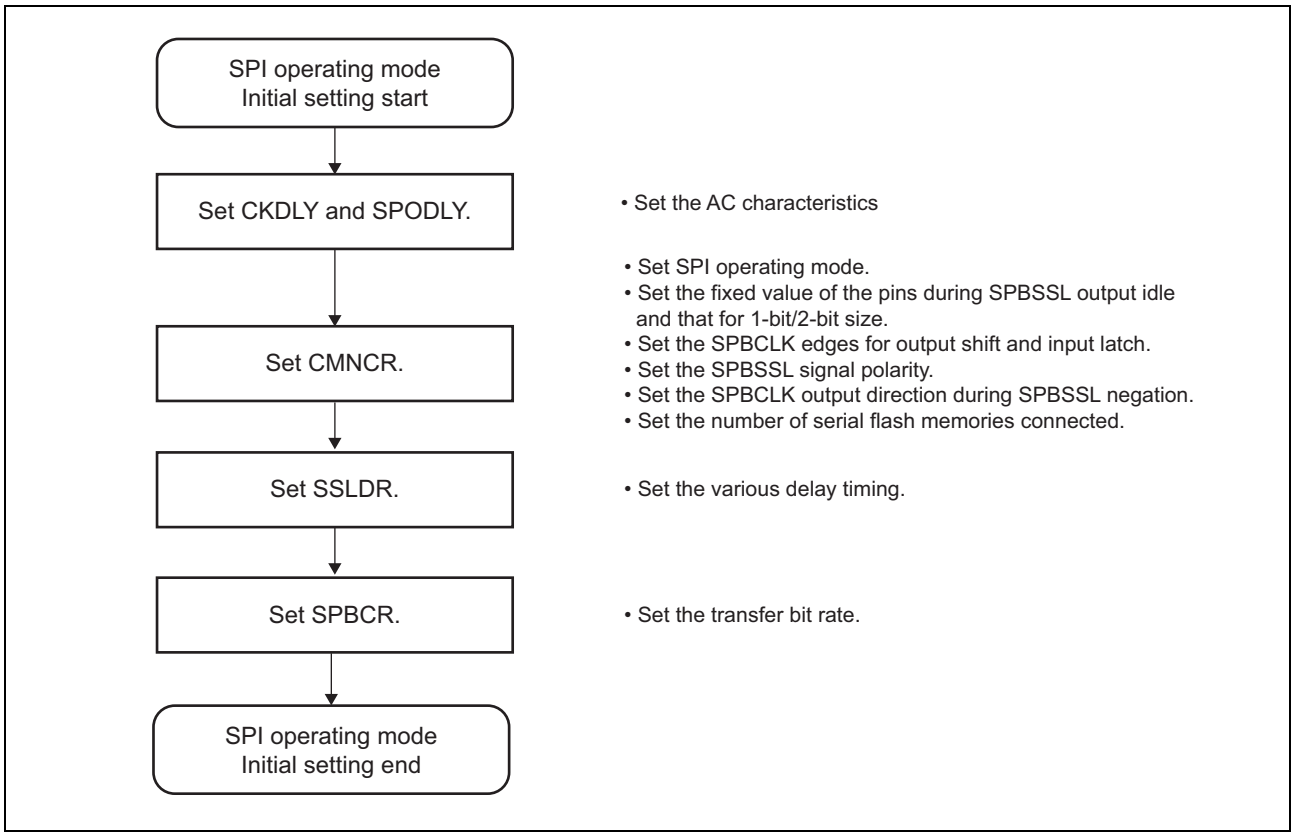


Figure 17.16 Example of Initial Setting Flow in SPI Operating Mode

17.5.8.5 Data Transfer Setting Flow

An example of a data transfer setting flow in SPI operating mode is shown in **Figure 17.17**.

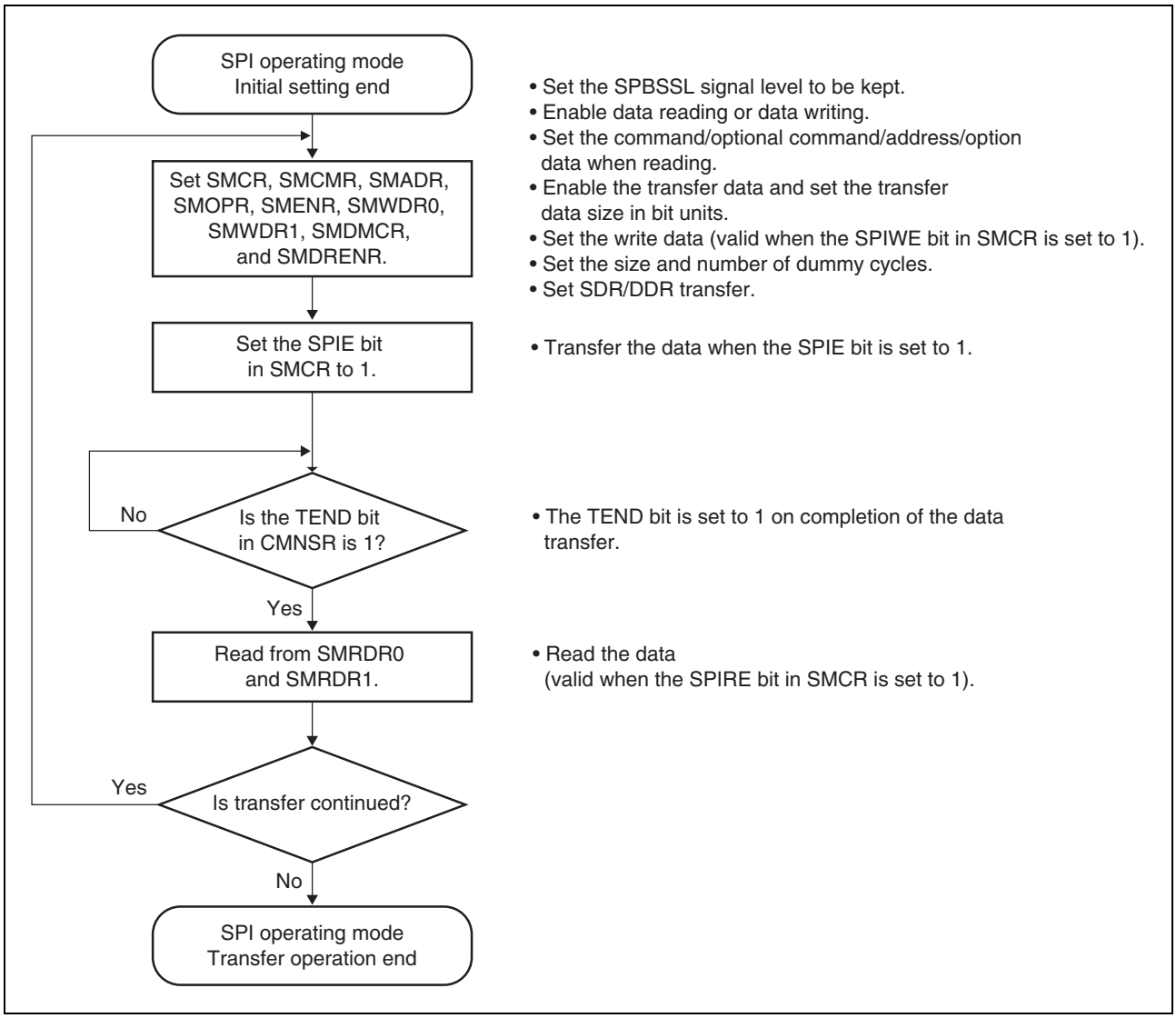


Figure 17.17 Example of a Data Transfer Setting Flow in SPI Operating Mode

17.5.9 Transfer Format

17.5.9.1 SPBSSL Pin Enable Polarity Control

The enable polarity of the SPBSSL pin can be changed with the SSLP bit in CMNCR.

17.5.9.2 SPBCLK Output

The SPBCLK output direction during SPBSSL negation can be set with the CPOL bit in CMNCR.

17.5.9.3 Data Transmission and Reception Timing

Data transmission and reception timing is different between SDR transfer and DDR transfer.

During SDR transfer, data is transmitted and received at either the odd or even edges. The data transmission timing can be set to the odd or even edge with the CPHAT bit in CMNCR. Similarly, the data reception timing can be set to the odd or even edge with the CPHAR bit in CMNCR.

During DDR transfer, data is transmitted and received at both the odd and even edges. The first data transmission timing can be set to the odd or even edge with the CPHAT bit in CMNCR. Similarly, the first data reception timing can be set to the odd or even edge with the CPHAR bit in CMNCR.

17.5.9.4 Delay Settings

t1 is the time period from SPBSSL pin assertion to SPBCLK oscillation (clock delay). It can be set with the SCKDL[2:0] bits in SSLDR. t2 is the time period till the SPBSSL signal negation after the SPBCLK oscillation is stopped (SPBSSL negation delay). It can be set with the SLNDL[2:0] bits in SSLDR. t3 is the time period required to prevent SPBSSL signal assertion for the next transfer after the end of the previous transfer (next access delay). It can be set with the SPNDL[2:0] bits in SSLDR.

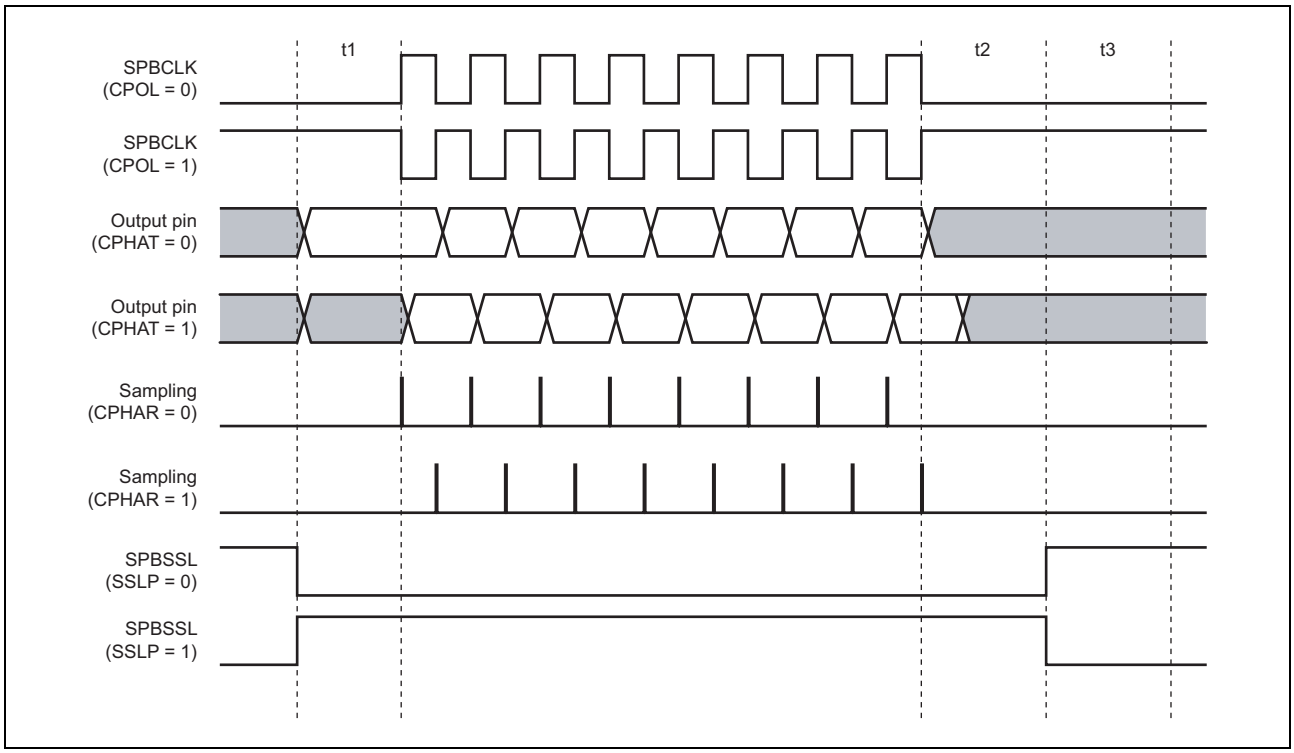


Figure 17.18 SDR Transfer Format

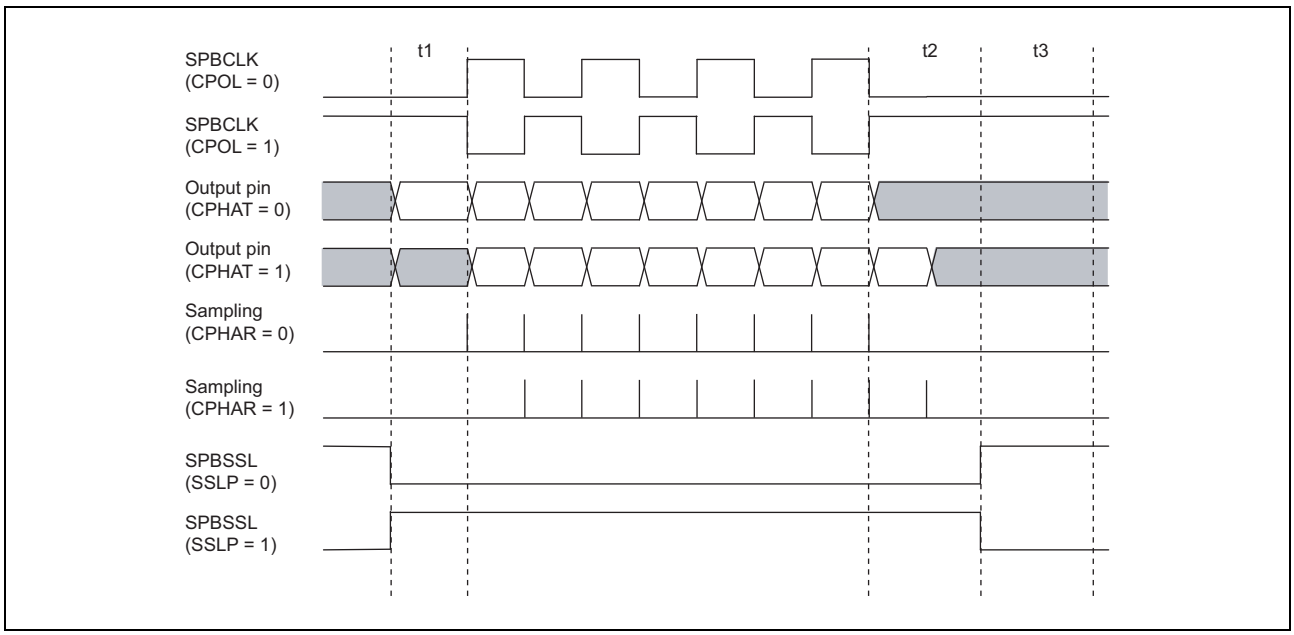


Figure 17.19 DDR Transfer Format

NOTE

In DDR reception when CPHAR = 1, the sampling timing of the last bit is based on the frequency-divided clock in this module.
When CPHAT = 1 in DDR transfer, the serial flash memory cannot provide the sampling timing of the last bit; therefore, a transfer ending with a DDR transfer is not performed correctly when CPHAT = 1.

17.5.10 Data Format

This module can input and output data in the order of command, optional command, address, option data, dummy cycle and data.

17.5.10.1 Data Registers

Table 17.8 shows the input and output data.

Table 17.8 Data Registers

Data		External Address Space Read Operation	SPI Operation
Command (8 bits)		CMD[7:0] bits in DRCMR	CMD[7:0] bits in SMCMR
Optional command (8 bits)		OCMD[7:0] bits in DRCMR	OCMD[7:0] bits in SMCMR
Address (32/24 bits)	BSZ[1:0] = 00 (one flash memory connected)	32 bits: DREAR.EAV[6:1 to 0] bits + lower [25 to 24:0] bits of the read address. 24 bits: Lower [23:0] bits of the read address	32 bits: ADR[31:0] bits in SMADR 24 bits: ADR[23:0] bits in SMADR
	BSZ[1:0] = 01 (two flash memories connected)	32 bits: DREAR.EAV[7:1 to 0] bits + lower [25 to 24:1] bits of the read address. 24 bits: Lower [24:1] bits of the read address	
Option data (8 bits × 4)		DROPR	SMOPR
Dummy cycle (1 to 8 cycles)		DRDMCR	SMDMCR (only when read)
Transfer data		Normal read: 8, 16, and 32 bits Burst read: 64 × (RBURST[3:0] bits + 1)	Read: SMRDR0, SMRDR1 Write: SMWDR0, SMWDR1

17.5.10.2 Data Enable

In external address space read mode, transfer enable or disable of the command, optional command, address, option data, and dummy cycle can be controlled with the CDE, OCDE, ADE[3:0], OPDE[3:0], and DME bits in DRENR, respectively. The size and number of dummy cycles can be controlled with the data read dummy cycle setting register (DRDMCR). When the SPBCLK frequency division ratio is two or larger, either SDR or DDR transfer can be selected for the address, option data, and read data, using the ADDRE, OPDRE, and DRDRE bits in the data read DDR enable register (DRDRENR).

Similarly, in SPI operating mode, enable or disable of the command, optional command, address, option data, dummy cycle, and transfer data can be controlled with the CDE, OCDE, ADE[3:0], OPDE[3:0], DME, and SPIDE[3:0] bits in SMENR, respectively. However, disabling all the above parameters is prohibited in SPI operating mode. At least one of them except dummy cycle must be enabled. The size and number of dummy cycles can be controlled with the SPI mode dummy cycle setting register (SMDMCR). When the SPBCLK frequency division ratio is two or larger, either SDR or DDR transfer can be selected for the address, option data, and transfer data, using the ADDRE, OPDRE, and SPIDRE bits in the SPI mode DDR enable register (SMDRENR).

For the address and option data in external address space read mode; and the address, option data, and transfer data in SPI operating mode, the enable bit setting allowed is determined according to the transfer data size. For the allowed setting combinations of the enable bits and transfer data size, refer to the description of the pertinent registers.

If data is disabled, that data is skipped, and input and output of the next data is carried out. The command, optional command, address, and option data are always output. During dummy cycles, the state of the used pins is Hi-Z. In external address space read mode, data is always input; and in SPI operating mode, input and output of data is determined based on the settings of the SPIRE and SPIWE bits in SMCR.

There are some restrictions on dummy cycle insertion; refer to the description of the DME bits in DRENR and SMENR for details.

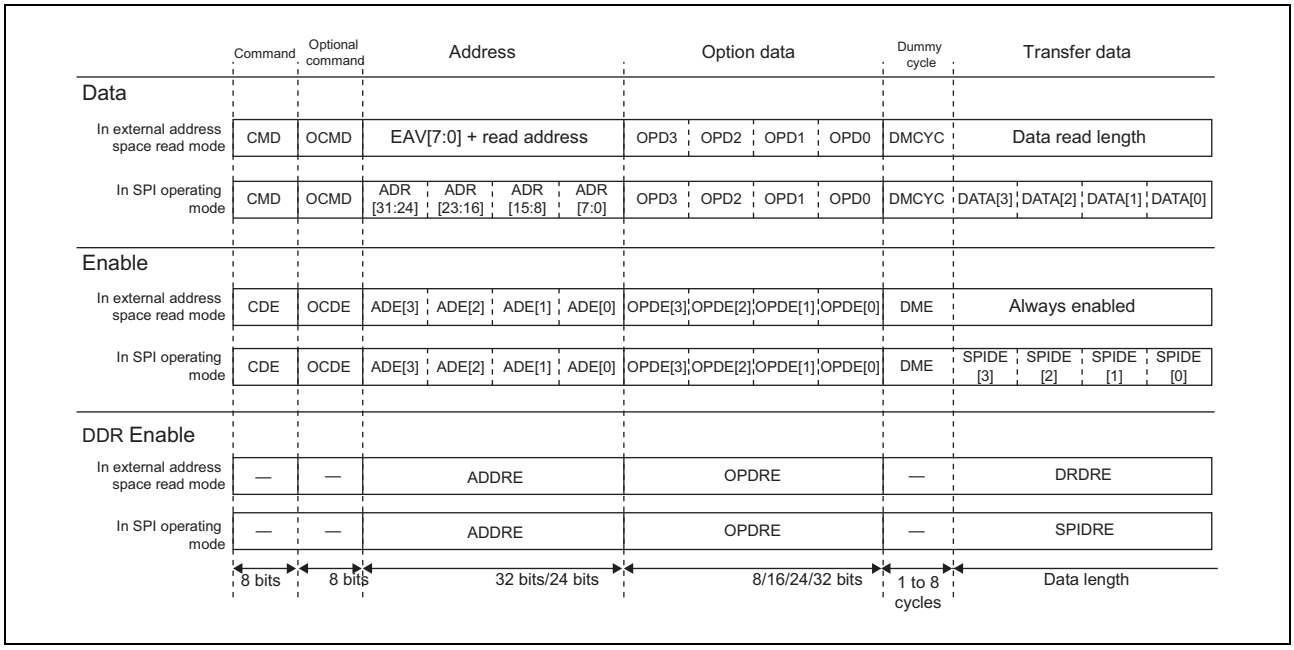


Figure 17.20 Data and Enable

17.5.10.3 Bit Size

In external address space read mode, the size of the command, optional command, address, option data, and the read data in bit units is respectively controlled with the CDB[1:0], OCDB[1:0], ADB[1:0], OPDB[1:0], and DRDB[1:0] bits in DRENR. The size of the dummy cycle in bit units is also controlled with the DMDB[1:0] bits in DRDMCR.
Similarly, in SPI operating mode, the size of the command, optional command, address, option data, and read write data in bit units is controlled with the CDB[1:0], OCDB[1:0], ADB[1:0], OPDB[1:0], and SPIDB[1:0] bits in SMENR. The size of the dummy cycle in bit units is also controlled with the DMDB[1:0] bits in SMDMCR.

(1) 1-bit Size

When the size is set to 1 bit, SPBMI0 and SPBMI1 pins will be the input pins and SPBMO0 and SPBMO1 pins will be the output pins. SPBIO20, SPBIO21, SPBIO30, and SPBIO31 pins are not used.

Figure 17.21 and **Figure 17.22** show the transfer format examples.

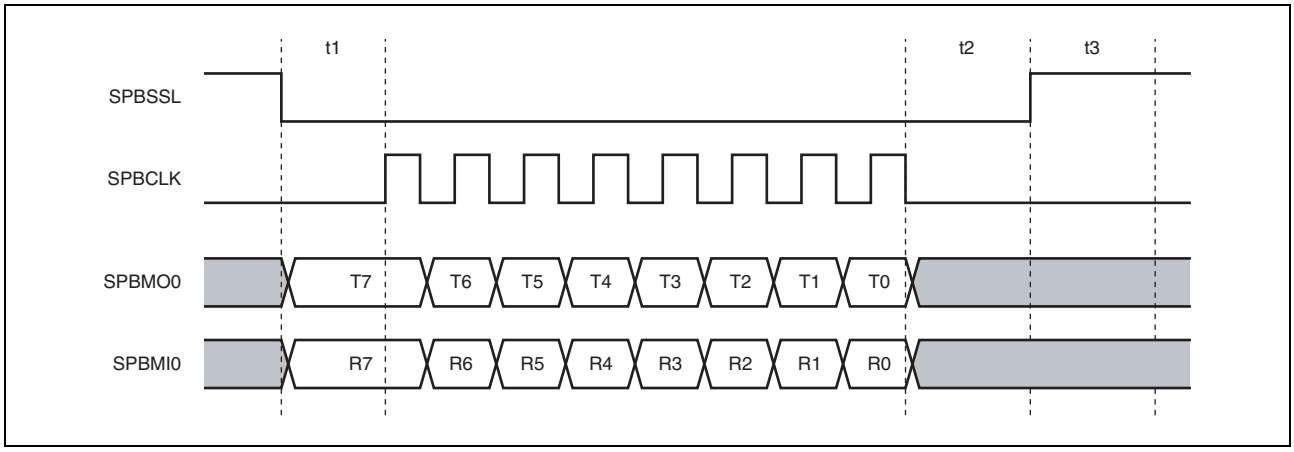


Figure 17.21 Transfer Format Example with 1-Bit Data Size and One Serial Flash Memory Connected

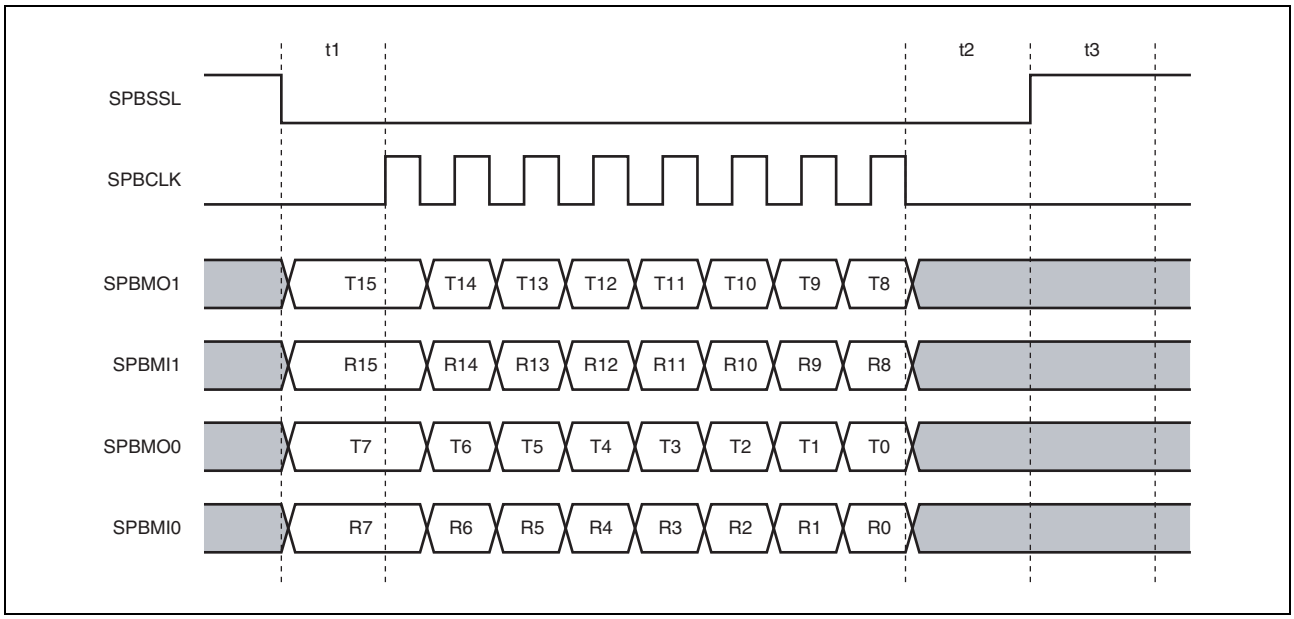


Figure 17.22 Transfer Format Example with 1-Bit Data Size and Two Serial Flash Memories Connected

(2) 2-bit Size

When the size is set to 2 bits, SPBIO00, SPBIO01, SPBIO10, and SPBIO11 pins will be either the input pins or the output pins. SPBIO20, SPBIO21, SPBIO30, and SPBIO31 pins are not used.

Figure 17.23 and **Figure 17.24** show the transfer format examples.

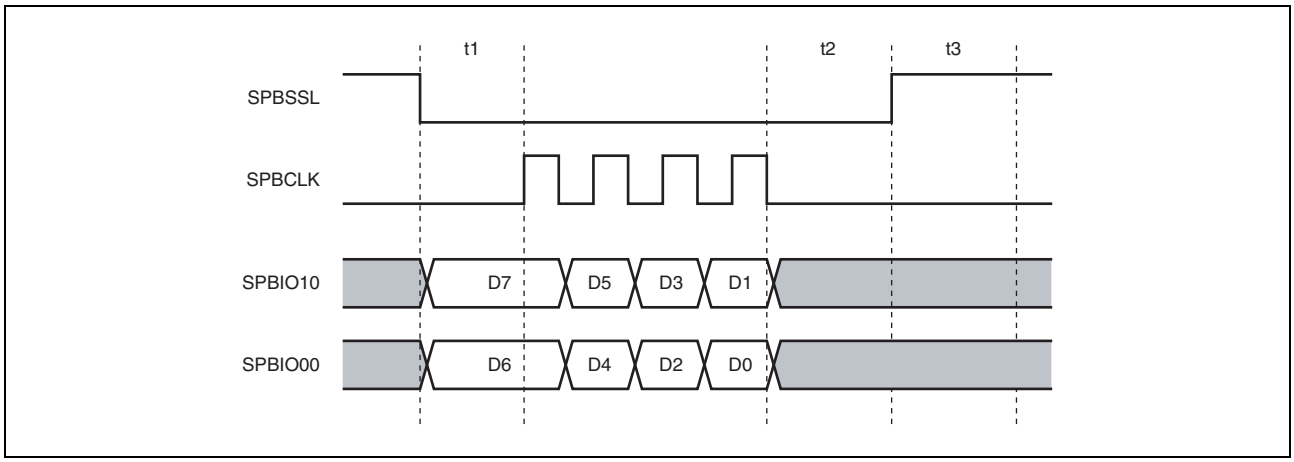


Figure 17.23 Transfer Format Example with 2-Bit Data Size and One Serial Flash Memory Connected

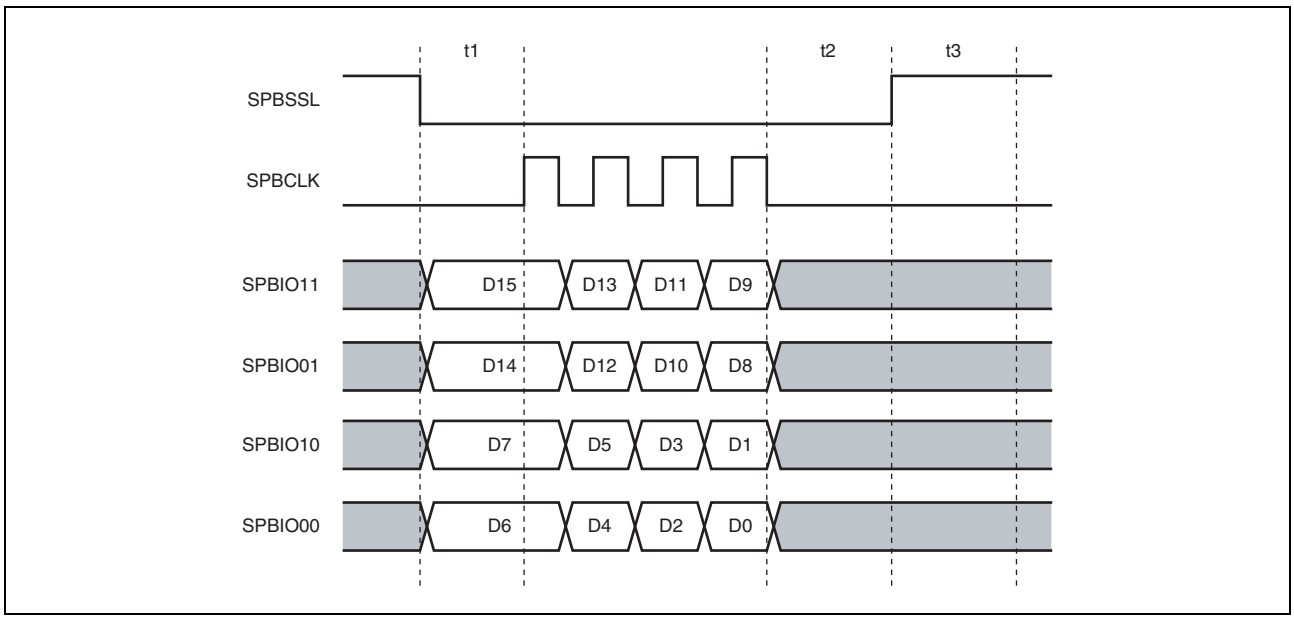


Figure 17.24 Transfer Format Example with 2-Bit Data Size and Two Serial Flash Memories Connected

(3) 4-bit Size

When the size is set to 4 bits, SPBIO00, SPBIO01, SPBIO10, SPBIO11, SPBIO20, SPBIO21, SPBIO30, and SPBIO31 pins will be either the input pins or the output pins.

Figure 17.25 and **Figure 17.26** show the transfer format examples.

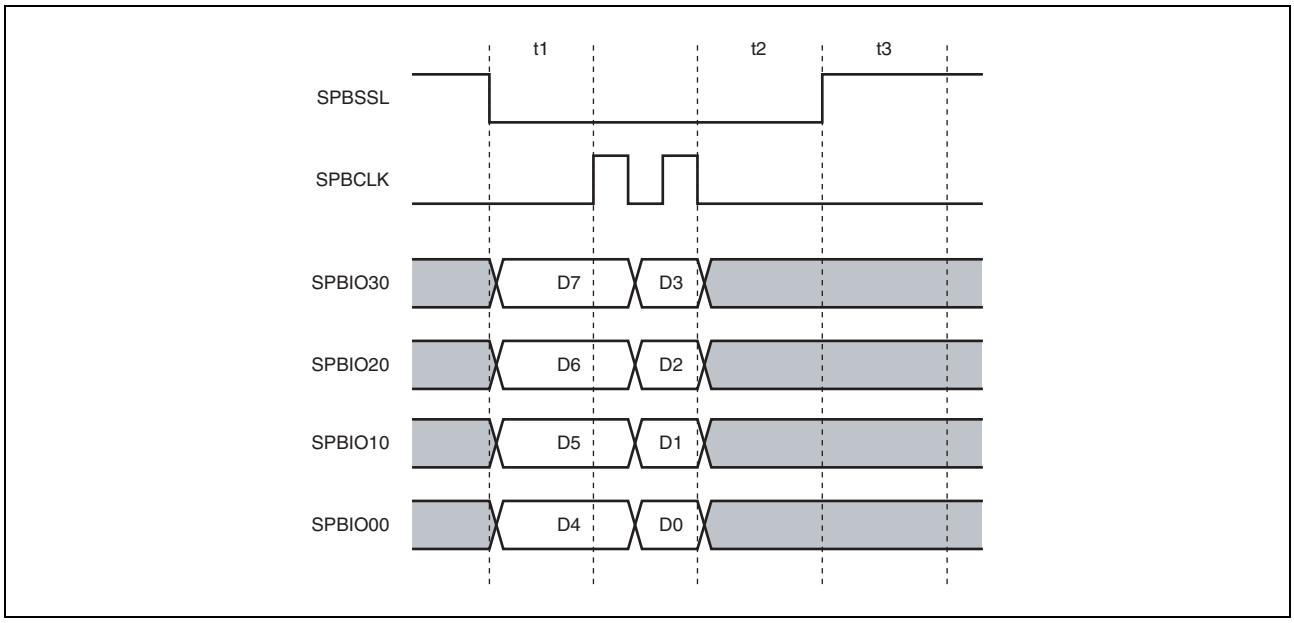


Figure 17.25 Transfer Format Example with 4-Bit Data Size and One Serial Flash Memory Connected

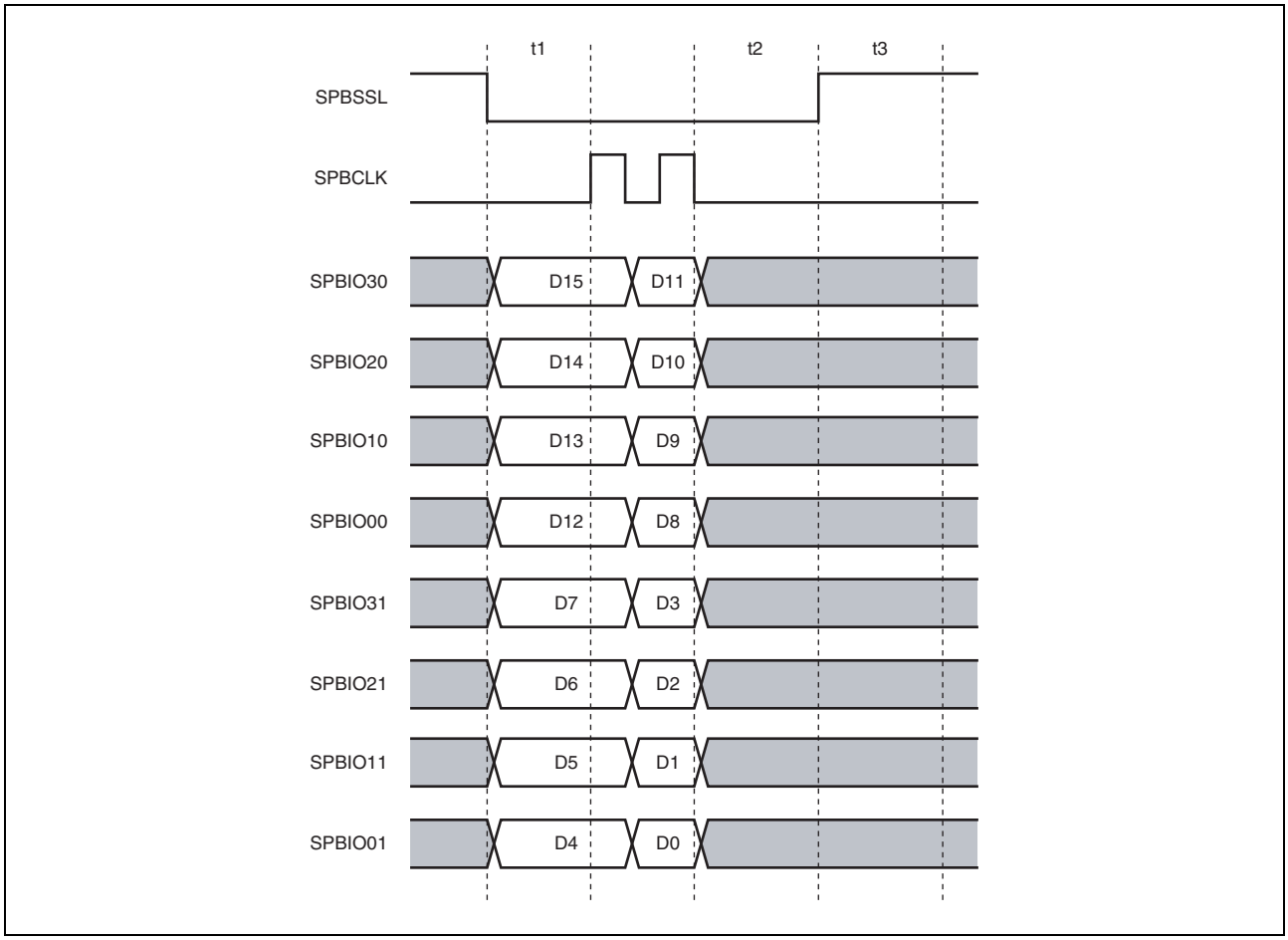


Figure 17.26 Transfer Format Example with 4-Bit Data Size and Two Serial Flash Memories Connected

17.5.11 Data Pin Control

With this module, the status of pins can be automatically changed based on the data size to be used and the read/write settings. The pin status during the SPBSSL negation can be set with the MOIIO3[1:0], MOIIO2[1:0], MOIIO1[1:0], and MOIIO0[1:0] bits in CMNCR. The SPBSSL and SPBCLK pins are always output pins. The status of respective pins is specified in **Table 17.9** to **Table 17.12**.

Table 17.9 Pin Status (1)

		SPBSSL Assertion		
		Command, Optional Command, Address, Option Data		
Pin	**SPBSSL Negation**	**1-bit Size**	**2-bit Size**	**4-bit Size**
SPBMO0/ SPBIO00, SPBMO1/ SPBIO01	MOIIO0[1:0] bit value	Output	Output	Output
SPBMI0/ SPBIO10, SPBMI1/ SPBIO11	MOIIO1[1:0] bit value	Hi-Z	Output	Output
SPBIO20, SPBIO21	MOIIO2[1:0] bit value	IO2FV[1:0] bit value	IO2FV[1:0] bit value	Output
SPBIO30, SPBIO31	MOIIO3[1:0] bit value	IO3FV[1:0] bit value	IO3FV[1:0] bit value	Output

Table 17.10 Pin Status (2)

	Transfer Data					
	External Address Space Read Operation			SPI Operation		
				SPIRE Bit = 1, SPIWE Bit = 0		
Pin	1-bit Size	2-bit Size	4-bit Size	1-bit Size	2-bit Size	4-bit Size
SPBMO0/ SPBIO00, SPBMO1/ SPBIO01	IO0FV[1:0] bit value	Input	Input	IO0FV[1:0] bit value	Input	Input
SPBMI0/ SPBIO10, SPBMI1/ SPBIO11	Input	Input	Input	Input	Input	Input
SPBIO20, SPBIO21	IO2FV[1:0] bit value	IO2FV[1:0] bit value	Input	IO2FV[1:0] bit value	IO2FV[1:0] bit value	Input
SPBIO30, SPBIO31	IO3FV[1:0] bit value	IO3FV[1:0] bit value	Input	IO3FV[1:0] bit value	IO3FV[1:0] bit value	Input

Table 17.11 Pin Status (3)

	Transfer Data					
	SPI Operation					
	SPIRE Bit = 0, SPIWE Bit = 1			SPIRE Bit = 1, SPIWE Bit = 1		
Pin	1-bit Size	2-bit Size	4-bit Size	1-bit Size	2-bit Size	4-bit Size
SPBMO0/ SPBIO00, SPBMO1/ SPBIO01	Output	Output	Output	Output	Setting prohibited	Setting prohibited
SPBMI0/ SPBIO10, SPBMI1/ SPBIO11	Hi-Z	Output	Output	Input	Setting prohibited	Setting prohibited
SPBIO20, SPBIO21	IO2FV[1:0] bit value	IO2FV[1:0] bit value	Output	IO2FV[1:0] bit value	Setting prohibited	Setting prohibited
SPBIO30, SPBIO31	IO3FV[1:0] bit value	IO3FV[1:0] bit value	Output	IO3FV[1:0] bit value	Setting prohibited	Setting prohibited

Table 17.12 Pin Status (4)

	Dummy Cycle		
Pin	1-bit Size	2-bit Size	4-bit Size
SPBMO0/ SPBIO00, SPBMO1/ SPBIO01	IO0FV[1:0] bit value	Hi-Z	Hi-Z
SPBMI0/ SPBIO10, SPBMI1/ SPBIO11	Hi-Z	Hi-Z	Hi-Z
SPBIO20, SPBIO21	IO2FV[1:0] bit value	IO2FV[1:0] bit value	Hi-Z
SPBIO30, SPBIO31	IO3FV[1:0] bit value	IO3FV[1:0] bit value	Hi-Z

17.5.12 SPBSSL Pin Control

Negation conditions of the SPBSSL pin are as follows.

17.5.12.1 External Address Space Read Mode

(1) Normal read operation (RBE bit in DRCR = 0)

SPBSSL negated after completing the data transfer and t2 cycle.

(2) Burst read without automatic SPBSSL negation (RBE bit in DRCR = 1, SSLE bit in DRCR = 0)

SPBSSL negated after completing the data transfer and t2 cycle.

(3) Burst read with automatic SPBSSL negation (RBE bit in DRCR = 1, SSLE bit in DRCR = 1)

- SPBSSL negated after t2 cycle when the read address is not continuous with the previously read address
- SPBSSL negated after the SSLN bit in DRCR is set to 1

17.5.12.2 SPI Operating Mode

(1) SPBSSL pin assertion not retained (SSLKP bit in SMCR = 0)

SPBSSL negated after completing the data transfer and t2 cycle.

(2) SPBSSL pin assertion retained (SSLKP bit in SMCR = 1)

SPBSSL not negated.

When to be negated, data should be transferred after setting the SSLKP bit to 0.

17.5.13 Flags

This module has two flag bits SSLF and TEND in CMNSR. These bits are read-only bits.

17.5.13.1 SSLF Bit

This bit indicates the SPBSSL pin status. The status is 1 when the SPBSSL is asserted, and the status is 0 when the SPBSSL is negated.

17.5.13.2 TEND Bit

This bit indicates whether transfer of data is in progress or the transfer of data has ended.

During t1 time period, data transfer, t2 time period, t3 time period, and waiting for read access by burst read and SPBSSL automatic negation, the TEND bit is read as 0 to indicate that the transfer of data is in progress.

When other than the above, the TEND bit is read as 1 to indicate that transfer of data has ended.

17.5.13.3 Register Re-writing Timing

The status of the TEND bit determines the rewritable registers.

The registers which can be written to, except the SSLN bit in DRCR, should be modified when TEND = 1. Read SMRDR0 and SMRDR1 when TEND = 1. CMNSR can always be read.

17.5.14 Adjustment between input/output data and sampling clocks

If the frequency of the SPBCLK output clock exceeds a certain limit, the phase relation between output and input data and the respective clock signals need to be adjusted:

- SPBCLK frequency < 60 MHz:
  - delay between SPBCLK and the output data is set to a fixed value
  - sampling point of input data is not shifted

  Refer to Section 17.5.14.1, Adjustments for SPBCLK frequency < 60 MHz.
- SPBCLK frequency ≥ 60 MHz:
  - delay between SPBCLK and the output data depends on the $f_{PHCLK}/f_{B\Phi}$ relation
  - sampling point of input data depends on the $f_{PHCLK}/f_{B\Phi}$ relation

  Refer to Section 17.5.14.2, Adjustments for SPBCLK frequency ≥ 60 MHz.

NOTE

The phase shift effects described in this section are independent from any other shift or signal polarity settings, like e.g. via the CPHAT, CPHAR, CPOL, SSLP bit settings of the common control register CMNCR.

17.5.14.1 Adjustments for SPBCLK frequency < 60 MHz

(1) Output clock adjustment

In this case the phase relation between SPBCLK and the output data is set fixed delay of minimum 2 ns by the output data delay register:

- SPODLY.DLYOE0 = SPODLY.DLYO0 = 1: 2 ns delay for port 0
- SPODLY.DLYOE1 = SPODLY.DLYO1 = 1: 2 ns delay for port 1

Refer to the Data Sheet for the delay value range.

Further the SPBCLK clock phase shift value in the clock phase shift adjust register CKDLY needs to be disabled:

- CKDLY.CKDLYOC[2:0] = 000_B: no phase shift

(2) Input sampling point calibration

The sampling point of the input data does not need to be calibrated, thus set

- CKDLY.CKDLYRX[2:0] = 000_B: no sampling point shift

17.5.14.2 Adjustments for SPBCLK frequency ≥ 60 MHz

(1) Output clock adjustment

In this case the delay between SPBCLK and the output data depends on the $f_{PHCLK}/f_{B\Phi}$ relation and needs to be adjusted:

- $f_{PHCLK}/f_{B\Phi} = 2$: set CKDLY.CKDLYOC[2:0] = 010_B
- $f_{PHCLK}/f_{B\Phi} = 3$: set CKDLY.CKDLYOC[2:0] = 011_B
- $f_{PHCLK}/f_{B\Phi} = 4$: set CKDLY.CKDLYOC[2:0] = 101_B

NOTE

The $f_{PHCLK}/f_{B\Phi}$ relation is selected via the clock divider register CKSC_ISFMAD_CTL, which allows to select ratios of 2, 3, 4, 6. If f_{PHCLK} is at its maximum of 480 MHz and $f_{PHCLK}/f_{B\Phi} = 6$ is selected, the serial clock SPBCLK is maximum 40 MHz by SPBR[7:0] = 1. Thus the settings described in Section 17.5.14.1, Adjustments for SPBCLK frequency < 60 MHz are to be used.

The fixed delay needs to be disabled:

- SPODLY.DLYOE0 = SPODLY.DLYO0 = 0: no fixed delay for port 0
- SPODLY.DLYOE1 = SPODLY.DLYO1 = 0: no fixed delay for port 1

(2) Input sampling point calibration

The sampling point calibration process optimizes the time of the sampling point of the input data, read from the external flash memory device.

This is achieved by shifting the sampling edge of the SPBCLK clock signal versus the data signals SPBIO. The shift values are defined as parts of the BΦ cycle period.

The sampling point value CKDLY.CKDLYRX[2:0] is used for the calibration. If AC timing between target flash memory device meets with CKDLY.CKDLYRX[2:0] = 000_B setting, fixed setting of CKDLY.CKDLYRX[2:0] = 000_B can be used and the calibration is not needed.

The StepSize of the sampling point shift value is calculated as follows:

$$StepSize = \frac{180°}{f_{PHCLK}/f_{B\Phi}}$$

The maximum number of shift steps MaxSteps, selectable by CKDLY.CKDLYRX[2:0], is

$$MaxSteps = \frac{360°}{StepSize} - 2$$

and thus the maximum sample point shift MaxShift

$$MaxShift = \left(\frac{360°}{StepSize} - 2\right) \bullet StepSize$$

The following table summarizes the clock frequencies and necessary register settings for some examples.

Table 17.13 Clock adjustment settings examples (for f_{PHCLK} = 480 MHz)

$f_{PHCLK}/f_{B\Phi}$	CKDLYOC[2:0]	$f_{B\Phi}$	max. f_{SPBCLK}	StepSize	MaxSteps	CKDLYRX[2:0]	Sampling point
2*1	010_B	240 MHz	120 MHz	90°	2 steps	000_B:	no shift
						001_B:	90° shift
						010_B: MaxShift =	180° shift
3	011_B	160 MHz	80 MHz	60°	4 steps	000_B:	no shift
						001_B:	60° shift
						010_B:	120° shift
						011_B:	180° shift
						100_B: MaxShift =	240° shift
4	101_B	120 MHz	60 MHz	45°	6 steps	000_B:	no shift
						001_B:	45° shift
						010_B:	90° shift
						011_B:	135° shift
						100_B:	180° shift
						101_B:	225° shift
						110_B: MaxShift =	270° shift

Note 1. The $f_{PHCLK}/f_{B\Phi}$ ratio of 2 can only be used for SDR mode operation.

Sampling point calibration procedure

Perform the calibration procedure below before using the external flash memory.

1. Configure the SFMA and the external flash memory devices for data read from the external address space.
2. First select the maximum sample point shift value for the selected clocks via CKDLY.CKDLYRX[2:0].
3. Read predefined data from flash.
   - If the read data is incorrect, decrement CKDLYRX[2:0] to next lower value and read data again.
   - If the read data is correct, decrement current CKDLYRX[2:0] value by 1 finally. Calibration is completed.

   If the read data is already correct with the maximum sampling point shift, increment current CKDLYRX[2:0] value by 1.
   If the CKDLYRX[2:0] value reaches 000_B and read data is correct at this point, set CKDLYRX[2:0] = 000_B as the phase shift value.
   - If Read data correct, previous value(maximum value) is to be used and Calibration is completed.
   - If Read data not correct, CKDLYRX[2:0] value decrement by 1 from maximum sampling point value.

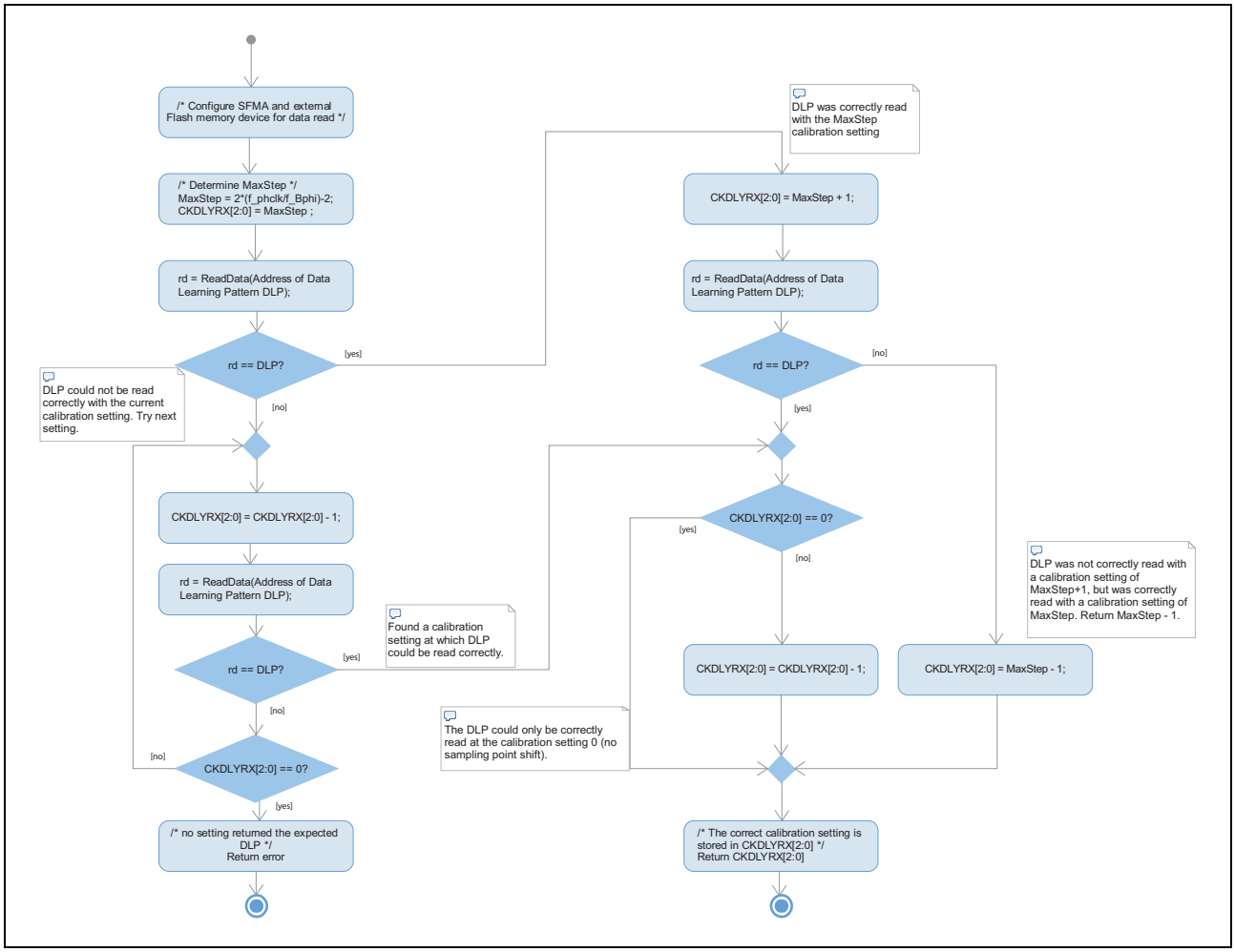


Figure 17.27 Sampling point calibration procedure flow

NOTE

For the calibration the usage of the below data pattern is preferred.

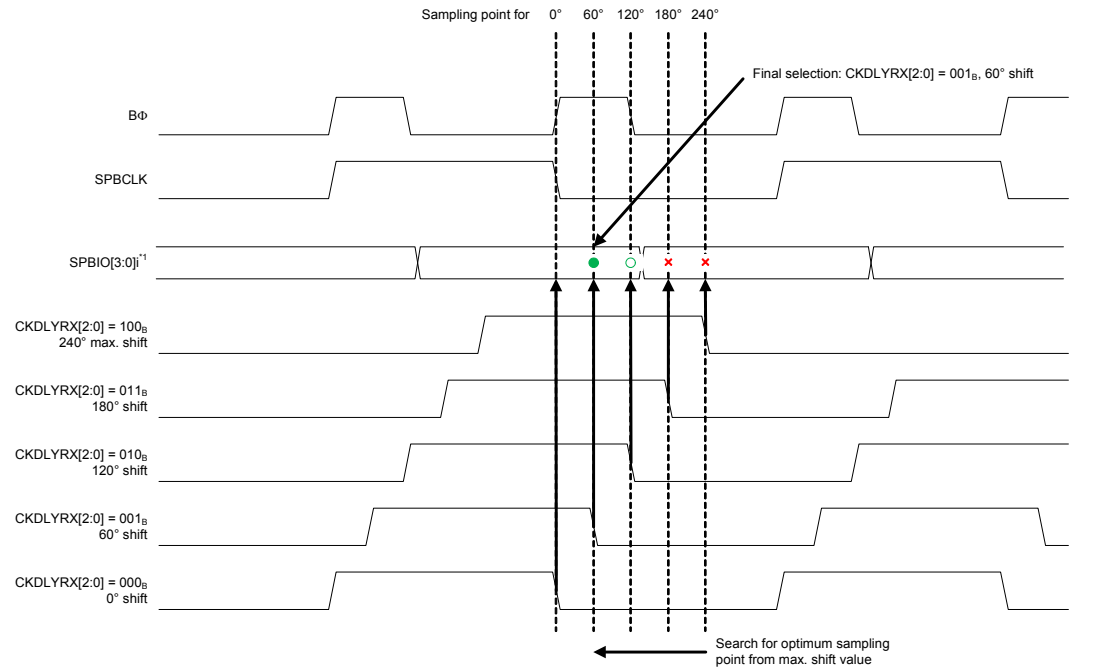
1. 1-bit DDR: 3434_H
2. 1-bit DDR with other Data-Learn Pattern: 6969_H
3. 2-bit DDR with other Data-Learn Pattern: $3CC33CC3_H$
4. 4-bit DDR with other Data-Learn Pattern: $0FF0F00F0FF0F00F_H$
5. 4-bit DDR x 2 device with other Data-Learn Pattern: $0F0FF0F0F0F00F0F0F0FF0F0F0F00F0F_H$

However also the Data-Learn Pattern (DLP) or any other data can be used.

Example: Calibration for f_{PHCLK} = 480 MHz, BΦ = 160 MHz

- start with MaxShift = 240°, i.e. CKDLYRX[2:0] = 100_B
- read data is not correct → CKDLYRX[2:0] decrement to 011_B, i.e. shift by 180°
- read data is not correct → CKDLYRX[2:0] decrement to 010_B, i.e. shift by 120°
- read data is correct → CKDLYRX[2:0] decrement to 001_B, i.e. shift by 60°
- read data is correct. Use CKDLYRX[2:0] value 001_B (i.e. shift by 60°) finally.

The following diagram visualizes this calibration process.



Note 1. i = 0, 1

Figure 17.28 Sampling point calibration

17.6 Usage Notes

17.6.1 Notes on Transfer to Read Data in SPI Operating Mode

If the setting for the bit mode is for division by two or more in SPI operating mode, take note of the following points for caution when setting the SPI mode enable setting register (SMENR) to enable transfer only for reading data.

"Transfer only for reading data" indicates transfer to read data while the CDE, OCDE, ADE[3:0], and OPDE[3:0] bits in SMENR are all 0.

17.6.1.1 Transfer to read data while the signal on the SPBSSL pin is de-asserted

Set the SMENR.SPIDE[3:0] bits to 1100 or 1111 when transfer only for reading data is to proceed.

Transfer will not proceed normally if the setting of the SMENR.SPIDE[3:0] bits is 1000.

17.6.1.2 Transfer to read data while the signal on the SPBSSL pin is asserted

When transfer only for reading data is to proceed, set the SMENR.SPIDE[3:0] bits to 1100 or 1111, or end the immediately preceding transfer with reading data.

When the immediately preceding transfer is of a command, optional command, address, or option data, or is transfer for writing data, the subsequent transfer only for reading data will not proceed normally if the setting of the SMENR.SPIDE[3:0] bits is 1000.

17.6.2 Notes on Starting Transfer from the SPBSSL Retained State in SPI Operating Mode

Be sure to set the SPIWE bit in the SMCR register to 1 when the transfer of a command, optional command, address, or option data is started while the SPBSSL pin is being asserted in SPI operating mode.

17.6.3 Notes on using the SFMA Cache

If the SFMA or its mirror addresses are accessed by CPU in External Address Space Read Mode, special care has to be taken for the XC Cache and the SFMA Cache settings.

Condition:

An XC Cache window has been defined for any address range used by SFMA and CPU accesses SFMA inside of this address window.

Action:

The SFMA Cache has to be activated by setting DRCR.BFM[2:0] = 111_B and DRCR.RBE = 1_B.

For activated SFMA cache, burst length is recommended to be set to DRCR.RBURST[3:0] = 1111_B.

Section 18 Clocked Serial Interface G (CSIG)

This section contains a generic description of the Clocked Serial Interface G (CSIG).

The first part in this section describes all RH850/D1L/D1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the CSIG.

18.1 Features of RH850/D1L/D1M CSIG

18.1.1 Number of Units

This microcontroller has the following number of units of the CSIG.

Each CSIG unit has one channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 18.1 Number of Units

Product Name	All products
Number of Units	4
Name	CSIGn (n = 0 to 3)

Table 18.2 Index

Index	Meaning
n	Throughout this section, the individual CSIG units are identified by the index “n” (n = 0 to 3): for example, CSIGnCTL0 is the CSIGn control register 0.

18.1.2 Register Base Address

CSIG base addresses are listed in the following table.

CSIG register addresses are given as offsets from the base addresses in general.

Table 18.3 Register Base Address

Base Address Name	Base Address
<CSIG0_base>	FFDA 0000_H
<CSIG1_base>	FFDA 2000_H
<CSIG2_base>	FFDA 4000_H
<CSIG3_base>	FFDA 6000_H

18.1.3 Clock Supply

The CSIG clock supply is shown in the following table.

Table 18.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
CSIGn	PCLK	Clock Controller CLKJIT

18.1.4 Interrupt Request

CSIG interrupt requests are listed in the following table.

Table 18.5 Interrupt Requests

CSIGn signals	Function	Connected to
CSIG0:		
INTCSIGTIC	Communication status interrupt	Interrupt Controller INTCSIG0IC DMA Controller trigger ID 95
INTCSIGTIR	Receive status interrupt	Interrupt Controller INTCSIG0IR DMA Controller trigger ID 96
INTCSIGTIRE	Communication error interrupt	Interrupt Controller INTCSIG0IRE
CSIG1:		
INTCSIGTIC	Communication status interrupt	Interrupt Controller INTCSIG1IC DMA Controller trigger ID 97
INTCSIGTIR	Receive status interrupt	Interrupt Controller INTCSIG1IR DMA Controller trigger ID 98
INTCSIGTIRE	Communication error interrupt	Interrupt Controller INTCSIG1IRE
CSIG2:		
INTCSIGTIC	Communication status interrupt	Interrupt Controller INTCSIG2IC DMA Controller trigger ID 99
INTCSIGTIR	Receive status interrupt	Interrupt Controller INTCSIG2IR DMA Controller trigger ID 100
INTCSIGTIRE	Communication error interrupt	Interrupt Controller INTCSIG2IRE
CSIG3:		
INTCSIGTIC	Communication status interrupt	Interrupt Controller INTCSIG3IC DMA Controller trigger ID 101
INTCSIGTIR	Receive status interrupt	Interrupt Controller INTCSIG3IR DMA Controller trigger ID 102
INTCSIGTIRE	Communication error interrupt	Interrupt Controller INTCSIG3IRE

18.1.5 Reset Sources

CSIG reset sources are listed in the following table. CSIG is initialized by these reset sources.

Table 18.6 Reset Sources

Unit Name	Reset Source
CSIGn	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

18.1.6 External Input/Output Signals

External input/output signals of CSIG are listed below.

Table 18.7 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
CSIG0		
CSIGTSCK	Serial clock signal	CSIG0SC*1
CSIGTSI	Serial data input signal	CSIG0SI*1
CSIGTSO	Serial data output signal	CSIG0SO
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG0SSI}}$*1
CSIGTRYI	Ready / busy input signal	CSIG0RYI*1*2
CSIGTRYO	Ready / busy output signal	CSIG0RYO*2
CSIG1		
CSIGTSCK	Serial clock signal	CSIG1SC*1
CSIGTSI	Serial data input signal	CSIG1SI*1
CSIGTSO	Serial data output signal	CSIG1SO
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG1SSI}}$*1
CSIGTRYI	Ready / busy input signal	CSIG1RYI*1*2
CSIGTRYO	Ready / busy output signal	CSIG1RYO*2
CSIG2		
CSIGTSCK	Serial clock signal	CSIG2SC*1
CSIGTSI	Serial data input signal	CSIG2SI*1
CSIGTSO	Serial data output signal	CSIG2SO
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG2SSI}}$*1
CSIGTRYI	Ready / busy input signal	CSIG2RYI*1*2
CSIGTRYO	Ready / busy output signal	CSIG2RYO*2
CSIG3		
CSIGTSCK	Serial clock signal	CSIG3SC*1
CSIGTSI	Serial data input signal	CSIG3SI*1
CSIGTSO	Serial data output signal	CSIG3SO
$\overline{\text{CSIGTSSI}}$	Slave select input signal	$\overline{\text{CSIG3SSI}}$*1
CSIGTRYI	Ready / busy input signal	CSIG3RYI*1*2
CSIGTRYO	Ready / busy output signal	CSIG3RYO*2

Note 1. These input signals are passed through a noise filter, refer to the section “Port Filters” in the section “Port Functions”.

Note 2. For availability of these signals, refer to the Section 2.3.2, List of Alternative Function Pins.

18.1.7 Data Consistency Check

The following table lists the port pins on which CSIGnSO pin functions are multiplexed and data consistency checking is possible. See Section 18.5.10, Error Detection for details on data consistency checking.

Table 18.8 CSIGn data consistency check ports

Unit Signal Name	Port Pin Name	Alternative Function
CSIG0		
CSIGTSO	P1_9	ALT_OUT1
	P45_3	ALT_OUT4
	P42_3	ALT_OUT4
CSIG1		
CSIGTSO	P1_3	ALT_OUT1
	P2_5	ALT_OUT1
	P45_12	ALT_OUT4
CSIG2		
CSIGTSO	P1_6	ALT_OUT1
	P44_2	ALT_OUT4
CSIG3		
CSIGTSO	P2_1	ALT_OUT1
	P44_7	ALT_OUT4
	P42_8	ALT_OUT4
	P42_12	ALT_OUT6

18.2 Overview

18.2.1 Functional Overview

- Three-wire serial synchronous data transfer
- Selection of master mode and slave mode
- Slave select input signal ($\overline{\text{CSIGTSSI}}$) is usable.
- Built-in baud rate generator
- Transfer clock frequency is adjustable in master mode, whereas it is determined by the input clock in slave mode.
- Maximum transfer clock frequency:
  - In master mode: equal to or lower than PCLK/4
  - In slave mode: equal to or lower than PCLK/6
- Selectable clock phase and data phase
- Data transfer selectable from MSB or LSB first
- Transfer data length selectable from 7 to 16 bits in 1-bit units
- Incorporates an EDL (extended data length) function for transferring more than 16 bits of data
- Three selectable transfer modes:
  - transmit-only mode
  - receive-only mode
  - transmit/receive mode
- Built-in handshake function
- Error detection (data consistency check, parity, overrun) included
- Three different interrupt request signals (INTCSIGTIC, INTCSIGTIR, INTCSIGTIRE)
- LBM (Loop Back Mode) function for self test included

18.2.2 Functional Overview Description

The CSIG uses three signals for communication:

- Transmission clock CSIGTSCK (output: in master mode, input: in slave mode)
- Serial data output signal CSIGTSO
- Serial data input signal CSIGTSI

The CSIGnCTL2 register is used to select whether the CSIG should be operated in master mode or slave mode.
In addition, the signals for the external control and monitoring are available.

- CSIGTSSI: Slave select input signal
- CSIGTRYO: Ready/busy output signal (handshake signal)
- CSIGTRYI: Ready/busy input signal (handshake signal)

Data transmission is bit-wise and serial and synchronous to the transmission clock.

The most important registers for setting up the CSIG are:

Table 18.9 Main Registers of CSIG

Register	Function
CSIGnCTL0	Provides and stops operating clock and enables/disables data transmission and data reception.
CSIGnCTL1	Controls options like interrupt timing, extended data length, data consistency check, loop-back mode, handshake, etc.
CSIGnCTL2	Selects master or slave mode, and the transfer clock frequency of the built-in baud rate generator (BRG) in master mode.
CSIGnCFG0	Configures the communication protocol.

18.2.3 Block Diagram

The block diagram shows the main components of the CSIG.

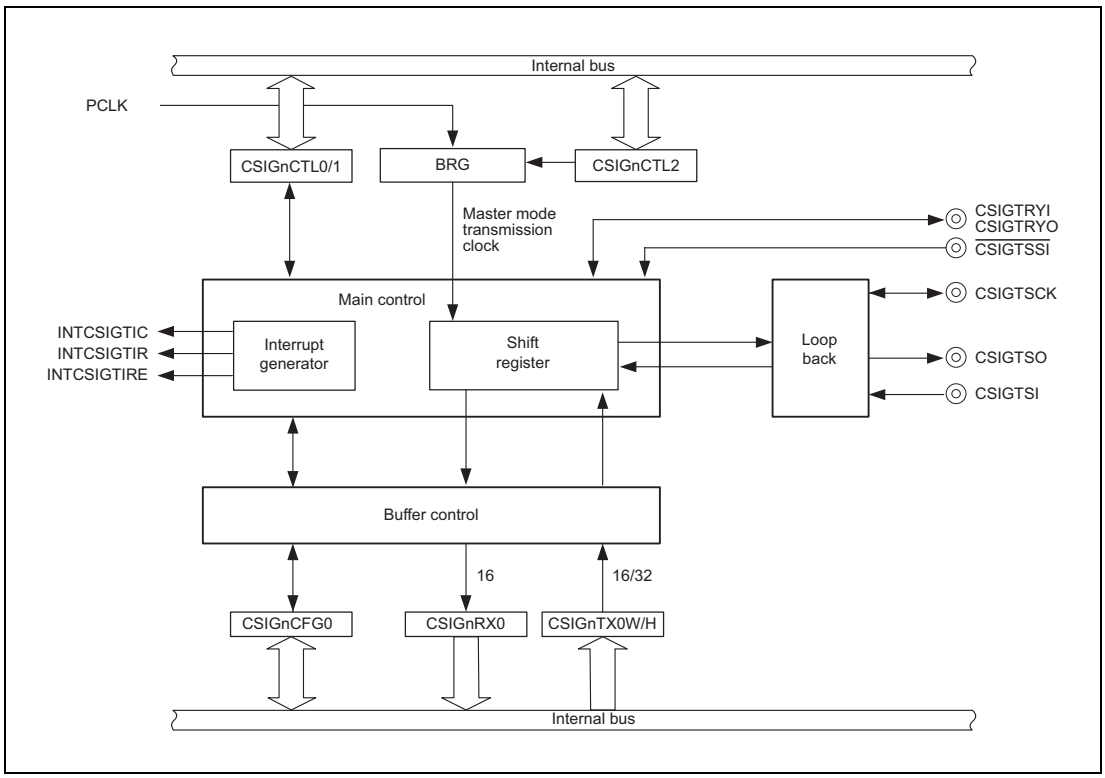


Figure 18.1 CSIG Block Diagram

In master mode, the transmission clock CSIGTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is received from an external source.

18.3 Registers

18.3.1 List of Registers

CSIG registers are listed in the following table.

For details on <CSIGn_base>, see Section 18.1.2, Register Base Address.

Table 18.10 List of Registers

Module Name	Register Name	Symbol	Address
CSIGn	CSIGn control register 0	CSIGnCTL0	<CSIGn_base> + 0000_H
CSIGn	CSIGn control register 1	CSIGnCTL1	<CSIGn_base> + 0010_H
CSIGn	CSIGn control register 2	CSIGnCTL2	<CSIGn_base> + 0014_H
CSIGn	CSIGn status register 0	CSIGnSTR0	<CSIGn_base> + 0004_H
CSIGn	CSIGn status clear register 0	CSIGnSTCR0	<CSIGn_base> + 0008_H
CSIGn	CSIGn Rx-only mode control register 0	CSIGnBCTL0	<CSIGn_base> + 1000_H
CSIGn	CSIGn configuration register 0	CSIGnCFG0	<CSIGn_base> + 1010_H
CSIGn	CSIGn transmission data register 0 for word access	CSIGnTX0W	<CSIGn_base> + 1004_H
CSIGn	CSIGn transmission data register 0 for half word access	CSIGnTX0H	<CSIGn_base> + 1008_H
CSIGn	CSIGn reception data register 0	CSIGnRX0	<CSIGn_base> + $100C_H$
CSIGn	CSIGn emulation register	CSIGnEMU	<CSIGn_base> + 0018_H

NOTE

In the IO header files, availability of each register depends on whether they can be used by each product. The register has no corresponding function terminal has been removed from the IO header file by 1-bit. For availability of terminals, refer to the Section 2.3.2, List of Alternative Function Pins.

18.3.2 CSIGnCTL0 — CSIGn Control Register 0

This register controls the operation clock, and enables or disables transmission/reception.

Access: This register can be read/written in 1-bit and 8-bit units.

Address: <CSIGn_base> + 0000_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CSIGnPWR	CSIGnTXE	CSIGnRXE	—	—	—	—	—*1
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R/W

Note 1. The default value “0” of this bit must be changed to “1” and must not be changed afterwards.

Table 18.11 CSIGnCTL0 Register Contents

Bit Position	Bit Name	Function
7	CSIGnPWR	Controls operation clock. 0: Stops operation clock. 1: Provides operation clock. Clearing CSIGnPWR to 0 resets the internal circuits, stops operation, and sets the CSIG to standby state. No clock is provided to internal circuits. If CSIGnPWR is cleared during communication, ongoing communication is aborted. A restart of the communication setting is then required.
6	CSIGnTXE	Enables/disables transmission. 0: Transmission disabled 1: Transmission enabled
5	CSIGnRXE	Enables/disables reception. 0: Receive disabled 1: Receive enabled
4 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	Bit 0	**CAUTION** The default value “0” of this bit must be changed to “1” and must not be changed afterwards.

CAUTION

When setting this register, see Table 18.24, List of Cautions when Setting Registers.

18.3.3 CSIGnCTL1 — CSIGn Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It enables/disables extended data length control, data consistency check, loop-back mode, handshake function, and slave select function.

Access: This register can be read/written in 32-bit units.

Address: <CSIGn_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CSIGn CKR	CSIGn SLIT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CSIGn EDLE	—	CSIGn DCS	—	CSIGn LBM	CSIGn SIT	CSIGn HSE	CSIGn SSE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 18.12 CSIGnCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 18	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
17	CSIGnCKR	CSIGTSCK clock inversion function 0: Default level of CSIGTSCK is high. 1: Default level of CSIGTSCK is low. The CSIGnCKR bit is used in combination with the CSIGnCFG0.CSIGnDAP bit. For details, see Section 18.3.8, CSIGnCFG0 — CSIGn Configuration Register 0.
16	CSIGnSLIT	Selects the timing of interrupt INTCSIGTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: Interrupt generation when CSIGnTX0W/H is free for next data. For details, see 18.4.2, INTCSIGTIC (Communication Status Interrupt)
15 to 8	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
7	CSIGnEDLE	Enables/disables extended data length (EDL) mode. 0: Extended data length mode disabled 1: Extended data length mode enabled For details, see 18.5.5.2, Data Length Selection with Extended Data Length.
6	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
5	CSIGnDCS	Enables/disables data consistency check. 0: Data consistency check disabled 1: Data consistency check enabled For details, see 18.5.10.1, Data Consistency Check.
4	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
3	CSIGnLBM	Controls loop-back mode (LBM). 0: Loop-back mode deactivated 1: Loop-back mode activated Loop-back mode can be set only in master mode. Set this bit to 0 in slave mode. For details, see Section 18.5.9, Loop-Back Mode.

Table 18.12 CSIGnCTL1 Register Contents (2/2)

Bit Position	Bit Name	Function
2	CSIGnSIT	Selects interrupt delay mode. 0: No delay 1: Half clock delay for all interrupts This bit is only valid in master mode. In slave mode, no delay is generated. For details, see Section 18.4.1, Interrupt Delay.
1	CSIGnHSE	Enables/disables handshake mode. 0: Handshake function disabled 1: Handshake function enabled For details, see Section 18.5.8, Handshake Function.
0	CSIGnSSE	Enables/disables slave select function. 0: Input signal $\overline{\text{CSIGTSSI}}$ is ignored. 1: Input signal $\overline{\text{CSIGTSSI}}$ is enabled. If the slave select function is not used, this bit must be set to 0 (see also Section 18.5.2, Master/Slave Connections).

Details about CSIGnCTL1.CSIGnSSE:

Table 18.13 Operation of the Slave Select Function during Reception

CSIGnCTL0. CSIGnRXE	CSIGnCTL1. CSIGnSSE	$\overline{\text{CSIGTSSI}}$	Receive Operation
0	—	—	Reception disabled
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

Table 18.14 Operation of the Slave Select Function during Transmission

CSIGnCTL0. CSIGnTXE	CSIGnCTL1. CSIGnSSE	$\overline{\text{CSIGTSSI}}$	Transmit Operation
0	—	—	Transmission disabled
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

CAUTION

When setting this register, see Table 18.24, List of Cautions when Setting Registers.

18.3.4 CSIGnCTL2 — CSIGn Control Register 2

This register selects the communication clock.

Access: This register can be read/written in 16-bit units.

Address: <CSIGn_base> + 0014_H

Value after reset: $E000_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnPRS[2:0]			—	CSIGnBRS[11:0]											
Value after reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.15 CSIGnCTL2 Register Contents

Bit Position	Bit Name	Function
15 to 13	CSIGnPRS [2:0]	Selects the value of the prescaler.
12	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
11 to 0	CSIGnBRS [11:0]	Selects the transfer clock frequency. Settings of the CSIGnBRS[11:0] bits are valid only in master mode. They are ignored in slave mode.

CSIGn PRS2	CSIGn PRS1	CSIGn PRS0	Prescaler Output (PRSOUT)
0	0	0	PCLK (master mode)
0	0	1	PCLK / 2 (master mode)
0	1	0	PCLK / 4 (master mode)
0	1	1	PCLK / 8 (master mode)
1	0	0	PCLK / 16 (master mode)
1	0	1	PCLK / 32 (master mode)
1	1	0	PCLK / 64 (master mode)
1	1	1	External clock via CSIGTSCK (slave mode)

CSIGnBRS [11:0]	Transfer Clock Frequency of CSIGTSCK
0	BRG is stopped
1	PCLK / (2^{α} × 1 × 2)
2	PCLK / (2^{α} × 2 × 2)
3	PCLK / (2^{α} × 3 × 2)
4	PCLK / (2^{α} × 4 × 2)
...	...
4095	PCLK / (2^{α} × 4095 × 2)

Note: α = 0 to 6 (value set by CSIGnPRS[2:0])

CAUTION

When setting this register, see Table 18.24, List of Cautions when Setting Registers.

18.3.5 CSIGnSTR0 — CSIGn Status Register 0

This register indicates the status of the CSIG.

Access: This register can only be read in 32-bit units.

Address: <CSIGn_base> + 0004_H

Value after reset: $0000\ 0010_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CSIGn TSF	—	—	—	CSIGn DCE	—	CSIGn PE	CSIGn OVE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.16 CSIGnSTR0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	CSIGnTSF	Transfer Status Flag 0: Idle state 1: Transmission is in progress or being prepared Setting and clearing of this bit is as follows:
6 to 4	Reserved	When read, the value after reset is returned.
3	CSIGnDCE	Data Consistency Check Error Flag 0: No data consistency check error detected 1: Data consistency check error detected This bit is cleared by writing 1 to CSIGnSTCR0.CSIGnDCEC. When setting to 1 due to data consistency error detection and clearing to 0 by CSIGnSTCR0.CSIGnDCEC occur simultaneously, setting to 1 due to data consistency error detection takes precedence. This bit is initialized when CSIGnCTL0.CSIGnPWR changes from 1 to 0 or from 0 to 1.
2	Reserved	When read, the value after reset is returned.

Master Mode	Set by	Cleared by
Tx-only mode	Writing to transmit register	Within a half clock cycle from the last serial clock edge
Tx/Rx mode		
Rx-only mode	Reading from receive register	

Slave Mode	Set by	Cleared by
Tx-only mode	Writing to transmit register	Within a half clock cycle from the last serial clock edge
Tx/Rx mode		
Rx-only mode	CSIGTSCK input	

Table 18.16 CSIGnSTR0 Register Contents (2/2)

Bit Position	Bit Name	Function
1	CSIGnPE	Parity Error Flag 0: No parity error detected 1: Parity error detected This bit is cleared by writing 1 to CSIGnSTCR0.CSIGnPEC. When setting to 1 due to parity error detection and clearing to 0 by writing to CSIGnSTCR0.CSIGnPEC occur simultaneously, setting to 1 due to parity error detection takes precedence. This bit is initialized by the value of CSIGnCTL0.CSIGnPWR changes from 0 to 1, or from 1 to 0.
0	CSIGnOVE	Overrun Error Flag 0: No overrun error detected 1: Overrun error detected This bit is cleared by writing 1 to CSIGnSTCR0.CSIGnOVEC. When setting to 1 due to overrun error detection and clearing to 0 by writing to CSIGnSTCR0.CSIGnOVEC occur simultaneously, setting to 1 due to overrun error detection takes precedence. This bit is initialized by the value of CSIGnCTL0.CSIGnPWR changes from 0 to 1, or from 1 to 0.

CAUTION

When setting this register, see Table 18.24, List of Cautions when Setting Registers.

18.3.6 CSIGnSTCR0 — CSIGn Status Clear Register 0

This register clears the status flags of the CSIGnSTR0 status register.

Access: This register can be read/written in 16-bit units.
When read, the value 0000_H is always returned.

Address: <CSIGn_base> + 0008_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CSIGn DCEC	—	CSIGn PEC	CSIGn OVEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W

Table 18.17 CSIGnSTCR0 Register Contents

Bit Position	Bit Name	Function
15 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	CSIGnDCEC	Controls the data consistency error flag clear command. 0: No operation. Read value is always 0. 1: Clear data consistency check error flag (CSIGnSTR0.CSIGnDCE).
2	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
1	CSIGnPEC	Controls the parity error flag clear command. 0: No operation. Read value is always 0. 1: Clear parity error flag (CSIGnSTR0.CSIGnPE).
0	CSIGnOVEC	Controls the overrun error flag clear command. 0: No operation. Read value is always 0. 1: Clear overrun error flag (CSIGnSTR0.CSIGnOVE).

18.3.7 CSIGnBCTL0 — CSIGn Rx-Only Mode Control Register 0

This register enables/disables the data transfer in Rx-only mode.

Access: This register can be read/written in 1-bit and 8-bit units.

Address: <CSIGn_base> + 1000_H

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CSIGnSCE
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R/W

Table 18.18 CSIGnBCTL0 Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	CSIGnSCE	Disables/enables next data reception start by reading CSIGnRX0. 0: Next reception disabled 1: Next reception enabled For details, see Section 18.5.4.2, Receive-Only Mode.

CAUTION

When setting this register, see Table 18.24, List of Cautions when Setting Registers.

18.3.8 CSIGnCFG0 — CSIGn Configuration Register 0

This register configures the communication protocol – data length, parity, transfer direction, clock phase, and data phase.

Access: This register can be read/written in 32-bit units.

Address: <CSIGn_base> + 1010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CSIGnPS[1:0]		CSIGnDLS[3:0]				—	—	—	—	—	CSIGn DIR	—	CSIGn DAP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.19 CSIGnCFG0 Register Contents (1/2)

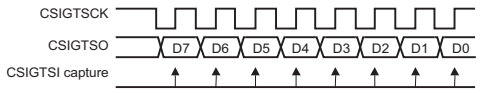
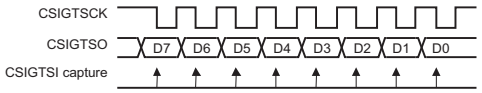
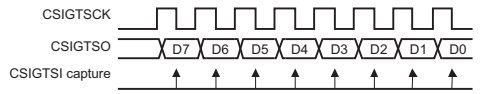
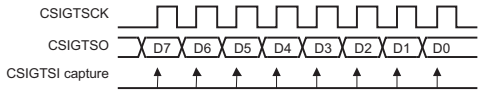
Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
29, 28	CSIGnPS[1:0]	Specifies parity.

CSIGn PS1	CSIGn PS0	Transmission	Reception
0	0	No parity transmitted	No parity is waited for.
0	1	Add parity bit fixed at 0	Parity bit is waited for but not judged.
1	0	Add odd parity	Odd parity bit is waited for.
1	1	Add even parity	Even parity bit is waited for.

27 to 24	CSIGnDLS [3:0]	Specifies data length. 0: Data length is 16 bits 1: Data length is 1 bit 2: Data length is 2 bits ... 15: Data length is 15 bits **CAUTION** Do not set bits CSIGnCFG0.CSIGnDLS[3:0] for a value 1 to 6 when the extended data length function is disabled with bit CSIGnCTL1.CSIGnEDLE set to 0. It is forbidden to transmit two consecutive data with a data length of less than 7 bits.
23 to 19	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
18	CSIGnDIR	Selects the serial data direction. 0: Data is sent/received with MSB first 1: Data is sent/received with LSB first
17	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.

Table 18.19 CSIGnCFG0 Register Contents (2/2)

Bit Position	Bit Name	Function
16	CSIGnDAP	Data Phase Selection Select data phase in line with CSIGnCTL1.CSIGnCKR bit. See below for clock and data phase. .
15 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

CSIGnCTL1.CSIGnCKR	CSIGnDAP	Clock and Data Phase Selection
0	0	
0	1	
1	0	
1	1	

CAUTION

When setting this register, see Table 18.24, List of Cautions when Setting Registers.

18.3.9 CSIGnTX0W — CSIGn Transmission Register 0 for Word Access

This register stores the transmission data and specifies the extended data length.

Access: This register can be read/written in 32-bit units.

Address: <CSIGn_base> + 1004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	CSIGn EDL	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnTX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.20 CSIGnTX0W Register Contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
29	CSIGnEDL	Specifies the extended data length. 0: Normal operation 1: Extended data length enabled The associated data is transmitted as 16-bit data. This bit can only be set if CSIGnCTL1.CSIGnEDLE = 1.
28 to 16	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
15 to 0	CSIGnTX[15:0]	Data to be transmitted

CAUTION

When setting this register, see Table 18.24, List of Cautions when Setting Registers.

18.3.10 CSIGnTX0H — CSIGn Transmission Register 0 for Half Word Access

This register stores the transmission data. This register is the same as bits 15 to 0 of register CSIGnTX0W.
The 16 high-order bits of CSIGnTX0W are applied for transfer.

Access: This register can be read/written in 16-bit units.

Address: <CSIGn_base> + 1008_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnTX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 18.21 CSIGnTX0H Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIGnTX[15:0]	Data to be transmitted

CAUTION

When setting this register, see Table 18.24, List of Cautions when Setting Registers.

18.3.11 CSIGnRX0 — CSIGn Reception Register 0

This register stores the received data.

Access: This register can only be read in 16-bit units.

Address: <CSIGn_base> + $100C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIGnRX[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18.22 CSIGnRX0 Register Contents

Bit Position	Bit Name	Function
15 to 0	CSIGnRX [15:0]	Received Data These bits are initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1 or from 1 to 0. When reading, the values of these bits must be read at least 1 clock before the generation of CSIGTIR interrupt.

CAUTION

When setting this register, see Table 18.24, List of Cautions when Setting Registers.

18.3.12 CSIGnEMU — CSIGn Emulation Register

This register controls operation by SVSTOP.

Access: This register can be read/written in 1-bit and 8-bit units.
Write to this register when EPC.SVSTOP = 0.

Address: <CSIGn_base> + 0018_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CSIGnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 18.23 CSIGnEMU Register Contents

Bit Position	Bit Name	Function
7	CSIGnSVSDIS	Selects to continue or stop transmit/receive operation during debugging. • When the EPC.SVSTOP bit is set to 0: Continues transmit/receive operation regardless of the setting of this bit. • When the EPC.SVSTOP bit is set to 1: 0: Stops transmit/receive operation. 1: Continues transmit/receive operation.
6 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

CAUTION

When setting this register, see Table 18.24, List of Cautions when Setting Registers.

18.3.13 List of Cautions

Table 18.24 List of Cautions when Setting Registers

Register Name	Bit Name	Contents
CSIGnCTL0	CSIGnPWR	If this bit is cleared during communication, ongoing communication is suspended. After the communication is suspended, it is necessary to restart the communication.
CSIGnCTL0	CSIGnTXE CSIGnRXE	Do not modify any of these bits while CSIGnCTL0.CSIGnPWR = 0. (These bits can be modified at the same time with the CSIGnCTL0.CSIGnPWR bit.) Do not modify these bits while CSIGnSTR0.CSIGnTSF = 1, because the specified operation is not guaranteed if ongoing communication is suspended.
CSIGnCTL0	Bit 0	When writing, always write 1. (The value after reset is "0".) This bit must be modified at the same time with CSIGnCTL0.CSIGnPWR bit.
CSIGnCTL1	CSIGnCKR	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0.
CSIGnCTL1	CSIGnSLIT CSIGnEDLE CSIGnDCS CSIGnHSE	Modification of these bits is only permitted while CSIGnCTL0.CSIGnPWR = 0.
CSIGnCTL1	CSIGnLBM	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting of this bit is prohibited in slave mode.
CSIGnCTL1	CSIGnSSE	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting this bit to 1 is prohibited in master mode.
CSIGnCTL1	CSIGnSIT	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. This bit is only valid in master mode. In slave mode, no delay is generated.
CSIGnCTL2	CSIGnPRS[2:0] CSIGnBRS[11:0]	Modification of this bit is only permitted while CSIGnCTL0.CSIGnPWR = 0. Setting of the max transfer clock frequency is as follows. • Master mode: up to PCLK/4 • Slave mode: up to PCLK/6
CSIGnSTR0	CSIGnTSF	Writing to this bit is prohibited, and only reading is permitted.
CSIGnSTR0	CSIGnDCE CSIGnPE CSIGnOVE	Writing to these bits is prohibited, and only reading is permitted. These bits are initialized when CSIGnCTL0.CSIGnPWR = 0 → 1 or CSIGnCTL0.CSIGnPWR = 1 → 0.
CSIGnBCTL0	CSIGnSCE	Write to this bit before CSIGnRX0 is read. Fix the CSIGnSCE bit to 0 when the transfer mode is transmit mode or transmit/receive mode.
CSIGnCFG0	CSIGnPS[1:0] CSIGnDLS[3:0] CSIGnDIR CSIGnDAP	Modification of these bits is only permitted while CSIGnCTL0.CSIGnPWR = 0.
CSIGnTX0W	CSIGnEDL	This bit is valid only when CSIGnCTL1.CSIGnEDLE = 1.
CSIGnTX0W CSIGnTX0H		Write access to these bits are prohibited when CSIGnCTL0.CSIGnTXE = CSIGnCTL0.CSIGnRXE = 0.
CSIGnRX0		This bit is initialized when CSIGnRX0 CSIGnCTL0.CSIGnPWR = 0 → 1 or CSIGnCTL0.CSIGnPWR = 1→ 0. Read access to this bit is prohibited when CSIGnCTL0.CSIGnTXE = CSIGnCTL0.CSIGnRXE = 0.
CSIGnEMU	CSIGnSVSDIS	Modification of this bit is prohibited while SVSTOP = 1.

18.4 Interrupt Sources

CSIG can generate the following interrupts:

- INTCSIGTIC (communication status interrupt)
- INTCSIGTIR (reception status interrupt)
- INTCSIGTIRE (communication error interrupt)

18.4.1 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by one half period of the transmission clock CSIGTSCK. This is not possible in slave mode.

The delay is specified by setting bit CSIGnCTL1.CSIGnSIT = 1. (The setting of the CSIGnSIT bit is invalid in slave mode.)

The following example illustrates the interrupt delay function, assuming a setting of CSIGnCTL1.CSIGnSIT = 1 (interrupt delay enabled), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), and CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits).

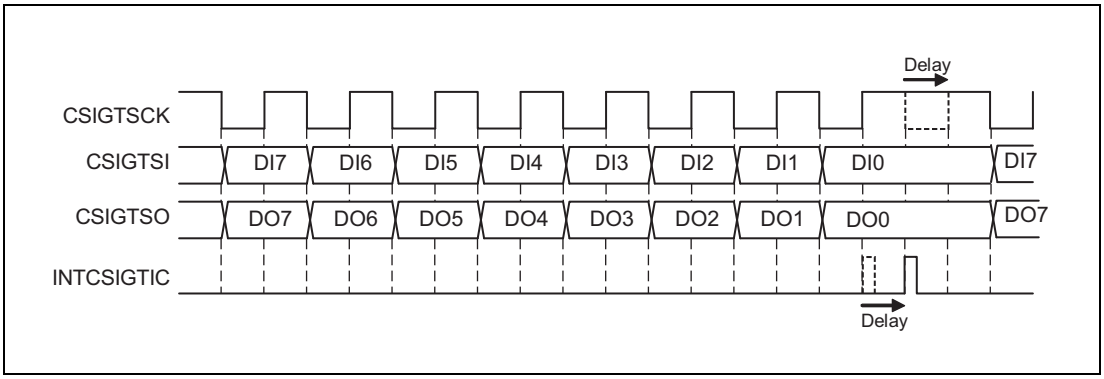


Figure 18.2 Interrupt Delay Function (CSIGnCTL1.CSIGnSIT = 1)

18.4.2 INTCSIGTIC (Communication Status Interrupt)

This interrupt is normally generated after every data transfer. It can be used to trigger a DMA for writing new transmission data to register CSIGnTX0W or CSIGnTX0H.

The following example assumes master mode and a setting of CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits), and CSIGnCTL1.CSIGnSLIT = 0 (normal interrupt timing).

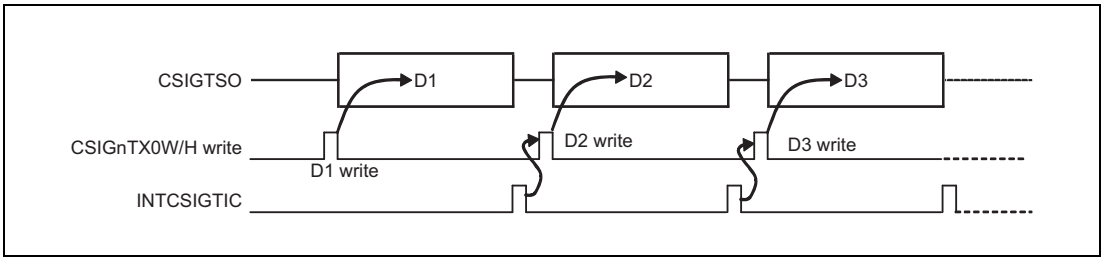


Figure 18.3 Generation of INTCSIGTIC after Communication (CSIGnCTL1.CSIGnSLIT = 0)

However, INTCSIGTIC can also be set up to occur when the CSIGnTX0W/H register is free for the next data. This is specified by setting CSIGnCTL1.CSIGnSLIT = 1.

This mode allows more efficient data transfers.

The effect is illustrated in the figure below.

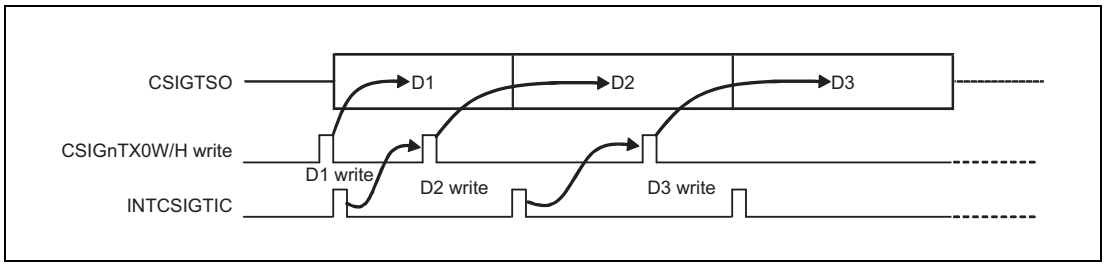


Figure 18.4 Generation of INTCSIGTIC at the Beginning of Communication

18.4.3 INTCSIGTIR (Reception Status Interrupt)

This interrupt is generated in receive-only and transmit/receive mode after data has been received and is available in the reception register. It can be used to trigger a DMA for reading the received data from register CSIGnRX0.

The following example assumes master mode and a setting of CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase), and CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits).

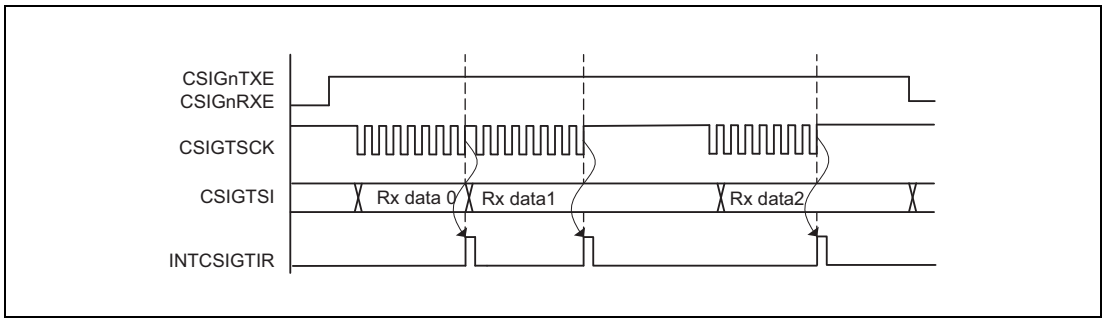


Figure 18.5 Generation of INTCSIGTIR

18.4.4 INTCSIGTIRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

Table 18.25 Data Error Types

Error Type	Communication Status After Error Interrupt
Parity error	Interrupt is generated and communication continues.
Data consistency error	Interrupt is generated and communication continues.
Overrun error	When CSIGnCTL1.CSIGnHSE = 0 (handshake function disabled) in slave mode, interrupt is generated and communication continues.
	When CSIGnCTL1.CSIGnHSE = 1 (handshake function enabled) in slave mode, communication stops due to the handshake. Interrupt is not generated and overrun errors do not occur.

Note 1. In master mode, overrun errors do not occur.
In slave mode, communication cannot be stopped.

The type of error that caused the generation of INTCSIGTIRE is indicated in register CSIGnSTR0.

For details about the various error types, see Section 18.5.10, Error Detection.

18.5 Operation

18.5.1 Master/Slave Mode

CSIG operation in master mode or in slave mode depends on the setting of bits CSIGnCTL2.CSIGnPRS[2:0]. If master mode is selected, the source of the transmission clock must be selected too.

18.5.1.1 Master Mode

In master mode, the serial communication clock is generated by the internal baud rate generator (BRG) and provided by signal CSIGTSCK.

Master mode is enabled by setting bits CSIGnCTL2.CSIGnPRS[2:0] to anything but 111_B. In master mode, the frequency setting of the BRG becomes effective by setting bits CSIGnCTL2. CSIGnPRS[2:0] and bits CSIGnCTL2.CSIGnBRS[11:0].

The default level of CSIGTSCK depends on the clock phase selection bit: it is high when CSIGnCTL1.CSIGnCKR = 0, and is low when CSIGnCTL1.CSIGnCKR = 1.

The example below shows the communication in master mode for 8 data bits, CSIGnCFG0.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

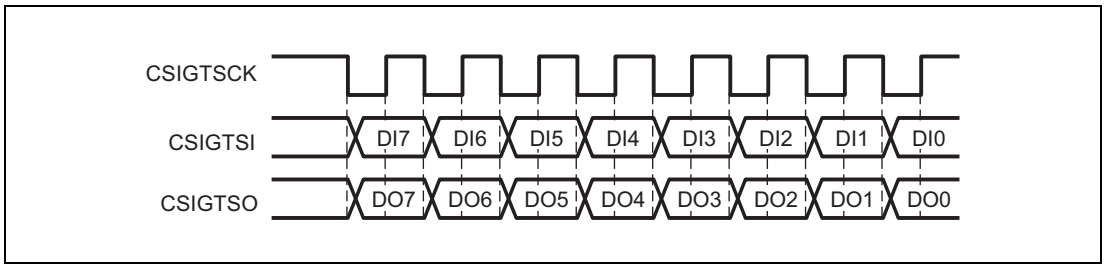


Figure 18.6 Transmission/Reception in Master Mode

18.5.1.2 Slave Mode

In slave mode, another device is the communication master. The external clock is received by signal CSIGTSCK. Send/receive operation starts as soon as a clock signal is detected.

Slave mode is selected by setting CSIGnCTL2.CSIGnPRS[2:0] to 111_B.

NOTE

When using slave mode, disable the baud rate generator (BRG) by setting bits CSIGnCTL2.CSIGnBRS[11:0] to 000_H.

The example below shows the communication in slave mode for 8 data bits, CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

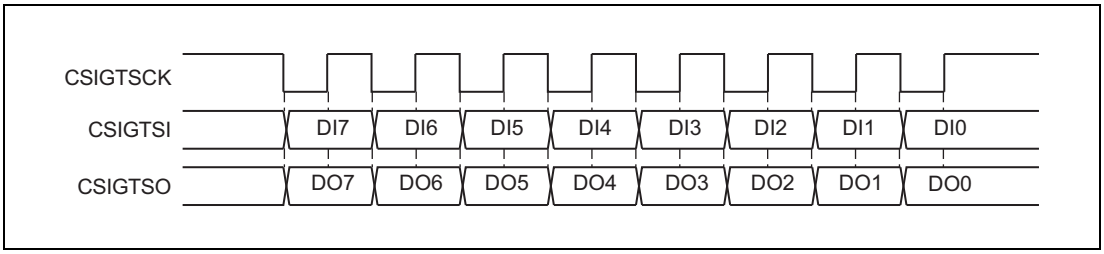


Figure 18.7 Transmission/Reception in Slave Mode

18.5.2 Master/Slave Connections

18.5.2.1 One Master and One Slave

The following figure illustrates the connections between one master and one slave.

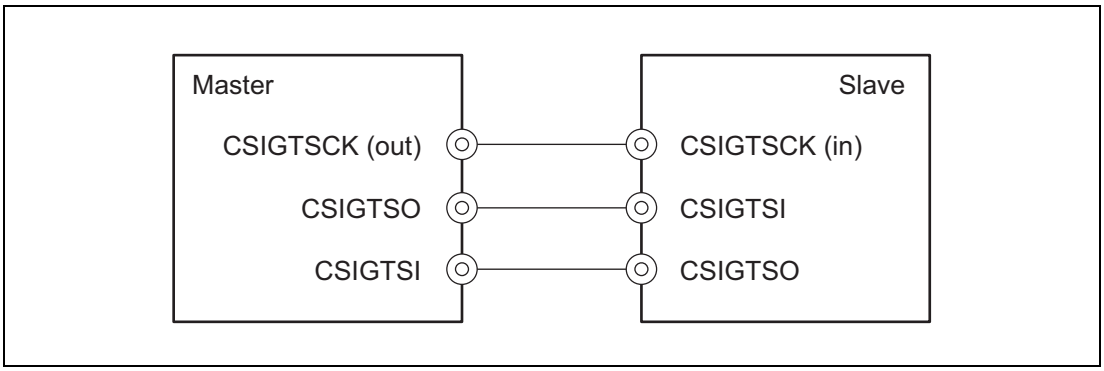


Figure 18.8 Direct Master/Slave Connection

18.5.2.2 One Master and Multiple Slaves

The following figure illustrates the connections between one master and multiple slaves. In this case, the master must provide one slave select (SS) signal to each of the slaves. This signal is connected to the slave select input $\overline{\text{CSIGTSSI}}$ of the slave.

The $\overline{\text{CSIGTSSI}}$ signal can be enabled/disabled by bit CSIGnCTL1.CSIGnSSE.

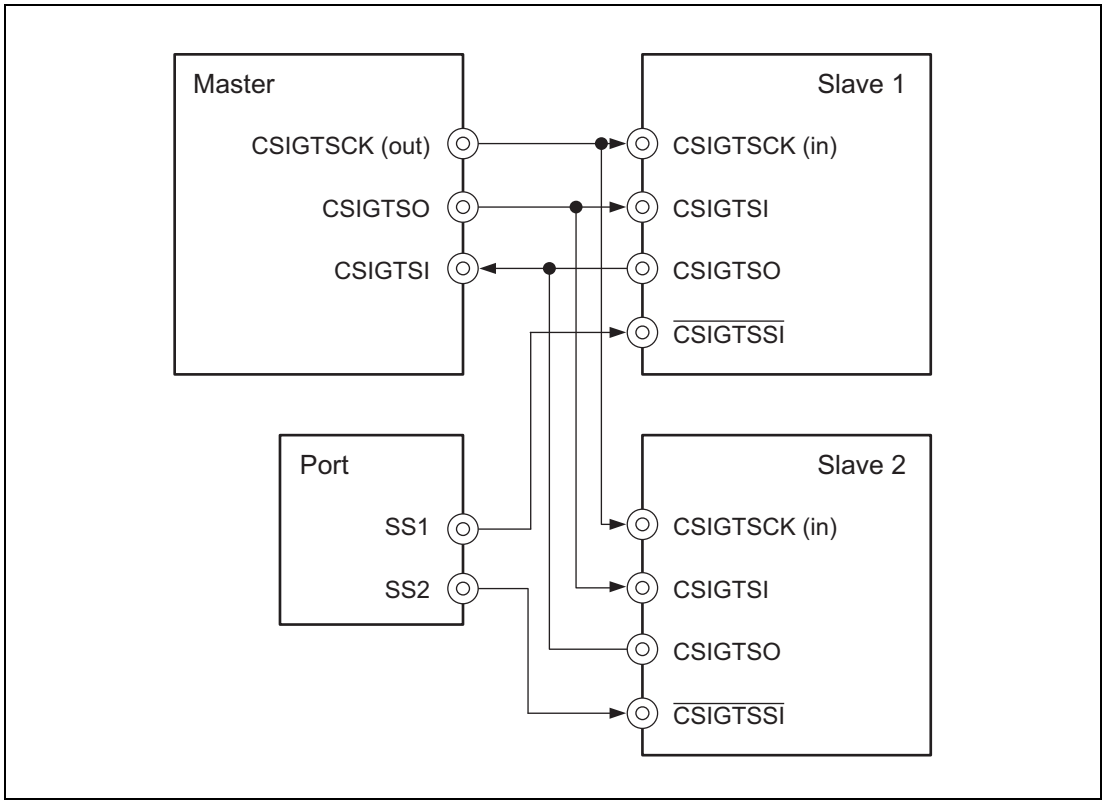


Figure 18.9 Master to Multiple Slaves Connection

A slave is selected (enabled) when its $\overline{\text{CSIGTSSI}}$ signal is low.

If it is not selected, the slave will neither receive nor transmit data. In addition, the CSIGTSO output buffer is disabled and set to input mode in order to avoid interference with the output of another slave which was selected, in transmit-only mode or transmit/receive mode (CSIGnCTL0.CSIGnTXE = 1).

18.5.3 Transmission Clock Selection

In master mode, the transfer clock frequency is selectable using the CSIGnPRS[2:0] and CSIGnBRS[11:0] bits in the CSIGnCTL2 register.

The following figure shows a block diagram of the BRG.

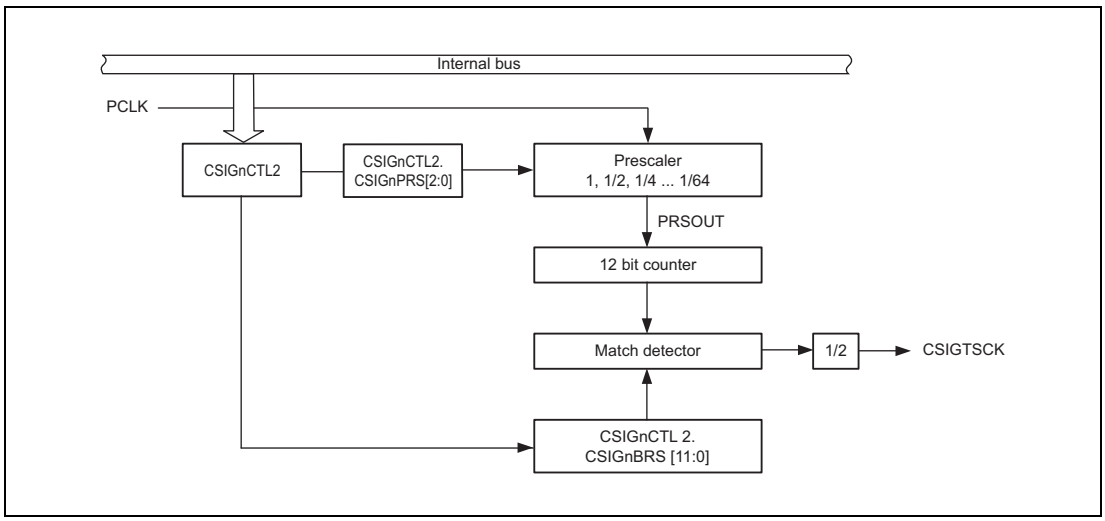


Figure 18.10 BRG Block Diagram

Setting CSIGnCTL2.CSIGnBRS[11:0] to 000_H disables the BRG.

Transfer clock frequency calculation

The transfer clock frequency in master mode is calculated as:

Transfer clock frequency (CSIGTSCK) = PCLK / (division ratio of PCLK) = PCLK / ($2^{\alpha} \times k \times 2$), where

α = CSIGnCTL2.CSIGnPRS[2:0] = 0 to 6

k = CSIGnCTL2.CSIGnBRS[11:0] = 1 to 4095

Transfer clock frequency upper and lower limits

When setting the transfer clock frequency, please note the followings.

- For the maximum transfer clock frequency of this product in master mode or slave mode, see the CSIG timing shown in the electrical characteristics. In addition, in either mode, set it within the specified range.
- The minimum transfer clock frequency in master mode and slave mode is PCLK/524160.
- The maximum transfer clock frequency is as follows:
  - In master mode: equal to or lower than PCLK/4
  - In slave mode: equal to or lower than PCLK/6

18.5.4 Data Transfer Modes

18.5.4.1 Transmit-Only Mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 0 puts the CSIG in transmit-only mode. Transmission starts when transmit data is written in the CSIGnTX0W or CSIGnTX0H register.

CAUTION

In case transmit-only mode has been entered after any reception mode, the data in the CSIGnRX0 buffer becomes undefined after completion of the first transmission. Consequently the reception register CSIGnRX0 has to be read before changing to transmit-only mode.

18.5.4.2 Receive-Only Mode

Setting CSIGnCTL0.CSIGnTXE = 0 and CSIGnCTL0.CSIGnRXE = 1 puts the CSIG in receive-only mode.

In master mode, reception starts when dummy data is read from the CSIGnRX0 register.
All following receptions are triggered by a read from the receive data register CSIGnRX0, as long as CSIGnBCTL0.CSIGnSCE = 1.

Moreover the communication start bit CSIGnBCTL0.CSIGnSCE has to be set to 1 and has to set back to 0 before reading the last received data from CSIGnRX0.

The recommended procedure is:

1. Set CSIGnBCTL0.CSIGnSCE = 1.
2. Read CSIGnRX0 (dummy data).
3. Wait for the reception interrupt INTCSIGTIR.
4. Read CSIGnRX0 (received data).
   In case of further data receptions continue at step 3, continue to read it until all data has been received.
   Before reading the last received data from CSIGnRX0, set CSIGnBCTL0.CSIGnSCE = 0.

In slave mode, reception starts when the communication clock CSIGTSCK from the master is received. In this case, it is not necessary to read data from the CSIGnRX0 register of the slave.

NOTE

In slave mode, any previously received data must be read from the reception register CSIGnRX0 in order to avoid any overwrite situation.

18.5.4.3 Transmit/Receive Mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 1 puts the CSIG in transmit/receive mode.

Data transfer (transmission and reception) starts when transmit data is written to the CSIGnTX0W or CSIGnTX0H register.

18.5.5 Data Length Selection

18.5.5.1 Data Length Selection without Extended Length

Transmission data length is selectable from 7 to 16 bits using the CSIGnDLS[3:0] bits in the CSIGnCFG0 register. The examples below show the communication with MSB first (CSIGnCFG0.CSIGnDIR = 0):

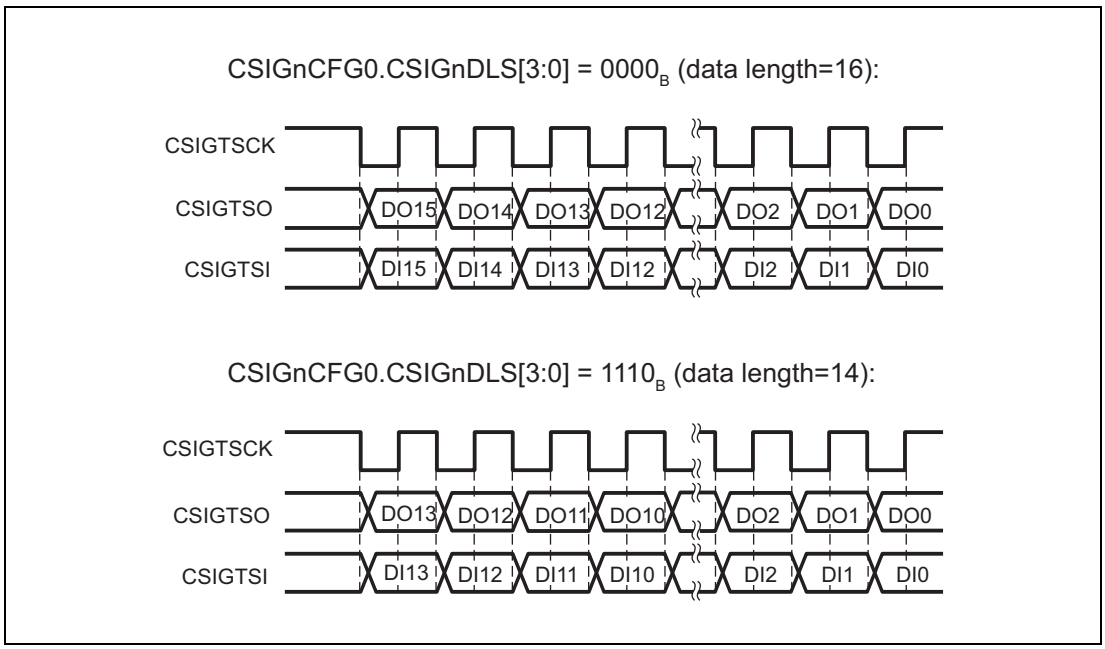


Figure 18.11 Data Length Select Function

18.5.5.2 Data Length Selection with Extended Data Length

If the data to be sent/received exceeds 16 bits, the extended data length (EDL) feature can be used.

The EDL function is enabled by setting bit CSIGnCTL1.CSIGnEDLE to 1.

The following describes how the EDL function works and how to specify the setting.

- The data has to be broken into 16-bit blocks plus remainder. For example, data of 42 bits would be broken into two 16-bit blocks plus 10 bits.
- The bit length of the remainder is set as “data length” in CSIGnCFG0.CSIGnDLS[3:0] bits.
- Set the CSIGnTX0W.CSIGnEDL bit to 1 to transmit 16-bit blocks. In this case, the data written to the CSIGnTX0W register is sent as 16-bit data regardless of the setting of the CSIGnCFG0.CSIGnDLS[3:0] bits.
- The transfer is complete after the data with the specified data length (the remainder with CSIGnTX0W.CSIGnEDL = 0) has been sent.

Example

Example for sending 40-bit data, for example the data $123456789A_H$:

40 bits are split into 2 × 16 bits plus 8 bits.

- Initialize CSIGnCFG0.CSIGnDLS[3:0] = 8_D.

- To send the data $123456789A_H$ with MSB first, write the following sequence to CSIGnTX0W:
  - $2000\ 1234_H$ (CSIGnTX0W.CSIGnEDL = 1)
  - $2000\ 5678_H$ (CSIGnTX0W.CSIGnEDL = 1)
  - $0000\ 009A_H$ (CSIGnTX0W.CSIGnEDL = 0)

The following figure illustrates the timing.

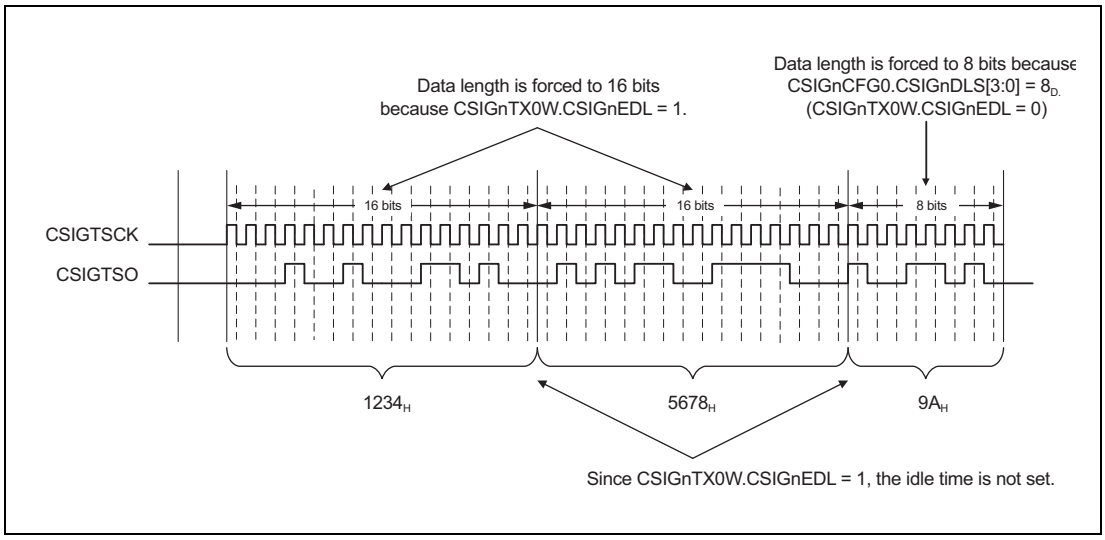


Figure 18.12 EDL Timing Diagram

NOTES

1. Data length with less than 7 bits can be set only when EDL mode is used.
2. It is not possible to send two consecutive data with a data length of less than 7 bits.
3. If parity is enabled, the parity bit is added after the last bit.
4. Explanation of data direction is provided in the following example.
   - Data to be transmitted: 123456_H
   - MSB first:
     Set CSIGnCFG0.CSIGnDIR to 0.
     Write $2000\ 1234_H$ to CSIGnTX0W (EDL bit = 1).
     Write $0000\ 0056_H$ to CSIGnTX0W (EDL bit = 0).
   - LSB first:
     Set CSIGnCFG0.CSIGnDIR to 1.
     Write $2000\ 3456_H$ to CSIGnTX0W (EDL bit = 1).
     Write $0000\ 0012_H$ to CSIGnTX0W (EDL bit = 0).
5. EDL mode cannot be used in the slave mode exclusive for receive.
   (CSIGnCTL2.CSIGnPRS[2:0] = 111B, CSIGnCTL0.CSIGnTXE = 0,
   CSIGnCTL0.CSIGnRXE = 1)

18.5.6 Serial Data Direction Select Function

The serial data direction is selectable using the CSIGnDIR bit in the CSIGnCFG0 register. The examples below show the communication for 8-bit data (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B):

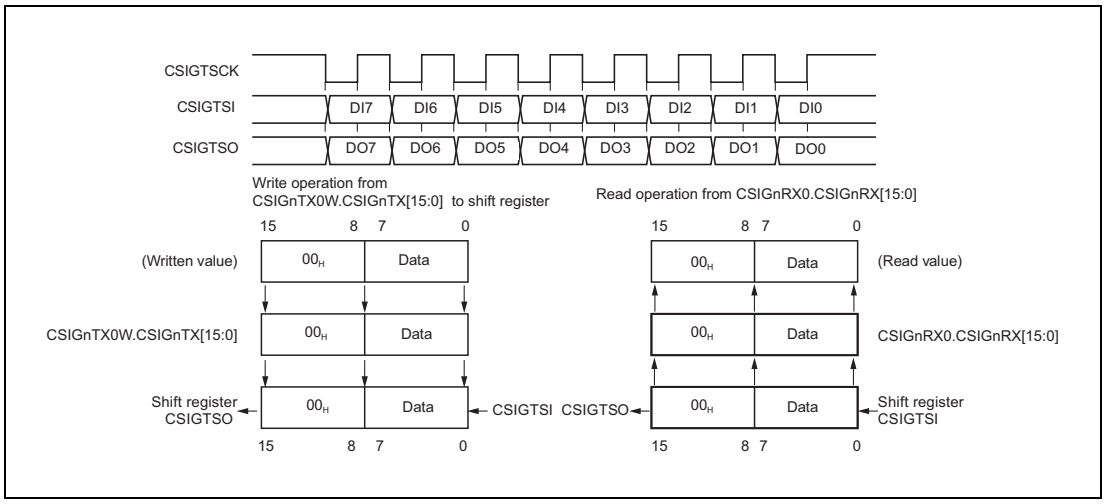


Figure 18.13 Serial Data Direction Select Function — MSB First (CSIGnDIR = 0)

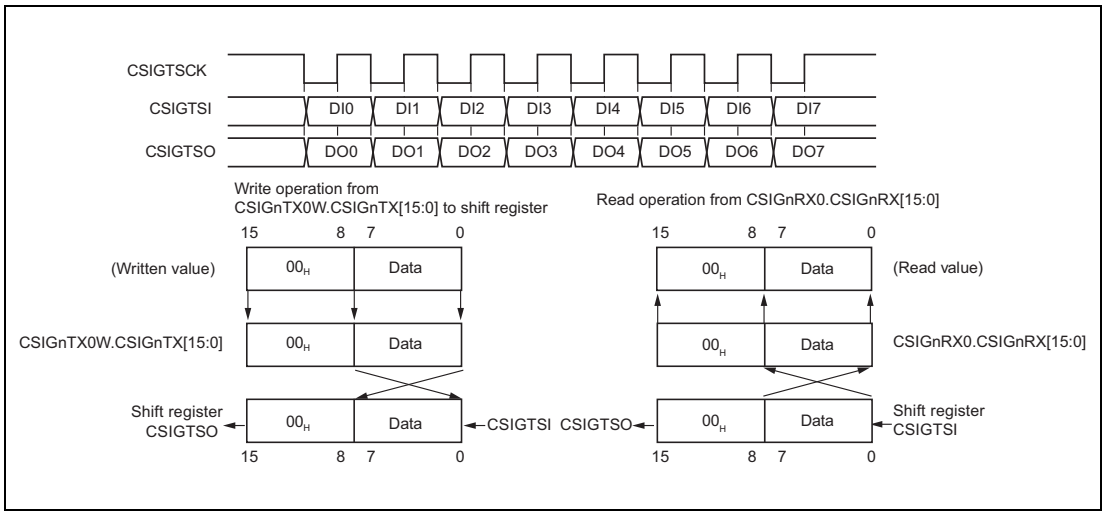


Figure 18.14 Serial Data Direction Select Function — LSB First (CSIGnDIR = 1)

18.5.7 Communication in Slave Mode

The following figure illustrates the communication signals and timings in slave mode.

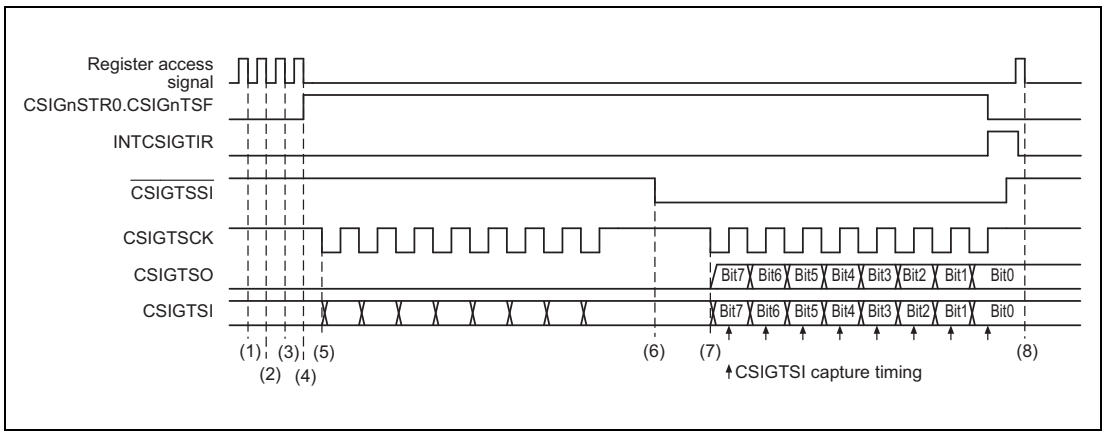


Figure 18.15 Rx/Tx Communication Timing in Slave Mode

1. CSIG is put into slave mode by setting CSIGnCTL2.CSIGnPRS[2:0] = 111_B. The $\overline{\text{CSIGTSSI}}$ signal is enabled (CSIGnCTL1.CSIGnSSE = 1) and the clock phase is at the high level (CSIGnCTL1.CSIGnCKR = 0).
2. Data length is 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B).
   Data direction is MSB first (CSIGnCFG0.CSIGnDIR = 0).
3. CSIG is set to transmit/receive mode (CSIGnCTL0.CSIGnPWR = 1, CSIGnCTL0.CSIGnTXE = 1, CSIGnCTL0.CSIGnRXE = 1).
4. When transfer data is written to the transmission register CSIGnTX0H, the “transmission in progress” flag CSIGnSTR0.CSIGnTSF is automatically set and the CSIG waits until signal $\overline{\text{CSIGTSSI}}$ goes low.
5. As long as signal $\overline{\text{CSIGTSSI}}$ is high, transmission/reception is not started even if the serial clock in input. CSIGTSO retains the values and input at CSIGTSI is ignored.
6. As soon as $\overline{\text{CSIGTSSI}}$ falls to low level, the chip serial data output is enabled and ready for transmission.
7. If the serial clock is input to the CSIG while $\overline{\text{CSIGTSSI}}$ is low, data is sent to CSIGTSO in synchronization with the serial clock, and at the same time, data is received from CSIGTSI.
8. The register CSIGnRX0 is read.

18.5.8 Handshake Function

CSIG features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by bit CSIGnCTL1.CSIGnHSE. For handshake, the signals CSIGTRYI and CSIGTRYO are used.

The timing depends on the data phase selection bit CSIGnCFG0.CSIGnDAP.

18.5.8.1 Slave Mode

If CSIGnCTL1.CSIGnHSE = 1, a low-level CSIGTRYO signal is output when the slave becomes busy. This happens when previous receive data is still in the CSIGnRX0 register, and new data cannot be copied from the shift register to CSIGnRX0 (CSIGnRX0 full condition).

The following examples assume a data length of 8 bits.

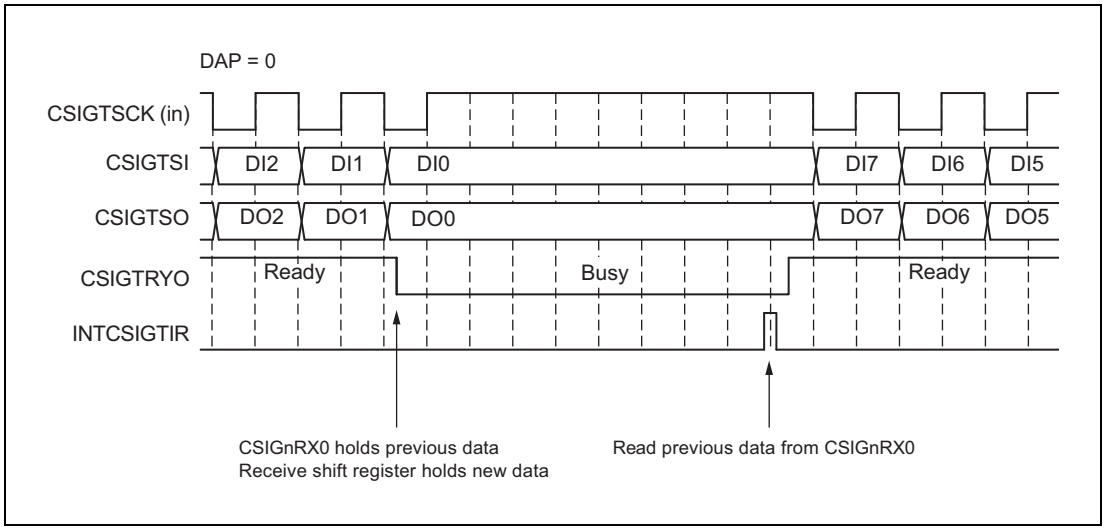


Figure 18.16 Ready/Busy Signal from Slave (CSIGnCFG0.CSIGnDAP = 0)

As long as the slave is busy, the master has to wait (i.e. suspend the transmission clock). The slave sets CSIGTRYO to high (“ready”) as soon as the reception register CSIGnRX0 has been read.

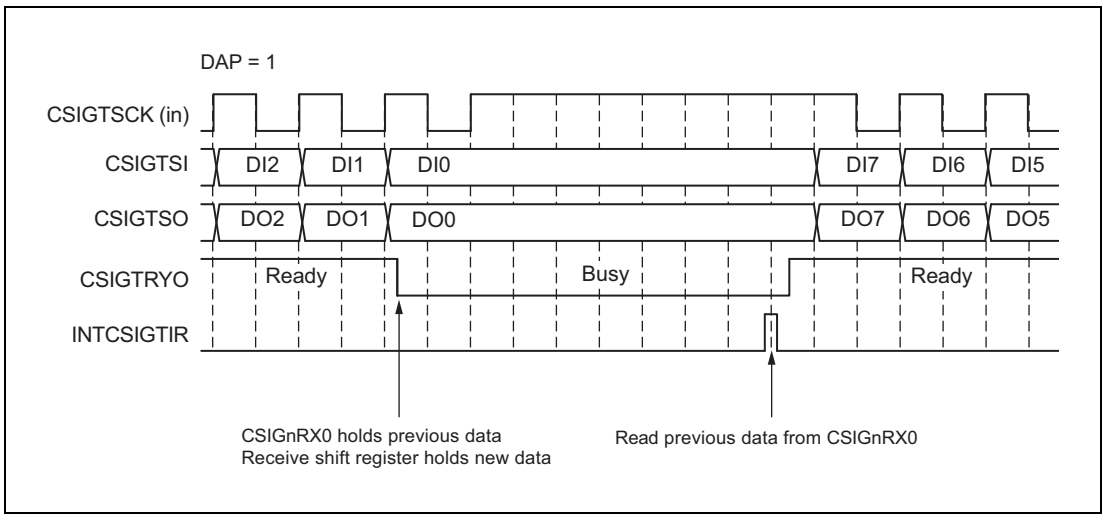


Figure 18.17 Ready/Busy Signal from Slave (CSIGnCFG0.CSIGnDAP = 1)

18.5.8.2 Master Mode

When the master detects low level of the CSIGTRYI, the following transfer is put on hold, and the master goes into wait status. It suspends the CSIGTSCK clock.

The CSIGTRYI level is checked at each half clock cycle of CSIGTSCK.

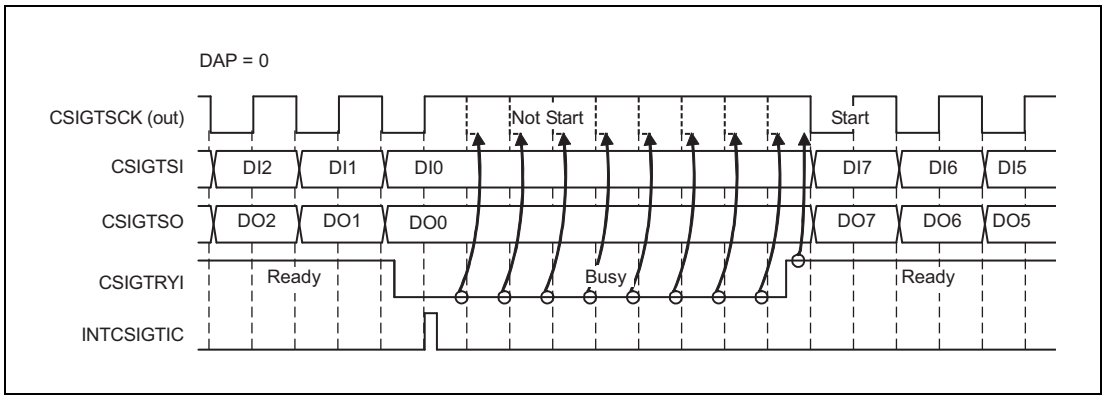


Figure 18.18 Master's Reaction to CSIGTRYI (CSIGnCFG0.CSIGnDAP = 0)

If the CSIGTRYI low signal comes from the slave while data transfer is in progress, the serial clock is suspended after the transfer is complete.

The master resumes the communication as soon as CSIGTRYI becomes high (slave is "ready").

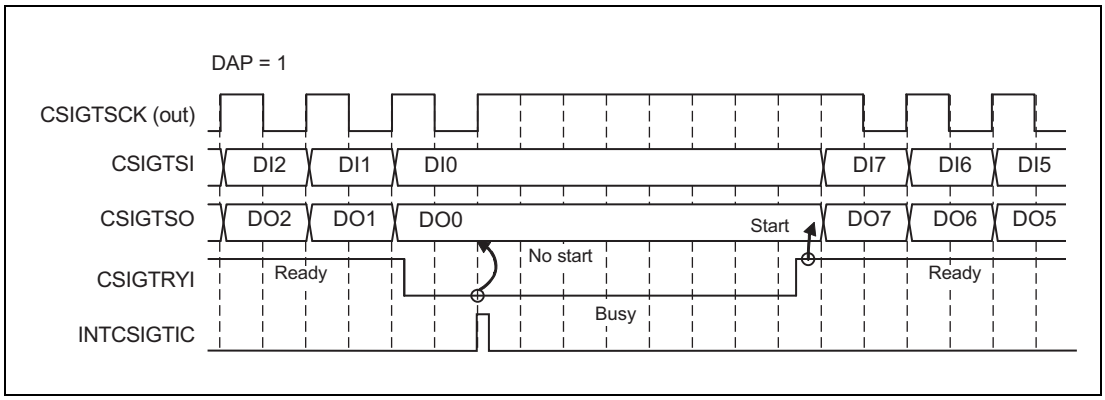


Figure 18.19 Master's Reaction to CSIGTRYI (CSIGnCFG0.CSIGnDAP = 1)

CAUTION

If multiple slaves are connected, the master must only detect the CSIGTRYI signal of the slave it has selected for communication.

CSIGTRYI must be pulled down by the slave before the next transfer starts. Even if the signal is pulled down by the slave during the transfer, the transfer will continue until it is finished.

18.5.9 Loop-Back Mode

Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active (CSIG0CTL1.CSIG0LBM = 1), the transmit and receive signals are internally connected, as shown in the figures below. The signals CSIGTSCK, CSIGTSO, and CSIGTSI are disconnected from the ports. In addition, the CSIGTSO output level is fixed to low, and CSIGTSCK is set to reset level (High). The rest of CSIG works as in normal operation.

In order to test the CSIG, set in loop-back mode and carry out normal transfer operations. Then check that the received data is the same as the transmitted data.

Table 18.26 Output Level of Pins

Pin	Output Level
CSIGTSCK(out)	High level
CSIGTSO	Low level (not dependent on the previous value)
Interrupt	Normal function
CSIGTRYO	Normal function (Low level)

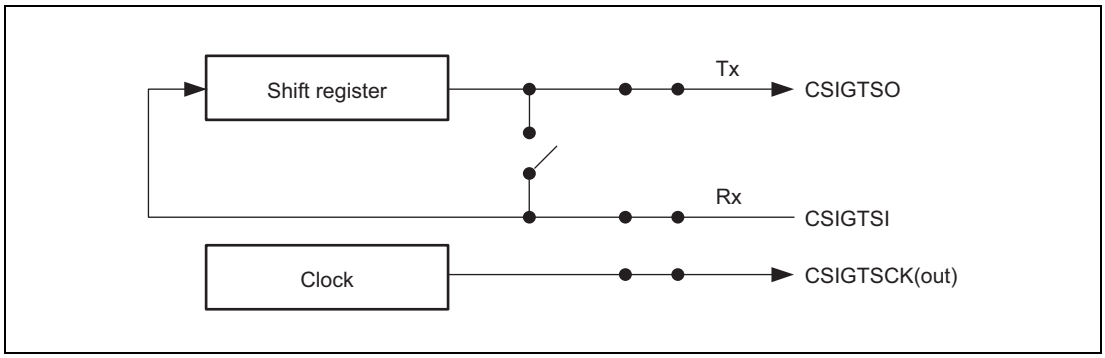


Figure 18.20 Normal Operation

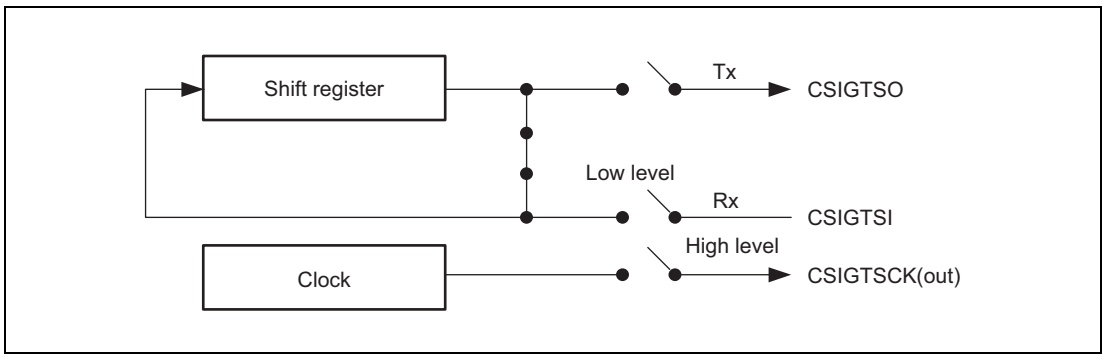


Figure 18.21 Operation in Loop-Back Mode

18.5.10 Error Detection

CSIG can detect three error types:

- Data consistency error (transmission data)
- Parity error (received data)
- Overrun error (received data)

Data consistency error and parity error check functions are independently enabled or disabled.

If one of these errors is detected, the interrupt INTCSIGTIRE is generated.

18.5.10.1 Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically sent as an output signal is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by bit CSIGnCTL1.CSIGnDCS (When checking data consistency, make sure that PIPCn.PIPCn_m = 1.). It is not active if data transmission is disabled (CSIGnCTL0.CSIGnTXE = 0).

When the data consistency check is active, the data transferred from CSIGnTX0W or CSIGnTX0H to the shift register is copied to a separate register. In addition, the physical levels at CSIGTSO are capture and the logical interpretation is written to an own shift register.

After completion of the transmission, the data sent is compared with the original transmission data.

Mismatch is considered as a data consistency error:

- Interrupt INTCSIGTIRE is generated.
- Bit CSIGnSTR0.CSIGnDCE is set.

The function is illustrated in the following block diagram.

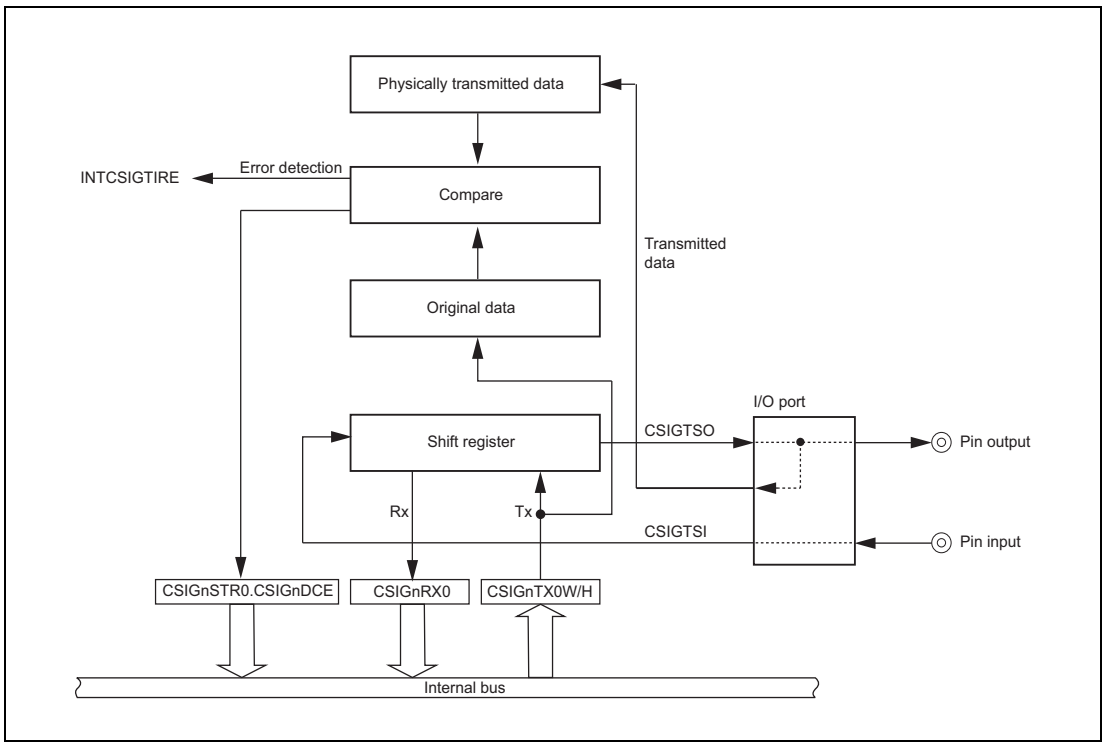


Figure 18.22 Functional Block Diagram of the Data Consistency Check

18.5.10.2 Parity Check

Parity is a common mean to detect a single bit failure during data transmission. CSIG can append a parity bit to the last data bit (even if extended data length is used).

The use and type of parity is specified in CSIGnCFG0.CSIGnPS[1:0].

Parity check is enabled if CSIGnCFG0.CSIGnPS[1] = 1.

The parity bit is checked after reception is complete. If a parity error occurs:

- Interrupt INTCSIGTIRE is generated.
- Bit CSIGnSTR0.CSIGnPE is set.

The following figure shows an example.

Data length is 8 bits. The data transmitted is 05_H and 35_H. Parity type is odd.

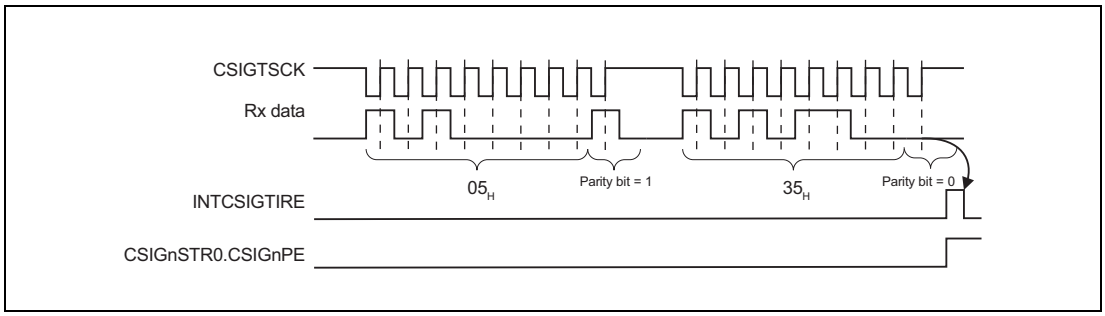


Figure 18.23 Parity Check Example

For the first 8 bits, the parity bit is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

For the second 8 bits, the parity bit is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

If the EDL (extended data length) function is used, the parity bit is added after the last bit of the data.

18.5.10.3 Overrun Error

This error occurs when previously received data still resides in the reception register CSIGnRX0, because it wasn't read, and new data is received.

The overrun error is not generated if data reception is disabled (CSIGnCTL0.CSIGnRXE = 0).

If overrun occurs:

- Interrupt INTCSIGTIRE is generated
- Bit CSIGnSTR0.CSIGnOVE is set
- Received data is overwritten and communication continues.

The following figure illustrates the function.

CSIG cannot store reception data because Rx register contains previous data.

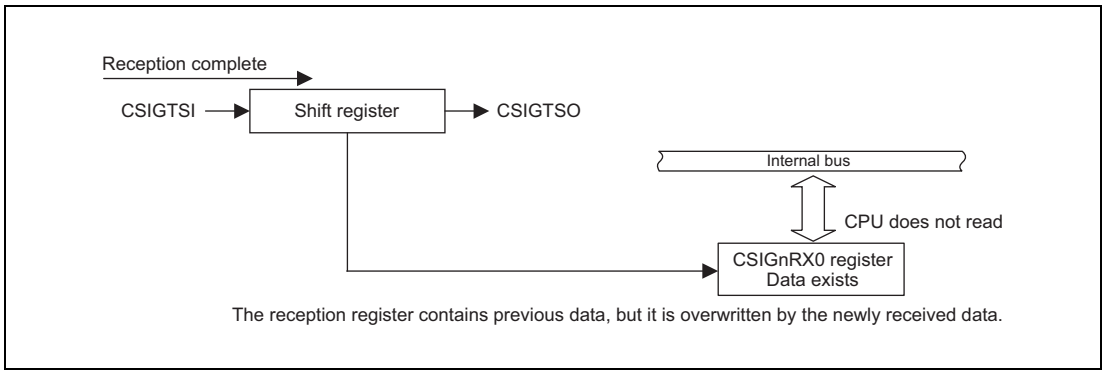


Figure 18.24 Overrun Error Detection

The following figure illustrates an example where:

- Rx data 3 was not read
- Rx data 4 was received, and data is overwritten.

Thus an overrun error occurs.

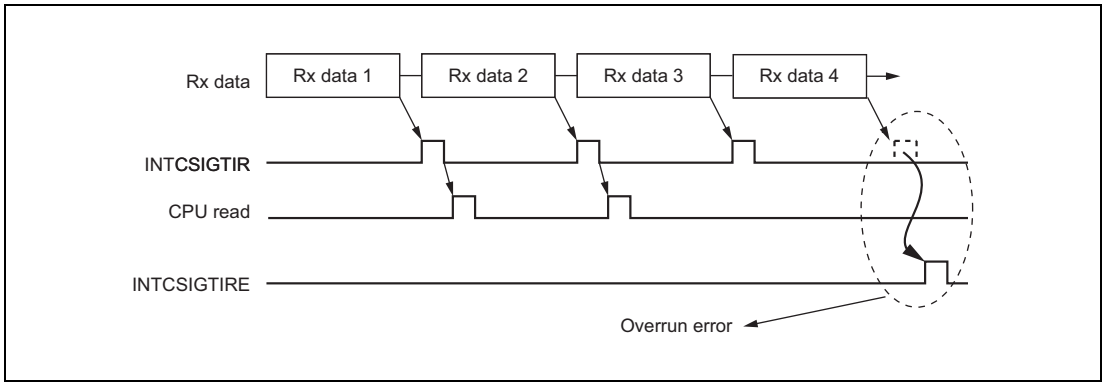


Figure 18.25 Overrun Error Detection - Example

NOTE

An overrun error can be avoided by using the handshake.

When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver has read its reception register and is ready again.

For details see Section 18.5.8, Handshake Function.

18.6 Operating Procedure

18.6.1 Master Mode Transmission/Reception by DMA

In the following a transmit/receive example in master mode in combination with a DMA is described.

The following instructions are based on the assumption that:

- Transmission data length is 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B)
- MSB is transmitted first (CSIGnCFG0.CSIGnDIR = 0)
- INTCSIGTIC interrupt at the end of the transfer (CSIGnCTL1.CSIGnSLIT = 0)
- Normal clock and data phase (CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0)
- The number of data is 10 (0 to 9)

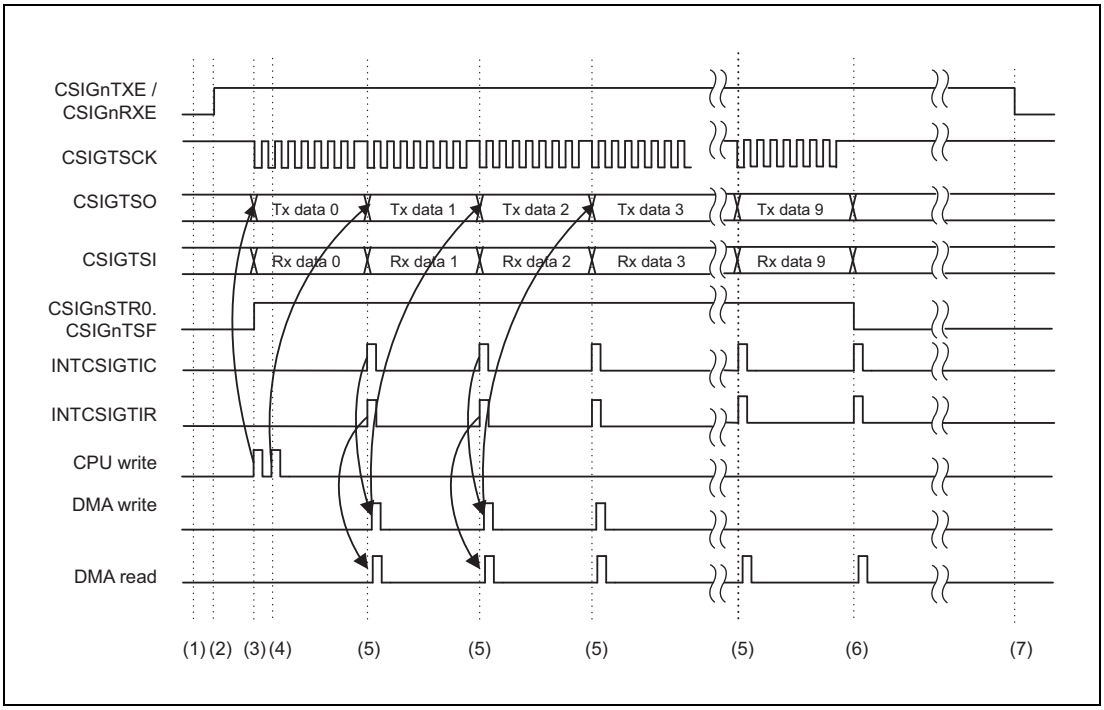


Figure 18.26 Communication in Master Mode

Procedure:

1. Configure the communication protocol in register CSIGnCFG0. Interrupt timing and operation mode are specified by setting the corresponding bits of the CSIGnCTL1 register and CSIGnCTL2 register.
2. In the CSIGnCTL0 register, set bits CSIGnPWR = 1 (enable the clock), CSIGnTXE = 1 (enable transmission), CSIGnRXE = 1 (enable reception).
3. Write the first data to be sent to the transmission register CSIGnTX0H. Transmission starts automatically when the first data is available.
4. Write the second data to CSIGnTX0H. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After every data that has been transmitted or received, the interrupts INTCSIGTIC and INTCSIGTIR are generated. INTCSIGTIC indicates that the next data can be written to

CSIGnTX0H. INTCSIGTIR indicates that the reception register CSIGnRX0 must be read.
In this example, CPU write and DMA write are equivalent.

6. No more write action is required after transmit completion of data 8. Data 9 (the last data) has been written after the transmission of data 7.
However, the reception register CSIGnRX0 must be read after receive completion of data 8 and 9.

7. To finally disable the transmit/receive operation, clear CSIGnCTL0.CSIGnTXE and CSIGnCTL0.CSIGnRXE. When no communication is taking place, set CSIGnCTL0.CSIGnPWR to "0" to minimize the power consumption of the CSIGn.

Section 19 Clocked Serial Interface H (CSIH)

This section contains a generic description of the Clocked Serial Interface H (CSIH).

The first part in this section describes all RH850/D1L/D1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of CSIH.

19.1 Features of RH850/D1L/D1M CSIH

19.1.1 Number of Units

This microcontroller has the following number of CSIH units.

Each CSIH unit has one channel interface.

Table 19.1 Number of Units

Product Name	All products
Number of units	2
Name	CSIHn (n = 0, 1)

Table 19.2 Index

Index	Meaning
n	Throughout this section, the individual CSIH units are identified by the index “n” (n = 0, 1): for example, CSIHnCTL0 is the CSIHn control register 0.
x	CSIHn has up to 8 chip select signals. Throughout this section, the individual chip select signals are identified by the index “x”: that is, CSx denotes a non-specified chip select signal.
y	A variable used for explanation is identified by the index “y”: for example, CSIHnBRSy is a non-specified baud rate setting register of CSIHn.

The following table shows values indicated by the indexes of each product.

Table 19.3 Indexes of Products

Indexes of Each Product
For the value of x, see Table 19.4, Number of Chip Select Signals.
y = 0 to 3

The numbers of chip select signals for each of the CSIH units are listed in the following table.

Table 19.4 Number of Chip Select Signals

Unit Name	Chip Select Index
CSIHn	CSx (x = 0)

19.1.2 Register Base Address

CSIH base addresses are listed in the following table.

CSIH register addresses are given as offsets from the base addresses in general.

Table 19.5 Register Base Address

Base Address Name	Base Address
<CSIH0_base>	FFDA 8000_H
<CSIH1_base>	FFDA $A000_H$

19.1.3 Clock Supply

The CSIH clock supply is shown in the following table.

Table 19.6 Clock Supply

Unit Name	Unit Clock Name	Internal Clock Signal
CSIHn	PCLK	Clock Controller CLKJIT

19.1.4 Interrupt Requests

CSIH interrupt requests are listed in the following table.

Table 19.7 Interrupt Requests

CSIHn signals	Function	Connected to
CSIH0:		
INTCSIHTIC	Communication status interrupt	Interrupt Controller INTCSIH0IC DMA Controller trigger ID 103
INTCSIHTIR	Receive status interrupt	Interrupt Controller INTCSIH0IR DMA Controller trigger ID 104
INTCSIHTIRE	Communication error interrupt	Interrupt Controller INTCSIH0IRE
INTCSIHTIJC	Job completion interrupt	Interrupt Controller INTCSIH0IJC DMA Controller trigger ID 105
CSIH1:		
INTCSIHTIC	Communication status interrupt	Interrupt Controller INTCSIH1IC DMA Controller trigger ID 106
INTCSIHTIR	Receive status interrupt	Interrupt Controller INTCSIH1IR DMA Controller trigger ID 107
INTCSIHTIRE	Communication error interrupt	Interrupt Controller INTCSIH1IRE
INTCSIHTIJC	Job completion interrupt	Interrupt Controller INTCSIH1IJC DMA Controller trigger ID 108

19.1.5 Reset Sources

CSIH reset sources are listed in the following table. CSIH is initialized by these reset sources.

Table 19.8 Reset Sources

Unit Name	Reset Source
CSIHn	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

19.1.6 External Input/Output Signals

External input/output signals of CSIH are listed below.

Table 19.9 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
CSIH0		
CSIHTSCK	Serial clock signal	CSIH0SC*1
CSIHTSI	Serial data input signal	CSIH0SI*1
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH0SSI}}$*1
CSIHTRYI	Ready/busy input signal	CSIH0RYI*1
CSIHTSO	Serial data output signal	CSIH0SO
CSIHTRYO	Ready/busy output signal	CSIH0RYO
CSIHTCSS0	Chip select signal	CSIH0CSS0
CSIHTCSS[7:1]		not connected
CSIH1		
CSIHTSCK	Serial clock signal	CSIH1SC*1
CSIHTSI	Serial data input signal	CSIH1SI*1
$\overline{\text{CSIHTSSI}}$	Slave select input signal	$\overline{\text{CSIH1SSI}}$*1
CSIHTRYI	Ready/busy input signal	CSIH1RYI*1
CSIHTSO	Serial data output signal	CSIH1SO
CSIHTRYO	Ready/busy output signal	CSIH1RYO
CSIHTCSS0	Chip select signal	CSIH1CSS0
CSIHTCSS[7:1]		not connected

Note 1. These input signals are passed through a noise filter, refer to the section “Port Filters” in the section “Port Functions”.

Note 2. For availability of these signals, refer to the Section 2.3.2, List of Alternative Function Pins.

19.1.7 Data Consistency Check

The following table lists the port pins on which CSIHnSO pin functions are multiplexed and data consistency checking. See Section 19.5.12, Error Detection for details on data consistency checking.

Table 19.10 Data Consistency Checking and Port Pins

Unit Signal Name	Port Pin Name	Alternative Function
CSIH0		
CSIHTSO	P0_1	ALT_OUT2
	P1_3	ALT_OUT2
CSIH1		
CSIHTSO	P1_0	ALT_OUT2

19.2 Overview

19.2.1 Functional Overview

- Three-wire serial synchronous data transfer
- Master mode and slave mode selectable
- Multiple slaves configuration plus RCB (Recessive Configuration for Broadcasting) due to eight configurable chip select output signals
- Slave select input signal ($\overline{\text{CSIHTSSI}}$) is usable
- Four built-in baud rate generators
- Transfer clock frequency is adjustable in master mode, whereas it is determined by the input clock in slave mode.
- Maximum transfer clock frequency:
  - Master mode: up to PCLK/4
  - Slave mode: up to PCLK/6
- Phase of clock and data selectable
- Data transfer with MSB or LSB first selectable
- Transfer data length selectable
  - CSIH0: from 2 to 16 bits in 1-bit units
  - CSIH1: from 7 to 16 bits in 1-bit units
- EDL (Extended Data Length) function for transferring data with more than 16 bits is included
- Three selectable transfer modes:
  - transmit-only mode
  - receive-only mode
  - transmit/receive mode
- Built-in handshake function
- Error detection (data consistency check, parity, time-out, overflow, and overrun) is included
- Support of job concept
- 16 words I/O buffer memory
- Selectable direct access mode and memory mode (FIFO, dual buffer, and transmit-only buffer)
- Four different interrupt request signals (INTCSIHTIC, INTCSIHTIR, INTCSIHTIRE, INTCSIHTIJC)
- LBM (Loop Back Mode) function for self test is included
- CPU-controlled high-priority communication function (CSIH0 only)
- Enforced chip select idle setting
- RCB (Recessive Configuration for Broadcasting) bit is included
- JOB enable control bit for AUTOSAR is included.

19.2.2 Functional Overview Description

The CSIH uses three signals for communication:

- Transmission clock CSIHTSCK (output in master mode, input in slave mode)
- Data output signal CSIHTSO
- Data input signal CSIHTSI

Additional signals are available for external control and monitoring.

- $\overline{\text{CSIHTSSI}}$: Slave select input signal
- CSIHTRYO: Ready/busy output signal (handshake signal)
- CSIHTRYI: Ready/busy input signal (handshake signal)
- CSIHTCSS[7:0]: Chip select signals

Data transmission is bit-wise and serial, and synchronous to the transmission clock.

The following table shows the most important registers for setting up the CSIH.

Table 19.11 Main Registers of CSIH

Register	Function
CSIHnCTL0	Enables/disables serial clock, and permits/ prohibits data transmission and data reception. Defines end-of-job behavior and enables/disables buffering (bypass of the buffer).
CSIHnCTL1	Controls options like interrupt timing, extended data length, job feature, data consistency check, loop-back mode, handshake, etc.
CSIHnCTL2	Selects master or slave mode, and the transfer clock frequency of the built-in baud rate generator (BRG) in master mode.
CSIHnMCTL0	Selects memory mode and specifies the time-out value
CSIHnMCTL1	Controls the memory in FIFO mode
CSIHnMCTL2	Controls the memory in dual buffer mode
CSIHnCFGx	Registers to configure the communication protocol for each chip select signal

19.2.3 Block Diagram

The block diagram shows the main components of the CSIH.

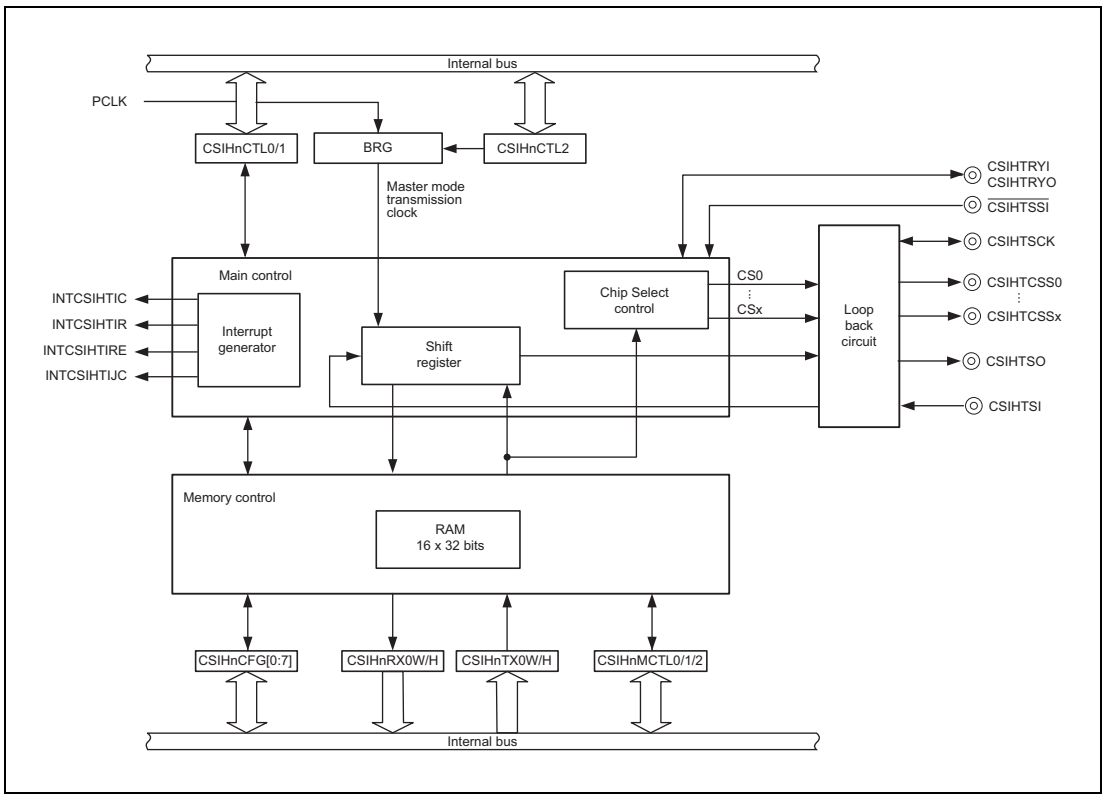


Figure 19.1 CSIH Block Diagram

NOTE

The following functional description of the CSIH module refers to a FIFO buffer size of 128 words. The description is also valid for the 16 words buffer size of the RH850/D1L/D1M devices.

In master mode, the transmission clock CSIHTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the transmission clock is provided from an external source.

The built-in memory can be configured as FIFO, dual buffer (separate transmit and receive buffers), or transmit-only buffer. It can also be bypassed for data transmission and reception without buffering.

The loop back circuit disconnects the CSIH completely from the ports and supports internal self test.

NOTE

This section describes the following modes:

- The “operating mode” separates between master and slave mode. In this context, only a master can control and communicate with several slaves (for details see **Section 19.5.1, Operating Modes (Master/Slave)**).
- The “job mode” is related to the AUTOSAR job concept (for details see **Section 19.5.3.3, Job Concept**).
- The “memory mode” takes the various configurations of the associated buffer memory into account (for details see Section 19.5.6, CSIH Buffer Memory).
- The “data transfer mode” specifies the kind of the communication – transmit-only, receive-only, or transmit/receive (for details see Section 19.5.7, Data Transfer Modes).

19.3 Registers

19.3.1 List of Registers

CSIH registers are listed in the following table.

For details about CSIHn_base, see Section 19.1.2, Register Base Address.

Table 19.12 Registers

Module	Register	Symbol	Address
CSIHn	CSIHn control register 0	CSIHnCTL0	<CSIHn_base> + 0000_H
CSIHn	CSIHn control register 1	CSIHnCTL1	<CSIHn_base> + 0010_H
CSIHn	CSIHn control register 2	CSIHnCTL2	<CSIHn_base> + 0014_H
CSIHn	CSIHn status register 0	CSIHnSTR0	<CSIHn_base> + 0004_H
CSIHn	CSIHn status clear register 0	CSIHnSTCR0	<CSIHn_base> + 0008_H
CSIHn	CSIHn memory control register 0	CSIHnMCTL0	<CSIHn_base> + 1040_H
CSIHn	CSIHn memory control register 1	CSIHnMCTL1	<CSIHn_base> + 1000_H
CSIHn	CSIHn memory control register 2	CSIHnMCTL2	<CSIHn_base> + 1004_H
CSIHn	CSIHn memory read/write pointer register 0	CSIHnMRWP0	<CSIHn_base> + 1018_H
CSIHn	CSIHn configuration register 0	CSIHnCFG0	<CSIHn_base> + 1044_H
CSIHn	CSIHn configuration register 1	CSIHnCFG1	<CSIHn_base> + 1048_H
CSIHn	CSIHn configuration register 2	CSIHnCFG2	<CSIHn_base> + $104C_H$
CSIHn	CSIHn configuration register 3	CSIHnCFG3	<CSIHn_base> + 1050_H
CSIHn	CSIHn configuration register 4	CSIHnCFG4	<CSIHn_base> + 1054_H
CSIHn	CSIHn configuration register 5	CSIHnCFG5	<CSIHn_base> + 1058_H
CSIHn	CSIHn configuration register 6	CSIHnCFG6	<CSIHn_base> + $105C_H$
CSIHn	CSIHn configuration register 7	CSIHnCFG7	<CSIHn_base> + 1060_H
CSIHn	CSIHn transmit data register 0 for word access	CSIHnTX0W	<CSIHn_base> + 1008_H
CSIHn	CSIHn transmit data register 0 for half word access	CSIHnTX0H	<CSIHn_base> + $100C_H$
CSIHn	CSIHn receive data register 0 for word access	CSIHnRX0W	<CSIHn_base> + 1010_H
CSIHn	CSIHn receive data register 0 for half word access	CSIHnRX0H	<CSIHn_base> + 1014_H
CSIHn	CSIHn emulation register	CSIHnEMU	<CSIHn_base> + 0018_H
CSIHn	CSIHn baud rate setting register 0 (CSHI0 only)	CSIHnBRS0	<CSIHn_base> + 1068_H
CSIHn	CSIHn baud rate setting register 1 (CSHI0 only)	CSIHnBRS1	<CSIHn_base> + $106C_H$
CSIHn	CSIHn baud rate setting register 2 (CSHI0 only)	CSIHnBRS2	<CSIHn_base> + 1070_H
CSIHn	CSIHn baud rate setting register 3 (CSHI0 only)	CSIHnBRS3	<CSIHn_base> + 1074_H

NOTE

In the IO header files, availability of each register depends on whether they can be used by each product. The register has no corresponding function terminal has been removed from the IO header file by 1-bit. For availability of terminals, refer to the Section 2.3.2, List of Alternative Function Pins.

19.3.2 CSIHnCTL0 — CSIHn Control Register 0

This register controls the operation clock and enables/disables transmission/reception and the memory part for transmission and/or reception. It forces the stop of communication at the end of the current job.

Access: This register can be read/written in 8-bit or 1-bit units.

Address: <CSIHn_base> + 0000_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CSIHnPWR	CSIHnTXE	CSIHnRXE	—	—	—	CSIHnJOBE	CSIHnMBS
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R/W	R/W

Table 19.13 CSIHnCTL0 Register Contents

Bit Position	Bit Name	Function
7	CSIHnPWR	Controls the operation clock. 0: Stops operation clock. 1: Provides operation clock. Clearing CSIHnPWR to 0 resets the internal circuits, stops operation, and sets CSIH to standby state. No clock is provided to internal circuits. If CSIHnPWR is cleared (to 0) during communication, ongoing communication is immediately aborted. In this case, communication setting must be started over.
6	CSIHnTXE	Permits or prohibits transmission. 0: Prohibits transmission. 1: Permits transmission.
5	CSIHnRXE	Permits or prohibits reception. 0: Prohibits reception. 1: Permits reception.
4 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	CSIHnJOBE	Stops communication at the end of the current job (Communication ends if data is written to the transmission buffer when CSIHnTX0W.CSIHnEOJ = 1 (job completion)). 0: Communication stop is not requested. 1: Stops communication. This bit can be used to abort an ongoing job. This bit is cleared to 0 automatically. If this bit is set to 1, the read value is always 0. In FIFO mode, the pointer must be cleared by setting CSIHnSTCR0.CSIHnPCT = 1 before the next communication is started
0	CSIHnMBS	Bypasses the memory for transmission and/or reception data. 0: Memory mode CSIH memory is used for transmission and/or reception data. 1: Direct access mode CSIH memory is bypassed.

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.3 CSIHnCTL1 — CSIHn Control Register 1

This register specifies the interrupt timing and the interrupt delay mode. It enables/disables extended data length control, data consistency check, loop-back mode, handshake functionality, and job mode. It selects the active output level of each chip select signal and the behavior of the chip select signals after the transfer of the final data.

Access: This register can be read/written in 32-bit units.

Address: <CSIHn_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	CSIH0 SLRS*1	—	—	—	—	—	CSIH0 PHE*1	CSIHn CKR	CSIHn SLIT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn CSL7	CSIHn CSL6	CSIHn CSL5	CSIHn CSL4	CSIHn CSL3	CSIHn CSL2	CSIHn CSL1	CSIHn CSL0	CSIHn EDLE	CSIHn JE	CSIHn DCS	CSIHn CSRI	CSIHn LBM	CSIHn SIT	CSIHn HSE	CSIHn SSE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. This bit is only available for CSIH0.

Table 19.14 CSIHnCTL1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
24 of CSIH0	CSIH0SLRS	Sets the internal synchronization timing for receive data input. 0: Rising edge of PCLK 1: Falling edge of PCLK For differences by the setting, see Data Sheet.
24 of CSIH1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
23 to 19	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
18 of CSIH0	CSIH0PHE	Sets the CPU-controlled priority-based communication function. 0: The CPU-controlled high-priority communication function is disabled. 1: The CPU-controlled high-priority communication function is enabled. To enable the CPU-controlled high-priority communication function, set this bit to 1 and set CSIHnJE = 1. This bit can only be set in transmit-only buffer mode.
18 of CSIH1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
17	CSIHnCKR	CSIHTSCK Clock Inversion Function 0: The default level of CSIHTSCK is high 1: The default level of CSIHTSCK is low For details, see Section 19.3.11, CSIHnCFGx — CSIHn Configuration Register x.
16	CSIHnSLIT	Selects the timing of interrupt INTCSIHTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: As soon as the contents of the CSIHnTX0W/H register are transferred to the shift register, an interrupt is generated (this function is activated only in direct access memory mode/transmit-only buffer mode). For details, see Section 19.4.3, INTCSIHTIC (Communication Status Interrupt).

Table 19.14 CSIHnCTL1 Register Contents (2/2)

Bit Position	Bit Name	Function
15 to 8	CSIHnCSLx	Selects the active output level of chip select signal x (CSIHTCSSx). 0: Chip select is active low. 1: Chip select is active high. For details, see Section 19.5.3, Chip Selection (CS) Features.
7	CSIHnEDLE	Enables/disables extended data length (EDL) mode. 0: Disables extended data length mode. 1: Enables extended data length mode. For details, see Section 19.5.8.2, Data Length Greater than 16 Bits.
6	CSIHnJE	Enables/disables job mode. 0: Disables job mode. 1: Enables job mode. For details, see Section 19.5.3.3, Job Concept. The CSIHnCTL0.CSIHnJOBE, CSIHnTX0W.CSIHnEOJ, and CSIHnTX0W.CSIHnCIRE bits are enabled only when CSIHnJE = 1. Setting this bit in slave mode is prohibited. **For CSIH0 only:** In addition, to enable the CPU-controlled high-priority communication function, set CSIHnPHE = 1 and this bit to 1.
5	CSIHnDCS	Enables/disables data consistency check. 0: Disables data consistency check. 1: Enables data consistency check. For details, see Section 19.5.12.1, Data Consistency Check.
4	CSIHnCSRI	Defines chip select signal behavior after last data transfer. 0: Chip select signal holds the active level. 1: Chip select signal returns to the inactive level. The last data is determined at the interrupt timing in direct access mode or FIFO mode. When CSIHnCTL1.CSIHnSLIT = 0, the last data is determined in direct access mode.
3	CSIHnLBM	Controls loop-back mode (LBM). 0: Deactivates loop-back mode. 1: Activates loop-back mode. For details, see Section 19.5.13, Loop-Back Mode.
2	CSIHnSIT	Selects interrupt delay mode. 0: No delay is generated. 1: Half clock delay is generated for all interrupts. This bit is only valid in master mode. In slave mode, no delay is generated. For details, see Section 19.4.2, Interrupt Delay.
1	CSIHnHSE	Enables/disables handshake function. 0: Disables the handshake function. 1: Enables the handshake function. For details see Section 19.5.11, Handshake Function.
0	CSIHnSSE	Enables/disables the slave select function. 0: Input signal $\overline{\text{CSIHTSSI}}$ is disabled. 1: Input signal $\overline{\text{CSIHTSSI}}$ is recognized. If the slave select function is not used, this bit must be set to 0 (see also Section 19.5.2, Master/Slave Connections).

Details about CSIHnCTL1.CSIHnSSE are shown in the following tables.

Table 19.15 Operation of the Slave Select Function during Reception

CSIHnCTL0. CSIHnRXE	CSIHnCTL1. CSIHnSSE	$\overline{\text{CSIHTSSI}}$	Receive Operation
0	—	—	Reception is prohibited
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

Table 19.16 Operation of the Slave Select Function during Transmission

CSIHnCTL0. CSIHnTXE	CSIHnCTL1. CSIHnSSE	$\overline{\text{CSIHTSSI}}$	Transmit Operation
0	—	—	Transmission is prohibited
1	0	—	Possible
1	1	0	Possible
1	1	1	Disabled

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.4 CSIHnCTL2 — CSIHn Control Register 2

This register selects operating mode and the basic clock value, and specifies the transfer clock frequency.

For details see Section 19.5.5, Transmission Clock Selection.

Access: This register can be read/written in 16-bit units.

Address: <CSIHn_base> + 0014_H

Value after reset: $E000_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnPRS[2:0]			–	CSIH1BRS[11:0]*1											
Value after reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. These bits are only available for CSIH1.

Table 19.17 CSIHnCTL2 Register Contents

Bit Position	Bit Name	Function
15 to 13	CSIHnPRS[2:0]	These bits select the operation mode and the reference clock value.
12	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
11 to 0 of CSIH0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
11 to 0 of CSIH1	CSIH1BRS[11:0]	Selects the baud rate:

CSIHnPRS2	CSIHnPRS1	CSIHnPRS0	Selection of Reference Clock (PRSOUT)
0	0	0	PCLK (Master mode)
0	0	1	PCLK/2 (Master mode)
0	1	0	PCLK/4 (Master mode)
0	1	1	PCLK/8 (Master mode)
1	0	0	PCLK/16 (Master mode)
1	0	1	PCLK/32 (Master mode)
1	1	0	PCLK/64 (Master mode)
1	1	1	External clock via CSIHTSCK(in) (Slave mode)

CSIHnBRS[11:0]	Baud rate at CSIHTSCK
0	BRG stopped
1	PCLK / (2^{α} × 1 × 2)
2	PCLK / (2^{α} × 2 × 2)
3	PCLK / (2^{α} × 3 × 2)
4	PCLK / (2^{α} × 4 × 2)
…	…
4095	PCLK / (2^{α} × 4095× 2)

19.3.4.1 CSIH0 transfer clock frequency setting

In master mode, the following bits are used to set the transfer clock frequency:

CSIH0CTL2.CSIH0PRS[2:0], CSIH0CFGx.CSIH0BRSS[1:0], CSIH0BRSy.CSIH0BRS[11:0]

In addition, any of the four different transfer clock frequency settings that are specified by the CSIH0BRSy.CSIH0BRS[11:0] bits is selected according to the chip select signal. To select the transfer clock frequency setting for each chip select signal, use the CSIH0CFGx.CSIH0BRSS[1:0] bits.

The following table shows the relationship between CSIH0CFGx.CSIH0BRSS[1:0] and CSIH0BRSy.CSIH0BRS[11:0].

CSIH0CFGx (x = 0) CSIH0BRSS1 to 0	Transfer Clock Frequency Setting Bit to be Selected
00	CSIH0BRS0.CSIH0BRS[11:0]
01	CSIH0BRS1.CSIH0BRS[11:0]
10	CSIH0BRS2.CSIH0BRS[11:0]
11	CSIH0BRS3.CSIH0BRS[11:0]

The following table shows the relationship between the transfer clock frequency and the transfer clock frequency setting (CSIH0BRSy[11:0]) selected by the CSIH0BRSS[1:0] bits when the bit value of the CSIH0PRS[2:0] bits is α.

CSIH0BRSy[11:0]	Transfer clock frequency
0	BRG stopped
1	PCLK / (2^{α} × 1 × 2)
2	PCLK / (2^{α} × 2 × 2)
3	PCLK / (2^{α} × 3 × 2)
4	PCLK / (2^{α} × 4 × 2)
...	…
4095	PCLK / (2^{α} × 4095× 2)

When a time-out error is used in slave mode, the clock selected by this setting is used. In slave mode, the CSIHnPRS[2:0] bits are set to 111_B. In this case, the prescaler has the same setting as when the CSIHnPRS[2:0] bits are set to 000_B. If you are using a time-out error of CSIH0, set the CSIH0BRSy.CSIH0BRS[11:0] bits to a value other than 000_H. If you are using a time-out error of CSIH1, set the CSIH1CTL2.CSIH1BRS[11:0] bits to a value other than 000_H.

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.5 CSIHnSTR0 — CSIHn Status Register 0

This register indicates the status of CSIH.

Access: This register can only be read in 32-bit units.

Address: <CSIHn_base> + 0004_H

Value after reset: $0000\ 0010_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHnSRP[7:0]								CSIHnSPF[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn TMOE	CSIHn OFE	—	—	—	—	—	CSIH0 HPST*1	CSIHn TSF	—	CSIHn FLF	CSIHn EMF	CSIHn DCE	—	CSIHn PE	CSIHn OVE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. This bit is only available for CSIH0.

Table 19.18 CSIHnSTR0 Register Contents (1/3)

Bit Position	Bit Name	Function
31 to 24	CSIHnSRP[7:0]	Indicates the number of received data packets in FIFO mode. **CSIHnSRP[7:0]** / **Description** 00_H / Number of received data (0 to 128) … 80_H Other than the above / Undefined These bits are cleared by CSIHnSTCR0.CSIHnPCT. In direct access mode, dual buffer memory mode, or transmit-only buffer memory mode, this value is fixed to 00_H. In direct access mode, this bit is fixed to 0 because there is no pointer. In buffer mode, this bit is fixed to 0 because the number of data packets is managed by CSIHnMCTL2.CSIHnND[7:0].
23 to 16	CSIHnSPF[7:0]	Indicates the number of unsent data in FIFO mode. (The number of data written by the CPU is the number of sent data.) **CSIHnSPF[7:0]** / **Description** 00_H / Number of unsent data packets (0 to 128) … 80_H Other than the above / Undefined These bits are cleared by CSIHnSTCR0.CSIHnPCT. In direct access mode, dual buffer memory mode, or transmit-only buffer memory mode, this value is fixed to 00_H. In direct access mode, this bit is fixed to 0 because there is no pointer. In buffer mode, this bit is fixed to 0 because the number of data packets is managed by CSIHnMCTL2.CSIHnND[7:0].

Table 19.18 CSIHnSTR0 Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15</td><td>CSIHnTMOE</td><td>Time-out Error Flag in FIFO Mode
Indicates whether a time-out error was detected in FIFO mode.
0: No time out error is detected.
1: A time out error is detected.
For details, see Section 19.5.12.3, Time-Out Error.
This bit is cleared by CSIHnSTCR0.CSIHnTMOEC.
When setting to 1 by time-out error detection and clearing to 0 by CSIHnSTCR0.CSIHnTMOEC occur simultaneously, setting to 1 takes precedence over clearing to 0.
This bit is also initialized when CSIHnCTL0.CSIHnPWR is changed to 0 to 1 or 1 to 0.</td></tr>
<tr><td>14</td><td>CSIHnOFE</td><td>Overflow Error Flag in FIFO mode
Indicates whether an overflow error was detected in FIFO mode.
0: No overflow error is detected.
1: An overflow error is detected.
For details, see Section 19.5.12.4, Overflow Error.
This bit is cleared by CSIHnSTCR0.CSIHnOFEC.
When setting to 1 by overflow error detection and clearing to 0 by CSIHnSTCR0.CSIHnOFEC occur simultaneously, setting to 1 by CSIHnSTCR0.CSIHnOFEC takes precedence over clearing to 0.
This bit is also initialized when CSIHnCTL0.CSIHnPWR is changed to 0 to 1 or 1 to 0.</td></tr>
<tr><td>13 to 9</td><td>Reserved</td><td>When read, the value after reset is returned.</td></tr>
<tr><td>8 of CSIH0</td><td>CSIHnHPST</td><td>Communication Priority Indication Flag
0: Indicates low-priority communication is in progress.
1: Indicates high-priority communication is in progress.
This bit always reads 0 if CPU-controlled high-priority communication is disabled (CSIHnPHE = 0).</td></tr>
<tr><td>8 of CSIH1</td><td>Reserved</td><td>When read, the value after reset is returned.</td></tr>
<tr><td>7</td><td>CSIHnTSF</td><td>Transfer Status Flag
0: Idle state
1: Communication is in progress or being prepared.
The timing to set or clear this bit is as in the following tables.
<table>
<tr><th rowspan="2">Master Mode</th><th colspan="2">Timing to set</th><th rowspan="2">Timing to clear</th></tr>
<tr><th>Direct Access Mode, FIFO Mode</th><th>Double Buffer Mode, Transmit-Only Mode</th></tr>
<tr><td>Transmit-only mode</td><td rowspan="3">Data is written to a transmit register (CSIHnTX0W/ CSIHnTX0H)</td><td rowspan="3">Bit CSIHnMCTL2.CSIHnBTST is set</td><td rowspan="3">Within a half clock of the last serial clock edge</td></tr>
<tr><td>Transmit/receive mode</td></tr>
<tr><td>Receive-only mode</td></tr>
</table>
<table>
<tr><th rowspan="2">Slave Mode</th><th colspan="2">Timing to set</th><th rowspan="2">Timing to clear</th></tr>
<tr><th>Direct Access Mode, FIFO Mode</th><th>Double Buffer Mode, Transmit-Only Mode</th></tr>
<tr><td>Transmit-only mode</td><td rowspan="2">Data is written to a transmit register (CSIHnTX0W/CSIHnTX0H)</td><td rowspan="2">Bit CSIHnMCTL2.CSIHnBTST is set</td><td rowspan="2">Within a half clock of the last serial clock edge</td></tr>
<tr><td>Transmit/receive mode</td></tr>
<tr><td>Receive-only mode</td><td>Input timing of CSIHTSCK</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>6</td><td>Reserved</td><td>When read, the value after reset is returned.</td></tr>
<tr><td>5</td><td>CSIHnFLF</td><td>A flag indicating that the buffer is full in FIFO mode.
0: FIFO buffer is not full.
1: FIFO buffer is full.
This bit is cleared by CSIHnSTCR0.CSIHnPCT.
The FIFO buffer might be filled with unsent data or received data.</td></tr>
</table>

Table 19.18 CSIHnSTR0 Register Contents (3/3)

Bit Position	Bit Name	Function
4	CSIHnEMF	A flag indicating that the buffer is empty in FIFO mode. 0: FIFO buffer is not empty. 1: FIFO buffer is empty. This bit is set to 1 by CSIHnSTCR0.CSIHnPCT. This bit is set to 1 when CSIHnSTR0.CSIHnSRP[7:0] + CSIHnSTR0.CSIHnSPF[7:0] = 00_H. The FIFO buffer might be filled with unsent data or received data.
3	CSIHnDCE	Data Consistency Check Error Flag 0: No data consistency error is detected. 1: Data consistency error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnDCEC. When setting to 1 by data consistency error detection and clearing to 0 by CSIHnSTCR0.CSIHnDCEC occur simultaneously, setting to 1 due to data consistency error detection takes precedence over clearing to 0. This bit is initialized when CSIHnCTL0.CSIHnPWR is changed from 0 to 1 or 1 to 0.
2	Reserved	When read, the value after reset is returned.
1	CSIHnPE	Parity Error Flag 0: No parity error is detected. 1: Parity error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnPEC. When setting to 1 due to parity error detection and clearing to 0 by CSIHnSTCR0.CSIHnPEC occur simultaneously, setting to 1 by parity error detection takes precedence over clearing to 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.
0	CSIHnOVE	Overrun Error Flag (Fixed to 0 in dual buffer mode) 0: No overrun error is detected. 1: Overrun error is detected. This bit is cleared by writing 1 to CSIHnSTCR0.CSIHnOVEC. When setting to 1 due to overrun error detection and clearing to 0 by writing to CSIHnSTCR0.CSIHnOVEC occur simultaneously, setting to 1 due to overrun error detection takes precedence over clearing to 0.This bit is Initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.

Table 19.19 Behavior in Memory ModeCSIH0CSIH0

Bit Name	Bit Position	Direct Access Mode	FIFO Mode	Transmit-Only Mode	Dual Buffer Mode
CSIHnSRP[7:0]	31 to 24	Fixed to 0	Number of received words	Fixed to 0	Fixed to 0
CSIHnSPF[7:0]	23 to 16	Fixed to 0	Number of unsent data	Fixed to 0	Fixed to 0
CSIHnTMOE	15	Fixed to 0	0: No error is detected. 1: An error is detected.	Fixed to 0	Fixed to 0
CSIHnOFE	14	Fixed to 0	0: No error is detected. 1: An error is detected.	Fixed to 0	Fixed to 0
CSIHnTSF	7	0: Idle state 1: Communication is in progress or being prepared			
CSIHnFLF	5	Fixed to 0	0: FIFO is not full 1: FIFO is full	Fixed to 0	Fixed to 0
CSIHnEMF	4	Fixed to 1	0: FIFO is not empty 1: FIFO is empty	Fixed to 1	Fixed to 1
CSIHnDCE	3	0: No error is detected. 1: An error is detected.			
CSIHnPE	1	0: No error is detected. 1: An error is detected.			
CSIHnOVE	0	0: No error is detected. 1: An error is detected.	0: No error is detected. 1: An error is detected.	0: No error is detected. 1: An error is detected.	Fixed to 0

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.6 CSIHnSTCR0 — CSIHn Status Clear Register 0

This register clears the status flags of the CSIHnSTR0 status register.

Access: This register can be read/written in 16-bit units.
When read, the value 0000_H is always returned.

Address: <CSIHn_base> + 0008_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn TMOEC	CSIHn OFEC	—	—	—	—	—	CSIHn PCT	—	—	—	—	CSIHn DCEC	—	CSIHn PEC	CSIHn OVEC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R/W	R	R	R	R	R/W	R	R/W	R/W

Table 19.20 CSIHnSTCR0 Register Contents

Bit Position	Bit Name	Function
15	CSIHnTMOEC	Controls the time-out error flag clear command. 0: No operation. The read value is always 0. 1: Clears the time out error flag (CSIHnSTR0.CSIHnTMOE).
14	CSIHnOFEC	Controls the overflow error flag clear command. 0: No operation. The read value is always 0. 1: Clears the overflow error flag (CSIHnSTR0.CSIHnOFE).
13 to 9	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
8	CSIHnPCT	Controls the FIFO pointer clear command. 0: No operation. The read value is always 0. 1: Clears the FIFO buffer pointers below (in FIFO mode, dual buffer mode, and transmit-only buffer mode) and the status bits. **FIFO Buffer Pointer** / **Status Bit** CSIHnMRWP0.CSIHnTRWA[6:0] / CSIHnSTR0.CSIHnSPF[7:0] CSIHnMRWP0.CSIHnRRA[6:0] / CSIHnSTR0.CSIHnSRP[7:0] CSIHnMCTL2.CSIHnSOP[6:0] / CSIHnSTR0.CSIHnFLF CSIHnSTR0.CSIHnTSF Additionally, the CSIHnSTR0.CSIHnEMF bit is set to 1 (FIFO empty) (in FIFO mode only).
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	CSIHnDCEC	Controls the data consistency error flag clear command. 0: No operation. The read value is always 0. 1: Clears the data consistency error flag (CSIHnSTR0.CSIHnDCE).
2	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
1	CSIHnPEC	Controls the parity error flag clear command. 0: No operation. The read value is always 0. 1: Clears the parity error flag (CSIHnSTR0.CSIHnPE).
0	CSIHnOVEC	Controls the overrun error flag clear command. 0: No operation. The read value is always 0. 1: Clears the overrun error flag (CSIHnSTR0.CSIHnOVE).

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.7 CSIHnMCTL0 — CSIHn Memory Control Register 0

This register selects the memory mode and the time-out setting.

Access: This register can be read/written in 16-bit units.

Address: <CSIHn_base> + 1040_H

Value after reset: $001F_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	CSIHn MMS[1:0]		—	—	—	CSIHnTO[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 19.21 CSIHnMCTL0 Register Contents

Bit Position	Bit Name	Function
15 to 10	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
9 to 8	CSIHnMMS [1:0]	Selects the memory mode. **CSIHn MMS1 / CSIHn MMS0 / Description** 0 / 0 / FIFO mode 0 / 1 / Dual buffer mode 1 / 0 / Transmit-only buffer mode 1 / 1 / Prohibited After a change of the memory mode, the respective buffer pointers must be cleared by setting the CSIHnSTCR0.CSIHnPCT bit to 1. In direct access mode, the setting of these bits is ignored.
7 to 5	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
4 to 0	CSIHnTO[4:0]	Selects the time-out setting in FIFO mode. **CSIHnTO[4:0] / Description** 00000_B / No time-out is detected 00001_B / Time-out is (1 × 8 × BRG output clocks) 00010_B / Time-out is (2 × 8 × BRG output clocks) ... 11111_B / Time-out is (31 × 8 × BRG output clocks) **CAUTION** Changing the time-out setting is only permitted when CSIHnCTL0.CSIHnPWR = 0. Set the CSIHnTO[4:0] bits to 00000_B in direct access mode, dual buffer mode, or transmit-only buffer mode (except FIFO mode). For details about time-out detection, see also Section 19.5.12.3, Time-Out Error.

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.8 CSIHnMCTL1 — CSIHn Memory Control Register 1

This register selects the conditions to generate the interrupt requests, INTCSIHTIC and INTCSIHTIR in FIFO mode.

Access: This register can be read/written in 32-bit units.

Address: <CSIHn_base> + 1000_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CSIHnFES[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CSIHnFFS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.22 CSIHnMCTL1 Register Contents

Bit Position	Bit name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
22 to 16	CSIHnFES[6:0]	Selects the conditions to generate the INTCSIHTIC interrupt (transmit data empty) in FIFO mode. When the number of unsent data to be transmitted in FIFO (checked by the CSIHnSTR0.CSIHnSPF[7:0] bit) and CSIHnMCTL1.CSIHnFES[6:0] match, the FIFO empty flag (CSIHnSTR0.CSIHnEMF bit) is set to 1, and the INTCSIHTIC interrupt request is generated.
15 to 7	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
6 to 0	CSIHnFFS[6:0]	Selects the conditions to generate the INTCSIHTIR interrupt (receive data full) in FIFO mode. When the number of received data in FIFO (checked by the CSIHnSTR0.CSIHnSRP[7:0] bit) and (128 - CSIHnMCTL1.CSIHnFFS[6:0]) match and the INTCSIHTIR interrupt request is generated.

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.9 CSIHnMCTL2 — CSIHn Memory Control Register 2

This register controls the operation of the memory in dual buffer or transmit-only buffer mode and triggers to start communication in buffer mode.

Access: This register can be read/written in 32-bit units.

Address: <CSIHn_base> + 1004_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHn BTST	—	—	—	—	—	—	—	CSIHnND[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CSIHnSOP[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.23 CSIHnMCTL2 Register Contents (1/2)

Bit Position	Bit Name	Function
31	CSIHnBTST	Provides a start trigger for buffer transfer. 0: No operation. 1: Issues the start transfer command. The read value is always 0. **CAUTION** This bit can only be used in dual buffer mode and transmit-only buffer mode.
30 to 24	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
23 to 16	CSIHnND[7:0]	Specifies the number of data for each memory mode. The read value indicates the number of remaining communication data.

CSIHn ND[7:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode
00_H	Send 0 data	Send 0 data	No influence	No influence
01_H	Send 1 data	Send 1 data	No influence	No influence
...	...	...	No influence	No influence
$3F_H$	Send 63 data	Send 63 data	No influence	No influence
40_H	Send 64 data	Send 64 data	No influence	No influence
...	Prohibited	...	No influence	No influence
$7F_H$	Prohibited	Send 127 data	No influence	No influence
80_H	Prohibited	Send 128 data	No influence	No influence
Other than the above	Setting is prohibited.			

The values are automatically decremented after data transfer (Not decremented in direct access mode).

15 to 7	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

Table 19.23 CSIHnMCTL2 Register Contents (2/2)

Bit Position	Bit Name	Function
6 to 0	CSIHnSOP[6:0]	Selects the pointer of the data to be sent. If communication is forced to stop by setting CSIHnCTL0.CSIHnPWR to 0 or CSIHnSTCR0.CSIHnPCT to 1, these bits are cleared by hardware. In FIFO mode, these bits indicate the send address.

CSIHn SOP[6:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode
00_H	0000_H	0000_H	0000_H	No influence
01_H	0004_H	0004_H	0004_H	No influence
...	...	...	...	No influence
$3F_H$	$00FC_H$	$00FC_H$	$00FC_H$	No influence
40_H	Prohibited	0100_H	0100_H	No influence
...	Prohibited	...	...	No influence
$7F_H$	Prohibited	$01FC_H$	$01FC_H$	No influence

CAUTION

In direct access mode, these bits are not incremented.

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.10 CSIHnMRWP0 — CSIHn Memory Read/Write Pointer Register 0

This register sets the pointers for reading from and writing to the dual buffer or transmit-only buffer.

Access: This register can be read/written in 32-bit units.

Address: <CSIHn_base> + 1018_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	CSIHnRRA[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	CSIHnTRWA[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.24 CSIHnMRWP0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
22 to 16	CSIHnRRA[6:0]	Selects the read pointer of the receive buffer.

CSIHn RRA[6:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode
00_H	0000_H	No influence	0000_H	No influence
01_H	0004_H	No influence	0004_H	No influence
...	...	No influence	...	No influence
$3F_H$	$00FC_H$	No influence	$00FC_H$	No influence
40_H	Prohibited	No influence	0100_H	No influence
...	Prohibited	No influence	...	No influence
$7F_H$	Prohibited	No influence	$01FC_H$	No influence

These bits are automatically incremented when received data is read.
If an overrun error is generated while reading the CSIHnRX0W or CSIHnRX0H register, the read pointer is not incremented.
These bits are cleared when CSIHnSTCR0.CSIHnPCT is set to 1.
In direct access mode and transmit-only buffer mode, these bits are not incremented.
To perform write access in transmit-only buffer mode, set 0000_H to these bits.
In FIFO mode, these bits indicate the read address of the received data.

15 to 7	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

Table 19.24 CSIHnMRWP0 Register Contents (2/2)

Bit Position	Bit Name	Function
6 to 0	CSIHnTRWA [6:0]	Selects the read/write pointer of the transmit buffer.

CSIHn TRWA[6:0]	Dual Buffer Mode	Transmit-Only Buffer Mode	FIFO Mode	Direct Access Mode
00_H	0000_H	0000_H	0000_H	No influence
01_H	0004_H	0004_H	0004_H	No influence
...	...	...	...	No influence
$3F_H$	$00FC_H$	$00FC_H$	$00FC_H$	No influence
40_H	Prohibited	0100_H	0100_H	No influence
...	Prohibited	...	...	No influence
$7F_H$	Prohibited	$01FC_H$	$01FC_H$	No influence

These bits are automatically incremented when the transmission data is written or read.
These bits are cleared when CSIHnSTCR0.CSIHnPCT is set to 1.
In direct access mode, these bits are not incremented.
In FIFO mode, these bits indicate the read/write address of transmission data.

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.11 CSIHnCFGx — CSIHn Configuration Register x

These up to eight registers specify for each chip select signal CSIHTCSSx prescaler, parity, data length, recessive configuration for broadcasting, serial data direction, clock and data phase, idle enforcement configuration, idle time, hold time, inter-data time, and setup time.

Slave mode

In slave mode, the transmission protocol setting of the CSIHnCFG0 register is effective.

- CSIHnPSx[1:0]: parity usage
- CSIHnDLSx[3:0]: data length selection
- CSIHnDIRx: data direction
- CSIHnCKPx, CSIHnDAPx: clock and data phase

In slave mode, set 0 to all the bits other than above in the CSIHnCFG0 register, and the CSIHnCFG1 to CSIHnCFG7 registers.

Access: This register can be read/written in 32-bit units.

Address: CSIHnCFG0: <CSIHn_base> + 1044_H
CSIHnCFG1: <CSIHn_base> + 1048_H
CSIHnCFG2: <CSIHn_base> + $104C_H$
CSIHnCFG3: <CSIHn_base> + 1050_H
CSIHnCFG4: <CSIHn_base> + 1054_H
CSIHnCFG5: <CSIHn_base> + 1058_H
CSIHnCFG6: <CSIHn_base> + $105C_H$
CSIHnCFG7: <CSIHn_base> + 1060_H

Value after reset: $0000\ 0000_H$

CSIH0:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIH0BRSSx[1:0]		CSIHnPSx[1:0]		CSIHnDLSx[3:0]				—	—	—	—	CSIHnRCBx	CSIHnDIRx	CSIHnCKPx	CSIHnDAPx
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnIDLx	CSIHnIDx[2:0]			CSIHnHDx[3:0]				CSIHnINx[3:0]				CSIHnSPx[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CSIH1:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIH1PSCLx [1:0]		CSIHnPSx[1:0]		CSIHnDLSx[3:0]				—	—	—	—	CSIHn RCBx	CSIHn DIRx	CSIHn CKPx	CSIHn DAPx
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHn IDLx	CSIHnIDx[2:0]			CSIHnHDx[3:0]				CSIHnINx[3:0]				CSIHnSPx[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.25 CSIHnCFGx Register Contents (1/5)

Bit Position	Bit Name	Function
31, 30 of CSIH0	CSIH0BRSSx [1:0]	These bits select the baud rate setting register (CSIH0BRSy)..

CSIH0 BRSSx1	CSIH0 BRSSx0	Baud Rate Setting Register Selection
0	0	The transfer clock frequency is set according to the CSIH0BRS0 setting.
0	1	The transfer clock frequency is set according to the CSIH0BRS1 setting.
1	0	The transfer clock frequency is set according to the CSIH0BRS2 setting.
1	1	The transfer clock frequency is set according to the CSIH0BRS3 setting.

The maximum value for setting the transfer clock frequency, combining the CSIH0CTL2.CSIH0PRS[2:0] setting, must be as follows:
Master mode: PCLK/4
Slave mode: PCLK/6

Bit Position	Bit Name	Function
31, 30 of CSIH1	CSIH1PSCLx [1:0]	Selects the prescaler for chip select x

CSIH0 PSCLx1	CSIH0 PSCLx0	Prescaler output
0	0	CSIHBCLK
0	1	CSIHBCLK / 2
1	0	CSIHBCLK / 4
1	1	CSIHBCLK / 8

These bits are only available in master mode.

Bit Position	Bit Name	Function
29, 28	CSIHnPSx[1:0]	Selects the parity for sending or receiving chip select signal x.

CSIHn PSx1	CSIHn PSx0	Transmission	Reception
0	0	No parity is transmitted	No parity is waited for.
0	1	Adds parity bit fixed to 0	Parity bit is waited for but not judged.
1	0	Adds odd parity only	Odd parity bit is waited for.
1	1	Adds even parity	Even parity bit is waited for.

Table 19.25 CSIHnCFGx Register Contents (2/5)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>27 to 24</td><td>CSIHnDLSx [3:0]</td><td>Selects the data length for chip select signal x.
<table>
<tr><th>CSIHnDLSx[3:0]</th><th>Data Length</th></tr>
<tr><td>0000_B</td><td>16 bits</td></tr>
<tr><td>0001_B</td><td>1 bit</td></tr>
<tr><td>0010_B</td><td>2 bits</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>1111_B</td><td>15 bits</td></tr>
</table>
CAUTION
Data length
• of 1 bit for CSIH0
• between 1 bit and 6 bits for CSIH1
When CSIHnTX0W.CSIHnEDL = 1, the setting of this bit has no effect.
When CSIHnTX0W.CSIHnEDL = 0 (the data length is 16 bits), the setting of this bit is valid. Only when the previous transmit data is 16 bits while CSIHnEDL = 1, writing 1 to this bit is enabled.</td></tr>
<tr><td>23 to 20</td><td>Reserved</td><td>When read, the value after reset is returned. When writing to these bits, write the value after reset.</td></tr>
<tr><td>19</td><td>CSIHnRCBx</td><td>Selects the recessive configuration for broadcasting for chip select x.
0: Dominant (higher priority)
1: Recessive (lower priority)
For details, see Section 19.5.3.1, Configuration Registers</td></tr>
<tr><td>18</td><td>CSIHnDIRx</td><td>Selects the serial data direction of chip select signal x.
0: Data is transmitted/received with MSB first.
1: Data is transmitted/received with LSB first.
For details, see Section 19.5.9, Serial Data Direction Selection.</td></tr>
</table>

Table 19.25 CSIHnCFGx Register Contents (3/5)

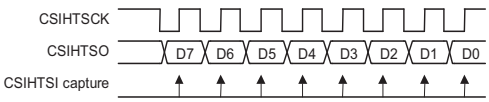
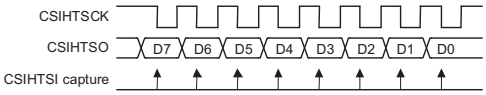
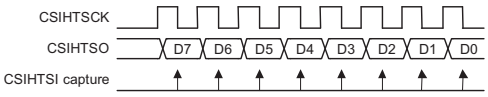
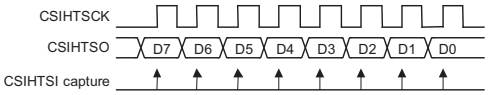
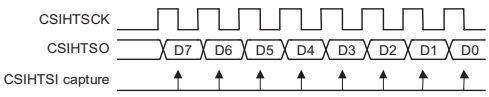
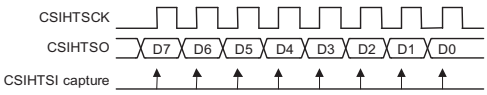
<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>17</td><td>CSIHnCKPx</td><td rowspan="2">CSIHnCKPx: Clock phase selection bit
CSIHnDAPx: Data phase selection bit
• CSIHnCTL1.CSIHnCKR = 0
<table><tr><th>CSIHn CKPx</th><th>CSIHn DAPx</th><th>Clock and Data Phase Selection</th></tr><tr><td>0</td><td>0</td><td>  </td></tr><tr><td>0</td><td>1</td><td>  </td></tr><tr><td>1</td><td>0</td><td>  </td></tr><tr><td>1</td><td>1</td><td>  </td></tr></table>• CSIHnCTL1.CSIHnCKR = 1
<table><tr><th>CSIHn CKPx</th><th>CSIHn DAPx</th><th>Clock and Data Phase Selection</th></tr><tr><td>0</td><td>0</td><td>  </td></tr><tr><td>0</td><td>1</td><td>  </td></tr><tr><td>1</td><td>—</td><td>Setting prohibited</td></tr></table></td></tr>
<tr><td>16</td><td>CSIHnDAPx</td></tr>
<tr><td>15</td><td>CSIHnIDLx</td><td>Selects the idle enforcement configuration for chip select x.
0: If the CSIHnTX0W.CSIHnCSx setting of two consecutive transfers is different, an idle state is inserted between two transfers. If the CSIHnTX0W.CSIHnCSx setting of two consecutive transfers is the same, an idle state is not inserted between two transfers.
1: Regardless of the CSIHnTX0W.CSIHnCSx setting of two consecutive transfers, an idle state is inserted between two transfers.
This bit is only available in master mode.
For details about the forced idle state, see Section 19.5.15, Enforced Chip Select Idle Setting.</td></tr>
</table>

Table 19.25 CSIHnCFGx Register Contents (4/5)

Bit Position	Bit Name	Function
14 to 12	CSIHnIDx[2:0]	Selects the idle time for chip select signal x. **CSIHn IDx[2:0]** / **Idle time** 000_B 0.5 transmission clock cycle 001_B 1.0 transmission clock cycle 010_B 1.5 transmission clock cycle 011_B 2.5 transmission clock cycle 100_B 3.5 transmission clock cycle 101_B 4.5 transmission clock cycle 110_B 6.5 transmission clock cycle 111_B 8.5 transmission clock cycle These bits are only available in master mode.
11 to 8	CSIHnHDx[3:0]	Specifies the hold time for chip select signal x in transmission clock cycles. **CSIHn HDx[3:0]** / **Hold time when CSIHnCTL1.CSIHnSIT is 0** / **Hold time when CSIHnCTL1.CSIHnSIT is 1** 0000_B 0.5 transmission clock cycle / 1.0 transmission clock cycle 0001_B 1.0 transmission clock cycle / 1.5 transmission clock cycle 0010_B 1.5 transmission clock cycle / 2.0 transmission clock cycle 0011_B 2.5 transmission clock cycle / 3.0 transmission clock cycle 0100_B 3.5 transmission clock cycle / 4.0 transmission clock cycle 0101_B 4.5 transmission clock cycle / 5.0 transmission clock cycle 0110_B 6.5 transmission clock cycle / 7.0 transmission clock cycle 0111_B 8.5 transmission clock cycle / 9.0 transmission clock cycle 1000_B 9.5 transmission clock cycle / 10.0 transmission clock cycle 1001_B 10.5 transmission clock cycle / 11.0 transmission clock cycle 1010_B 11.5 transmission clock cycle / 12.0 transmission clock cycle 1011_B 12.5 transmission clock cycle / 13.0 transmission clock cycle 1100_B 14.5 transmission clock cycle / 15.0 transmission clock cycle 1101_B 16.5 transmission clock cycle / 17.0 transmission clock cycle 1110_B 18.5 transmission clock cycle / 19.0 transmission clock cycle 1111_B 20.5 transmission clock cycle / 21.0 transmission clock cycle These bits are only available in master mode.

Table 19.25 CSIHnCFGx Register Contents (5/5)

Bit Position	Bit Name	Function
7 to 4	CSIHnINx[3:0]	Specifies the inter-data time for chip select signal x in transmission clock cycles.

CSIHn INx[3:0]	Inter-Data Delay when CSIHnCTL1.CSIHnSIT is 0	Inter-Data Delay when CSIHnCTL1.CSIHnSIT is 1
0000_B	0.0 transmission clock cycle	0.5 transmission clock cycle
0001_B	0.5 transmission clock cycle	1.0 transmission clock cycle
0010_B	1.0 transmission clock cycle	1.5 transmission clock cycle
0011_B	2.0 transmission clock cycle	2.5 transmission clock cycle
0100_B	3.0 transmission clock cycle	3.5 transmission clock cycle
0101_B	4.0 transmission clock cycle	4.5 transmission clock cycle
0110_B	6.0 transmission clock cycle	6.5 transmission clock cycle
0111_B	8.0 transmission clock cycle	8.5 transmission clock cycle
1000_B	9.0 transmission clock cycle	9.5 transmission clock cycle
1001_B	10.0 transmission clock cycle	10.5 transmission clock cycle
1010_B	11.0 transmission clock cycle	11.5 transmission clock cycle
1011_B	12.0 transmission clock cycle	12.5 transmission clock cycle
1100_B	14.0 transmission clock cycle	14.5 transmission clock cycle
1101_B	16.0 transmission clock cycle	16.5 transmission clock cycle
1110_B	18.0 transmission clock cycle	18.5 transmission clock cycle
1111_B	20.0 transmission clock cycle	20.5 transmission clock cycle

These bits are only available in master mode.

3 to 0	CSIHnSPx[3:0]	Specifies the setup time for chip select signal x in transmission clock cycles.

CSIHnSPx[3:0]	Setup Time
0000_B	0.5 transmission clock cycle
0001_B	1.0 transmission clock cycle
0010_B	1.5 transmission clock cycle
0011_B	2.5 transmission clock cycle
0100_B	3.5 transmission clock cycle
0101_B	4.5 transmission clock cycle
0110_B	6.5 transmission clock cycle
0111_B	8.5 transmission clock cycle
1000_B	9.5 transmission clock cycle
1001_B	10.5 transmission clock cycle
1010_B	11.5 transmission clock cycle
1011_B	12.5 transmission clock cycle
1100_B	14.5 transmission clock cycle
1101_B	16.5 transmission clock cycle
1110_B	18.5 transmission clock cycle
1111_B	20.5 transmission clock cycle

These bits are only available in master mode.

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.12 CSIHnTX0W — CSIHn Transmit Data Register 0 for Word Access

This register stores transmission data. In addition, it specifies the communication interrupt request, the end-of-job, the extended data length, and the chip select activation.

Access: This register can be read/written in 32-bit units.

Address: <CSIHn_base> + 1008_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CSIHnCIRE	CSIHnEOJ	CSIHnEDL	—	—	—	—	—	CSIHnCS7	CSIHnCS6	CSIHnCS5	CSIHnCS4	CSIHnCS3	CSIHnCS2	CSIHnCS1	CSIHnCS0
Value after reset	—	—	—	0	0	0	0	0	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnTX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.26 CSIHnTX0W Register Contents (1/2)

Bit Position	Bit name	Function
31	CSIHnCIRE	Enables the communication interrupt request INTCSIHTIC in dual buffer or transmit-only buffer mode, or the job completion interrupt INTCSIHTJC request in FIFO mode. 0: No interrupt is requested. 1: An interrupt is requested. Generates interrupt INTCSIHTIC or INT_CSIHTIJC after transmission. For details, see **Section 19.4.3, INTCSIHTIC (Communication Status Interrupt)** and **Section 19.4.6, INTCSIHTIJC (Job Completion Interrupt)**. **CAUTION** This bit is only valid when job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
30	CSIHnEOJ	Specifies the end of a job. 0: Indicates that the job does not end. The job continues. 1: Indicates end-of-job data. **CAUTION** This bit is only valid when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). This bit must be set to 0 in slave mode.
29	CSIHnEDL	Specifies whether the associated data requires the extended data length (EDL) option. 0: Normal operation 1: Enables the extended data length. The associated data is transmitted as a 16-bit packet. No inter-data time or idle time will be inserted after the data is transmitted. If CSIHnCTL1.CSIHnEDLE = 1 and CSIHnTX0W.CSIHnEDL = 1, the subsequent data must have the same CS selection. If CS is modified for the subsequent data, the correct operation is not assured. **CAUTION** This bit is only available if CSIHnCTL1.CSIHnEDLE = 1.

Table 19.26 CSIHnTX0W Register Contents (2/2)

Bit Position	Bit name	Function
28 to 24	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
23 to 16	CSIHnCS[7:0]	Activates one or several chip select signals. 0: Activates chip select signals x for the associated transmission. 1: Deactivates chip select signals x for the associated transmission. Setting CSIHnTX0W.CSIHnCS[7:0] = FF_H is prohibited. **CAUTION** If several chip select signals are enabled for broadcasting, the configuration of one with CSIHnCFGx.CSIHnRCBx = 0 (dominant) is used. In this case, all dominant chip select signals must be set to precisely the same value. In slave mode, set the CSIHnCS[7:0] bit to FE_H.
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.13 CSIHnTX0H — CSIHn Transmit Data Register 0 for Half Word Access

This register stores the transmission data. This register is the same as bits 15 to 0 of register CSIHnTX0W.

The upper 16 bits of CSIHnTX0W are applied for transfer. Set transmit data to CSIHnTX0W before using this register because the value of CSIHnTX0W is undefined after the reset.

Access: This register can be read/written in 16-bit units.

Address: <CSIHn_base> + $100C_H$

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnTX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.27 CSIHnTX0H Register Contents

Bit Position	Bit name	Function
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.14 CSIHnRX0W — CSIHn Receive Data Register 0 for Word Access

This register stores the received data.

Access: This register can only be read in 32-bit units.

Address: <CSIHn_base> + 1010_H

Value after reset: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	CSIHn RPE	CSIHn TDCE	CSIHn CS7	CSIHn CS6	CSIHn CS5	CSIHn CS4	CSIHn CS3	CSIHn CS2	CSIHn CS1	CSIHn CS0
Value after reset	0	0	0	0	0	0	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnRX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.28 CSIHnRX0W Register Contents

Bit Position	Bit name	Function
31 to 26	Reserved	When read, the value after reset is returned.
25	CSIHnRPE	Indicates whether a reception data parity error was detected. 0: No parity error was detected on the associated reception data. 1: A parity error was detected on the associated reception data.
24	CSIHnTDCE	Indicates whether a transmission data consistency error was detected. 0: No consistency error was detected on the associated transmission data. 1: A consistency error was detected on the associated transmission data.
23 to 16	CSIHnCSx (x = 7 to 0)	Indicates which chip select signal was activated. 0: Chip select x was activated for the associated reception. 1: Chip select x was deactivated for the associated reception.
15 to 0	CSIHnRX[15:0]	Stores the received data.

NOTE

To read this register, do so one serial clock cycle before an interrupt is generated.

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.15 CSIHnRX0H — CSIHn Receive Data Register 0 for Half Word Access

This register stores the received data. This register is the same as bits 15 to 0 of register CSIHnRX0W.

Access: This register can only be read in 16-bit units.

Address: <CSIHn_base> + 1014_H

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CSIHnRX[15:0]															
Value after reset	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 19.29 CSIHnRX0H Register Contents

Bit Position	Bit name	Function
15 to 0	CSIHnRX[15:0]	Stores the received data.

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.16 CSIHnEMU — CSIHn Emulation Register

This register controls operation of SVSTOP.

Access: This register can be read/written in 8-bit or 1-bit units.
Perform write operation when (EPC.SVSTOP = 0).

Address: <CSIHn_base> + 0018_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CSIHnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 19.30 CSIHnEMU Register Contents

Bit Position	Bit name	Function
7	CSIHn SVSDIS	Selects to continue or stop transmit/receive operation during debugging. • When the EPC.SVSTOP bit is set to 0 Continues transmit/receive operation regardless of the setting of this bit. • When the EPC.SVSTOP bit is set to 1 0: Stops transmit/receive operation. 1: Continues transmit/receive operation.
6 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.17 CSIH0BRSy — CSIH0 Baud Rate Setting Register y (y = 0 to 3) (CSIH0 only)

This register sets the transfer clock frequency for each chip select signal.

With CSIH0CFG0 to 7.CSIH0BRSSx[1:0] bits, one of the four types of transfer clock frequency settings can be selected for each chip select signal. For details of transfer clock frequency setting, see Section 19.5.5, Transmission Clock Selection.

Access: This register can be read/written in 16-bit units.

Address: CSIH0BRS0: <CSIH0_base> + 1068_H
CSIH0BRS1: <CSIH0_base> + $106C_H$
CSIH0BRS2: <CSIH0_base> + 1070_H
CSIH0BRS3: <CSIH0_base> + 1074_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CSIH0BRS[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19.31 CSIH0BRSy Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
11 to 0	CSIH0BRS [11:0]	0: BRG stopped 1: PCLK / (2^{α} × 1 × 2) 2: PCLK / (2^{α} × 2 × 2) 3: PCLK / (2^{α} × 3 × 2) 4: PCLK / (2^{α} × 4 × 2) . . . 4095: PCLK / (2^{α} × 4095 × 2) α is the value of CSIH0CTL2.CSIH0PRS[2:0].

CAUTION

For the setting of this register, see Table 19.32, Notes on Setting Registers.

19.3.18 List of Caution

Table 19.32 Notes on Setting Registers (1/3)

Register	Bit	Content
CSIHnCTL0	CSIHnPWR	If this bit is cleared during communication, ongoing communication is suspended. After cancelling the suspension, it is necessary to restart the communication.
CSIHnCTL0	CSIHnTXE CSIHnRXE	Do not modify any of these bits while CSIHnCTL0.CSIHnPWR = 0. (These bits can be modified at the same time with the CSIHnCTL0.CSIHnPWR bit.) Do not modify these bits while CSIHnSTR0.CSIHnTSF = 1, because the specified operation is not guaranteed if ongoing communication is suspended.
CSIHnCTL0	CSIHnJOBE	Do not modify this bit while CSIHnCTL0.CSIHnPWR = 0. This bit is only valid when CSIHnCTL1.CSIHnJE = 1. Setting this bit is prohibited in slave mode.
CSIHnCTL0	CSIHnMBS	Do not modify this bit while CSIHnCTL0.CSIHnPWR = 0. (This bit can modify at the same time with the CSIHnCTL0.CSIHnPWR bit.) Modification of this bit is only allowed while CSIHnSTR0.CSIHnTSF = 0. Do not change the mode between FIFO mode and direct access mode while CSIHnCTL0.CSIHnPWR = 1. When the CPU-controlled high-priority communication function is enabled, the operation is the same as that in direct access mode regardless of the CSIHnMBS bit setting.
CSIHnCTL1	CSIHnCKR	Modification of this bit is only allowed while CSIHnCTL0.CSIHnPWR = 0. When CS is not used, use this bit instead of CSIHnCFGx.CSIHnCKPx and set CSIHnCFGx.CSIHnCKPx to 0. This bit must be used in slave mode.
CSIHnCTL1	CSIHnSLIT CSIHnCSL[7:0] CSIHnEDLE CSIHnDCS CSIHnCSRI CSIHnHSE	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0.
CSIHnCTL1	CSIHnPHE CSIHnJE CSIHnLBM	Modification of these bits is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting of this bit is prohibited in slave mode.
CSIHnCTL1	CSIHnSSE	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting of this bit is prohibited in master mode.
CSIHnCTL1	CSIHnSIT	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. This bit is only valid in master mode. In slave mode, no delay is generated.
CSIHnCTL2	CSIHnPRS[2:0]	Modification of this bit is only permitted while CSIHnCTL0.CSIHnPWR = 0. Setting of the max transfer clock frequency is as follows. • Master mode: up to PCLK/4 • Slave mode: up to PCLK/6
CSIHnSTR0	CSIHnSRP[7:0] CSIHnSPF[7:0] CSIHnFLF CSIHnEMF CSIHnTSF CSIHnHPST	Writing these bits is prohibited, and only reading is permitted.
CSIHnSTR0	CSIHnTMOE CSIHnOFE CSIHnDCE CSIHnPE CSIHnOVE	Writing these bits is prohibited, and only reading is permitted. These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0.
CSIHnSTCR0	CSIHnPCT	If this bit is set to1 during communication, ongoing communication is suspended.
CSIHnMCTL0	CSIHnMMS[1:0]	Modification of these bits is only allowed while CSIHnCTL0.CSIHnPWR = 0 and CSIHnCTL0.CSIHnMBS = 0.

Table 19.32 Notes on Setting Registers (2/3)

Register	Bit	Content
CSIHnMCTL0	CSIHnTO[4:0]	Modification of these bits is only allowed while CSIHnCTL0.CSIHnPWR = 0. Set these bits to 0 in master mode. Set these bits to 0 in direct access, dual buffer, and transmit-only buffer mode.
CSIHnMCTL1	CSIHnFES[6:0] CSIHnFFS[6:0]	Write to these bits while communication is permitted.
CSIHnMCTL2	CSIHnBTST CSIHnND[7:0] CSIHnSOP[6:0]	Writing these bits is prohibited when CSIHnCTL0.CSIHnPWR = 0. Writing these bits is prohibited when CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0. Writing these bits is prohibited when CSIHnSTR0.CSIHnTSF = 1. Writing these bits is prohibited in direct access or FIFO mode.
CSIHnMRWP0	CSIHnRRA[6:0]	Write to these bits while communication is permitted. Writing these bits is prohibited in direct access or FIFO mode. When writing is required, set "0000_H" to these bits in transmit only buffer mode.
CSIHnMRWP0	CSIHnTRWA[6:0]	Write to these bits while communication is permitted. Writing these bits is prohibited in direct access or FIFO mode.
CSIHnCFGx x = 0	CSIHnBRSSx[1:0] CSIHnRCBx CSIHnIDLx CSIHnIDx[2:0] CSIHnHDx[3:0] CSIHnINx[3:0] CSIHnSPx[3:0]	Modification of these bits is only allowed while CSIHnCTL0.CSIHnPWR = 0. These bits must be set to 0 in slave mode.
CSIHnCFGx x = 0	CSIHnPSx[1:0] CSIHnDLSx[3:0] CSIHnDIRx CSIHnDAPx	Modification of these bits is only allowed while CSIHnCTL0.CSIHnPWR = 0. In slave mode, the CSIHnCFG0 setting is used for the configuration. Therefore, all the bits in CSIHnCFG1 to CSIHnCFG7 must be set to 0.
CSIHnCFGx x = 0	CSIHnCKPx	Modification of this bit is only allowed while CSIHnCTL0.CSIHnPWR = 0. As CSIHnCTL1.CSIHnCKR must be used, set this bit to 0 in slave mode. If CS is not used, set the CSIHnCTL1.CSIHnCKR bit instead of this bit, and clear this bit to 0.
CSIHnTX0W	CSIHnEOJ CSIHnCIRE	This bit is only valid when CSIHnCTL1.CSIHnJE = 1. While CSIHnCTL1.CSIHnJE = 0, the value of this bit is ignored even if 1 is read. Set these bits to 0 in slave mode.
CSIHnTX0W	CSIHnEDL	This bit is only valid when CSIHnCTL1.CSIHnEDLE = 1. While CSIHnCTL1.CSIHnEDLE = 0, the value of this bit is ignored even if 1 is read.
CSIHnTX0W	CSIHnCS[7:0]	In master mode, setting this bit to "FF_H" is prohibited. In slave mode, set this bit to "FE_H".
CSIHnTX0W CSIHnTX0H		Reading to these bits while communication is prohibited in FIFO mode. While CSIHnCTL0.CSIHnPWR = 0 and FIFO mode is on, reading and writing these bits are prohibited. While CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0, writing to these bits are prohibited in direct access mode.
CSIHnRX0W		These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0. While CSIHnCTL0.CSIHnPWR = 0, reading and writing to these bits is prohibited in FIFO mode. While CSIHnCTL0.CSIHnPWR = 0, writing and reading to these bits is valid in direct access, dual buffer, and transmit-only buffer mode. While CSIHnCTL0.CSIHnPWR = 1, writing to these bits is invalid and reading to these bits is valid.

Table 19.32 Notes on Setting Registers (3/3)

Register	Bit	Content
CSIHnRX0H		These bits are initialized when CSIHnCTL0.CSIHnPWR = 0 → 1 or CSIHnCTL0.CSIHnPWR = 1 → 0. While CSIHnCTL0.CSIHnPWR = 0, reading and writing to these bits is prohibited in FIFO mode. While CSIHnCTL0.CSIHnPWR = 1, writing to these bits is invalid and reading to these bits is valid in FIFO mode. Regardless of CSIHnCTL0.CSIHnPWR value, writing to these bits invalid and reading to these bits is valid in direct access, dual buffer, and transmit-only buffer mode.
CSIHnEMU	CSIHnSVSDIS	Modification of this bit is only permitted while SVSTOP = 0.
CSIH0BRSy y = 0 to 3		Modification of this bit is only permitted while CSIH0CTL0.CSIH0PWR = 0.

19.4 Interrupt Sources

CSIH can generate the following interrupt requests:

- INTCSIHTIC (communication status interrupt)
- INTCSIHTIR (reception status interrupt)
- INTCSIHTIRE (communication error interrupt)
- INTCSIHTIJC (job completion interrupt)

19.4.1 Overview

The communication error interrupt INTCSIHTIRE is generated when an error is detected. The generation of the other interrupts depends on the memory mode, the job mode, and – in case of the job completion interrupt INTCSIHTIJC –also the operating mode.

The job completion interrupt INTCSIHTIJC is only generated when job mode is enabled (CSIHnCTL1.CSIHnJE = 1). It is not available in slave mode.

The following table gives an overview.

Table 19.33 Interrupt Generation (1/2)

		Cause of Interrupt	
Memory Mode	Interrupt	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	INTCSIHTIC	Tx data empty*1 except job abort*4	Tx data empty*1 except job abort*4
	INTCSIHTIR	Rx data full*2 *and* CSIHnCTL0.CSIHnRXE = 1	Rx data full*2 *and* CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*3	Not applicable	CSIHnTX0W.CSIHnCIRE = 1 (except Tx data empty), or job abort*4
Transmit-only buffer	INTCSIHTIC	End of communication	CSIHnTX0W.CSIHnCIRE = 1 and (CSIHnCTL0.CSIHnJOBE = 0 or CSIHnTX0W.CSIHnEOJ = 0)
	INTCSIHTIR	Data received *and* CSIHnCTL0.CSIHnRXE = 1	Data received *and* CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*3	Not applicable	Job abort*4

Table 19.33 Interrupt Generation (2/2)

Memory Mode	Interrupt	Cause of Interrupt	
		Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
Dual buffer	INTCSIHTIC	End of communication	CSIHnTX0W.CSIHnCIRE = 1 and (CSIHnCTL0.CSIHnJOBE = 0 or CSIHnTX0W.CSIHnEOJ = 0)
	INTCSIHTIR	End of communication *and* CSIHnCTL0.CSIHnRXE = 1	Data received *and* CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*3	Not applicable	Job abort*4
Direct access	INTCSIHTIC	One single data transfer	One data transfer except the state of job abort*4
	INTCSIHTIR	Data received *and* CSIHnCTL0.CSIHnRXE = 1	Data received *and* CSIHnCTL0.CSIHnRXE = 1
	INTCSIHTIRE	Error detected	Error detected
	INTCSIHTIJC*3	Not applicable	Job abort*4

Note 1. "Tx data empty" refers to the FIFO fill level, defined by CSIHnMCTL1.CSIHnFES[6:0].
Note 2. "Rx data full" refers to the FIFO fill level, defined by CSIHnMCTL1.CSIHnFFS[6:0].
Note 3. INTCSIHTIJC is not available in slave mode.
Note 4. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1.
CSIH0 only: During high priority communication in transmit-only buffer mode, the operation is the same as that in direct access mode.

19.4.2 Interrupt Delay

In master mode, all interrupts generated by the master can be delayed by one half period of the transmission clock, CSIHTSCK. This is not possible in slave mode.

The delay is specified by setting bit CSIHnCTL1.CSIHnSIT = 1. (The setting of the CSIHnSIT bit is invalid in slave mode.)

The following example illustrates the interrupt delay function, assuming a setting of CSIHnCTL1.CSIHnSIT = 1 (interrupt delay enabled), CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0 (clock and data phase), and CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B (data length 8 bits).

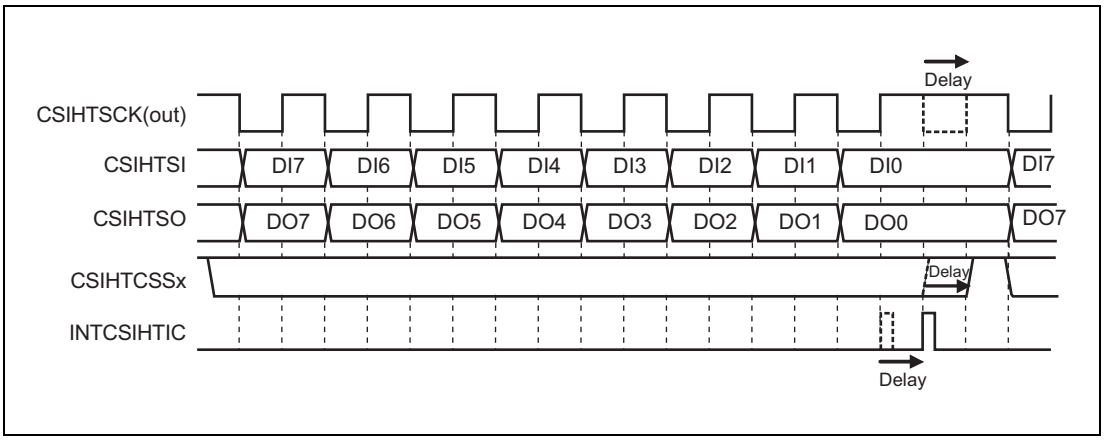


Figure 19.2 Interrupt Delay Function (CSIHnCTL1.CSIHnSIT = 1)

Setting CSIHnCTL1.CSIHnSIT = 1 adds half period delay to the transmission clock. This delays also the end of the present chip select signal (CSIHTCSSx).

19.4.3 INTCSIHTIC (Communication Status Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the conditions shown in the following table.

Table 19.34 INTCSIHTIC Interrupt Generation

	Cause of Interrupt	
Memory Mode	**Job Mode Disabled CSIHnCTL1.CSIHnJE = 0**	**Job Mode Enabled CSIHnCTL1.CSIHnJE = 1**
FIFO	This interrupt is generated just before transmission data is about to be missing in the FIFO, indicating to the application that new data should be added. INTCSIHTIC is generated, if the number of data to be sent remaining in the FIFO (CSIHnSTR0.CSIHnSPF[7:0]) equals CSIHnMTCL1.CSIHnFES[6:0].	Similar to “when JE is 0”, an interrupt is generated when the number of transmit data remained in the FIFO CSIHnSTR0.CSIHnSPF[7:0] is the same number as CSIHnMCTL1.CSIHnFES[6:0]. At the time of job abort, no interrupt is generated.
Transmit-only buffer, dual buffer	Generated at the End of communication. (Specified by the CSIHnMTLC2.CSIHnND[7:0] bit)	Generated when data with CSIHnTX0W.CSIHnCIRE = 1 is sent. Note that if data with CSIHnTX0W.CSIHnCIRE = 1 and job abort*1 are sent, the INTCSIHTIJC interrupt is generated instead of INTCSIHTIC.
Direct access	Generated after every data transfer.	Generated after every data transfer, except when the communication was aborted.

Note 1. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1.
CSIH0 only: During high priority communication in transmit-only buffer mode, the operation is the same as that in direct access mode.

19.4.3.1 INTCSIHTIC in Direct Access Mode

The examples below show the INTCSIHTIC behavior in direct access mode.

The examples assume:

- Master mode
- Direct access mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

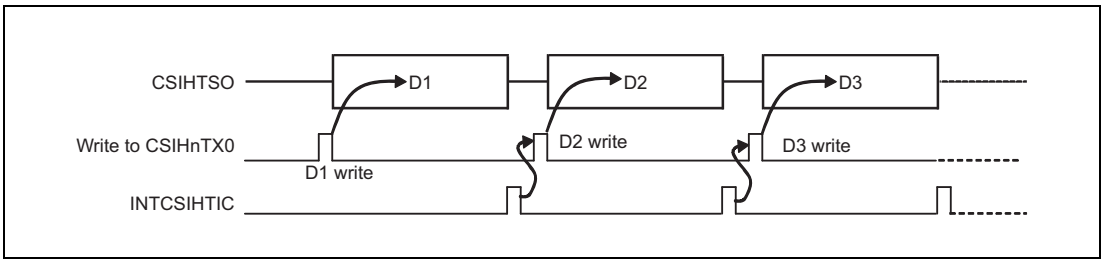


Figure 19.3 Generation of INTCSIHTIC after Transfer (CSIHnCTL1.CSIHnSLIT = 0)

If job mode is enabled (CSIHnCTL1.CSIHnJE = 1) and a job ends because data is sent with CSIHnTXOW.CSIHnEOJ = 1 and communication stop is requested (CSIHnCTL0.CSIHnJOBE = 1), then INTCSIHTIC is replaced by the job completion interrupt INTCSIHTIJC.

INTCSIHTIC can also be set up to occur as soon as the CSIHnTX0W/H register is free for the next data. This is specified by setting CSIHnCTL1.CSIHnSLIT = 1.

The effect is illustrated in the figure below.

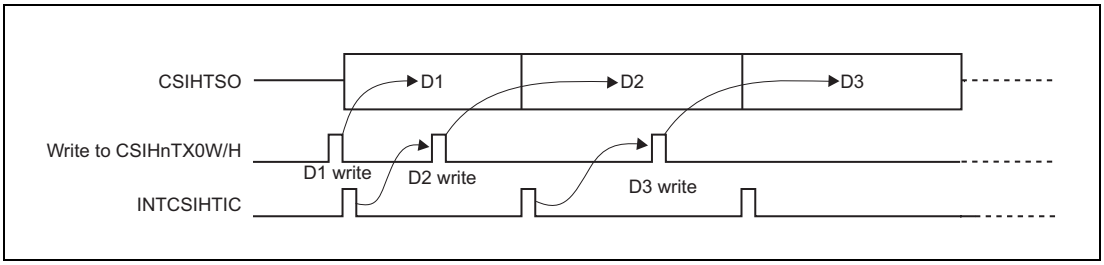


Figure 19.4 Immediate Generation of INTCSIHTIC (CSIHnCTL1.CSIHnSLIT = 1)

Thus, the new data can be written in advance.

NOTE

CSIH0 only: During high priority communication in transmit-only buffer mode, the operation is in the same way as direct access mode.

19.4.3.2 INTCSIHTIC in FIFO Mode

The example below shows the INTCSIHTIC behavior in FIFO mode.

The example assumes:

- Master mode
- FIFO mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

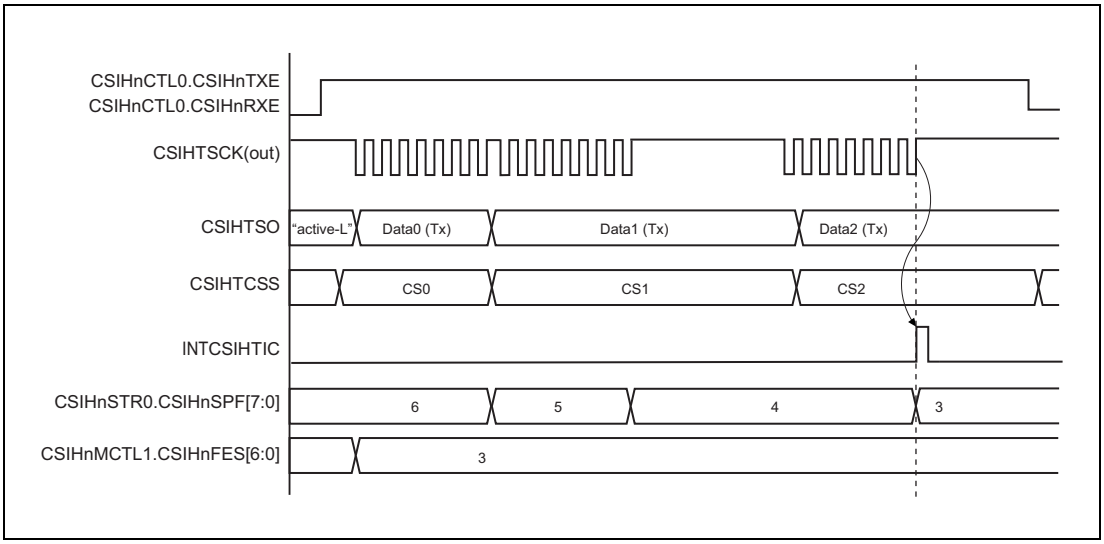


Figure 19.5 Generation of INTCSIHTIC in FIFO Memory Mode

The condition for “FIFO empty” is specified in CSIHnMCTL1.CSIHnFES[6:0]. In the example of the diagram above, the number of unsent data in FIFO is set to 3.

CSIHnSTR0.CSIHnSPF[7:0] indicates the number of unsent data. When both match, the interrupt INTCSIHTIC occurs.

19.4.3.3 INTCSIHTIC in Job Mode

The example below shows the INTCSIHTIC behavior in job mode.

The example assumes:

- Master mode
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

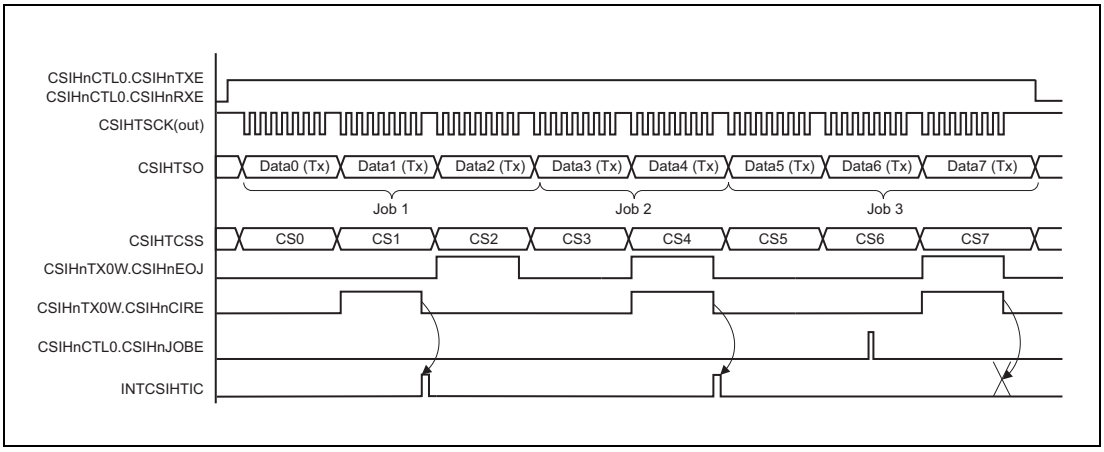


Figure 19.6 Generation of INTCSIHTIC in Job Mode

The rules for generating INTCSIHTIC in job mode are shown in the following table.

Table 19.35 Generation of INTCSIHTIC in Job Mode

CSIHnTX0W. CSIHnEOJ	CSIHnTX0W. CSIHnCIRE	INTCSIHTIC
0	0	Not generated
0	1	Generated
1	0	Not generated
1	1	CSIHnCTL0.CSIHnJOBE = 0: Generated
		CSIHnCTL0.CSIHnJOBE = 1: Not generated, replaced by interrupt INTCSIHTIJC

19.4.4 INTCSIHTIR (Reception Status Interrupt)

Depending on the memory mode and the job mode, this interrupt is generated according to the conditions below.

Table 19.36 INTCSIHTIR Interrupt Generation

Memory Mode	Cause of Interrupt	
	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	This interrupt occurs when CSIHnCTL0.CSIHnRXE is 1 and the FIFO buffer is almost full with received data, indicating to the application that the FIFO must be emptied. INTCSIHTIR is generated, if the number of received data in the FIFO (CSIHnSTR0.CSIHnSRP[7:0]) equals (128 – CSIHnMCTL1.CSIHnFFS[6:0]).	
Dual buffer	Generated when the communication has finished (as specified by the CSIHnMCTL2.CSIHnND[7:0] bit) and CSIHnCTL0.CSIHnRXE = 1.	Generated after every data transfer.
Transmit-only buffer, Direct access	Generated after every data transfer.	

19.4.4.1 INTCSIHTIR in Direct Access Mode

The example below shows the INTCSIHTIR behavior in direct access mode.

The example below assumes:

- Master mode
- Direct access mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

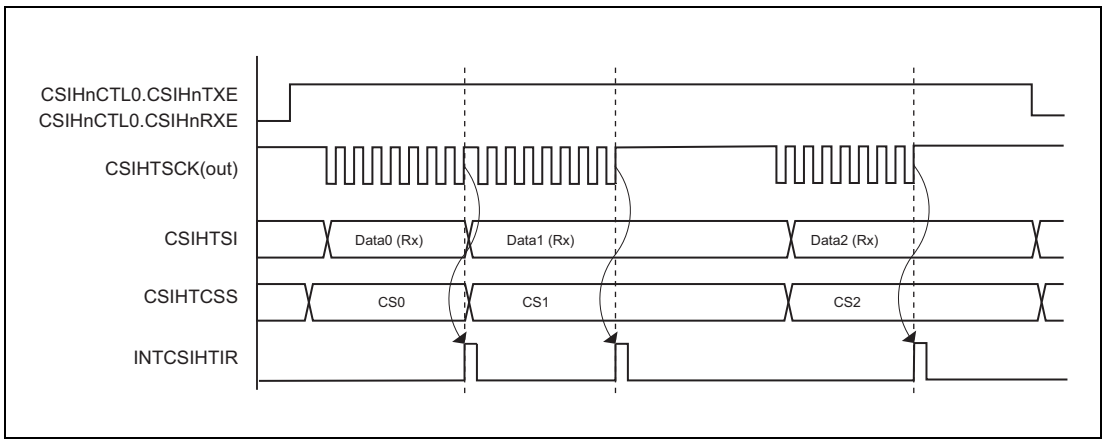


Figure 19.7 Generation of INTCSIHTIR in Direct Access Memory Mode

19.4.4.2 INTCSIHTIR in Dual Buffer Mode

The example below shows the INTCSIHTIR behavior in dual buffer mode.

The example assumes:

- Master mode
- Dual buffer mode
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Default clock and data phase
  (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

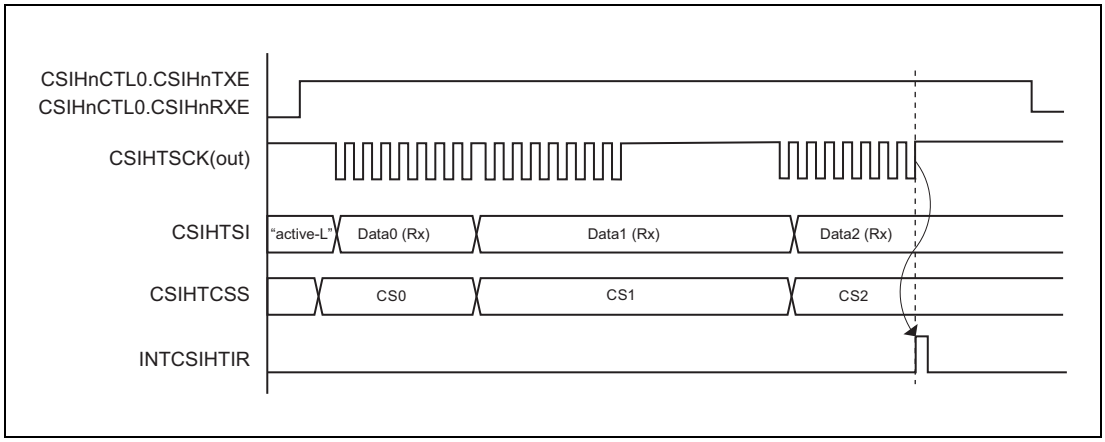


Figure 19.8 Generation of INTCSIHTIR in Dual Buffer Mode

19.4.5 INTCSIHTIRE (Communication Error Interrupt)

This interrupt is generated whenever an error is detected.

For details about generation interruption timing, see Section 19.5.12, Error Detection.

Table 19.37 Data Error Types

Error Type	Communication Status after Error Interrupt	Comment
FIFO overflow error	Interrupt is generated and communication continues.	The data are not written to the FIFO buffer and the overflow of data is lost, but communications started before the error is continued.
Parity error	Interrupt is generated and communication continues.	—
Data consistency error	Interrupt is generated and communication continues.	—
Time-out error	Interrupt is generated and communication continues.	—
Overrun error	**Condition for errors 1:** In FIFO mode, when the number of received data is 0 and CPU reads the CSIHnRX0W/H register, an interrupt is generated and communication continues.	—
	Condition for errors 2: In slave mode, when CSIHnCTL1.CSIHnHSE = 0 (handshake function disabled): (1) In direct access mode or transmit-only buffer mode, when reception is completed while the previous received data is remained in the CSIHnRX0W/H register, an interrupt is generated, and communication continues. (2) In FIFO mode, when reception by the FIFO buffer is completed and the buffer is in the full state, an interrupt is generated. Communication continues.	In slave mode, when CSIHnCTL1.CSIHnHSE = 1 (handshake function enabled), communication is suspended due to handshake, an overrun error is not generated.

The type of error that caused the generation of INTCSIHTIRE is flagged in register CSIHnSTR0.

Additionally a parity error flag and a data consistency error flag are attached to the received data in CSIHnRX0W.

For details about the various error types, see Section 19.5.12, Error Detection.

19.4.6 INTCSIHTIJC (Job Completion Interrupt)

This interrupt supports the handling of jobs, see Section 19.5.3.3, Job Concept. This interrupt is only available in master mode.

Job mode is enabled by setting CSIHnCTL1.CSIHnJE = 1. When CSIHnCTL1.CSIHnJE = 0, INTCSIHTIJC is not generated.

Depending on the memory mode, this interrupt is generated according to the condition shown in the following table.

Table 19.38 INTCSIHTIJC Interrupt Generation

Memory Mode	Job Mode Disabled CSIHnCTL1.CSIHnJE = 0	Job Mode Enabled CSIHnCTL1.CSIHnJE = 1
FIFO	Not applicable	Indicates that the communication has stopped at the end of a job after a job abort*1 was triggered If FIFO empty is not detected and when CSIHnCIRE is 1, INTCSIHTIJC is generated.
Transmit-only buffer		Indicates that the communication has stopped at the end of a job after a job abort*1 was triggered.
Dual buffer		
Direct access		

Note 1. Job abort condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1

19.5 Operation

19.5.1 Operating Modes (Master/Slave)

For a particular CSIH module, the master or slave mode determines the source of the serial clock.

19.5.1.1 Master Mode

In master mode, the serial transmission clock is generated by the internal baud rate generator (BRG) and provided to the slave(s) by signal CSIHTSCK.

Master mode is enabled by setting CSIHnCTL2.CSIHnPRS[2:0] to anything but 111_B.\

In master mode, the BRG frequency can be set by combining the CSIHnCTL2.CSIHnPRS[2:0] bit and the

- CSIH0BRSy.CSIH0BRS[11:0] bit for CSIH0
- CSIH1CTL2.CSIH1BRS[11:0] bit for CSIH1

(1) Chip select signals

In master mode, one or several chip select signals can be used. If several slaves are connected to the master, the chip select signals can be used to address one or several of the slaves. Only a selected slave is then enabled for communication.

The communication protocol as well as additional parameters are stored separately for each chip select signal. This makes it possible to adapt the data transfer individually to the requirements of each slave. For details, see Section 19.5.3, Chip Selection (CS) Features.

(2) Clock defaults

The default level of CSIHTSCK depends on the clock phase selection bit, and is high when CSIHnCTL1.CSIHnCKR = 0 and is low when CSIHnCTL1.CSIHnCKR = 1.

The example below shows the communication in master mode for 8-bit data, CSIHnCTL1.CSIHnCKR = 0, CSIHnCFGx.CSIHnDAPx = 0, and MSB first.

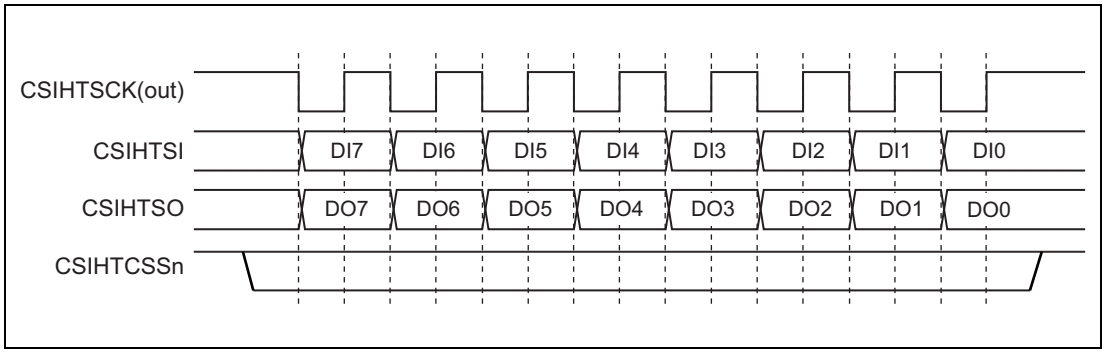


Figure 19.9 Transmit/Receive in Master Mode

19.5.1.2 Slave Mode

In slave mode, another device is the communication master and provides the transmission clock. Transmit/receive operation normally is started as soon as a clock signal is detected.

Slave mode is selected by setting the CSIHnCTL2.CSIHnPRS[2:0] bit to 111_B.

In slave mode, the transmission protocol setting of the CSIHnCFG0 register is enabled (setting of the CSIHnCFG1-CSIHnCFG7 register is disabled).

- CSIHnPSx[1:0]: parity usage
- CSIHnDLSx[3:0]: data length selection
- CSIHnDIRx: data direction
- CSIHnCKPx, CSIHnDAPx: clock phase and data phase

NOTE

When using slave mode, disable the baud rate generator (BRG) by setting the bits

- CSIH0BRSy.CSIH0BRS[11:0] to 000_H for CSIH0
- CSIH1CTL2.CSIH1BRS[11:0] to 000_H for CSIH1

However, if you are using a time-out error of CSIH0, set the CSIH0BRSy.CSIH0BRS[11:0] bits to a value other than 000_H. If you are using a time-out error of CSIH1, set the CSIH1CTL2.CSIH1BRS[11:0] bits to a value other than 000_H.

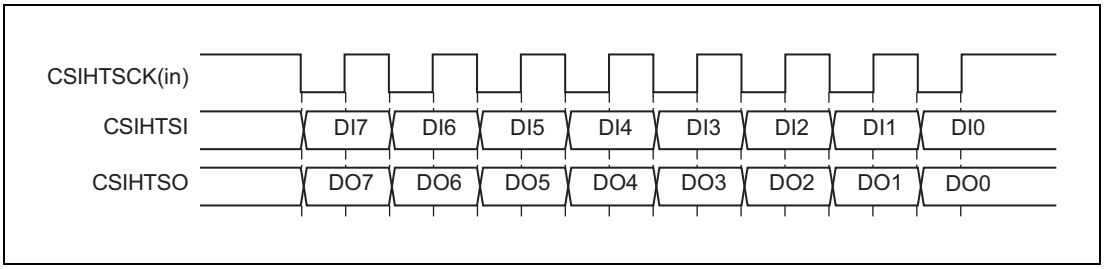


Figure 19.10 Transmit/Receive in Slave Mode

19.5.2 Master/Slave Connections

19.5.2.1 One Master and One Slave

The following figure illustrates the connections between one master and one slave.

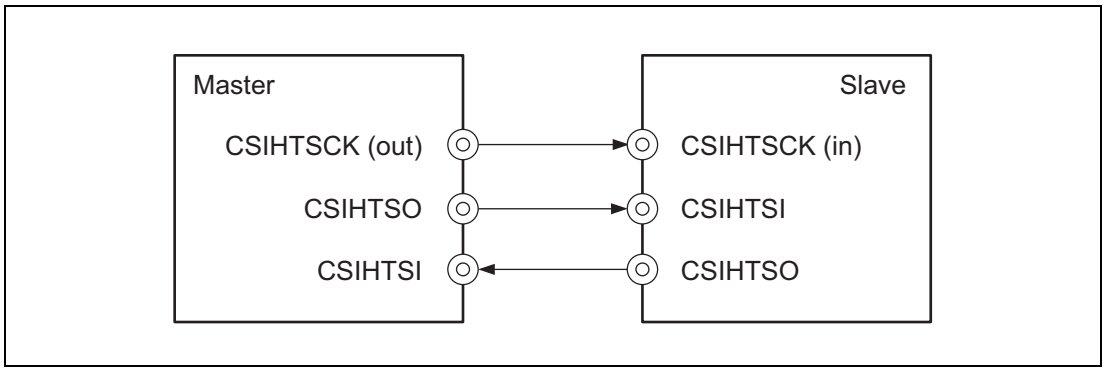


Figure 19.11 Direct Master/Slave Connection

19.5.2.2 One Master and Multiple Slaves

The following figure illustrates the connections between one master and multiple slaves. In this example, the master provides one chip select (CS) signal to each of the slaves. This signal is connected to the slave select input $\overline{\text{CSIHTSSI}}$ of the slave.

The $\overline{\text{CSIHTSSI}}$ signal can be enabled/disabled by using the CSIHnCTL1.CSIHnSSE bit.

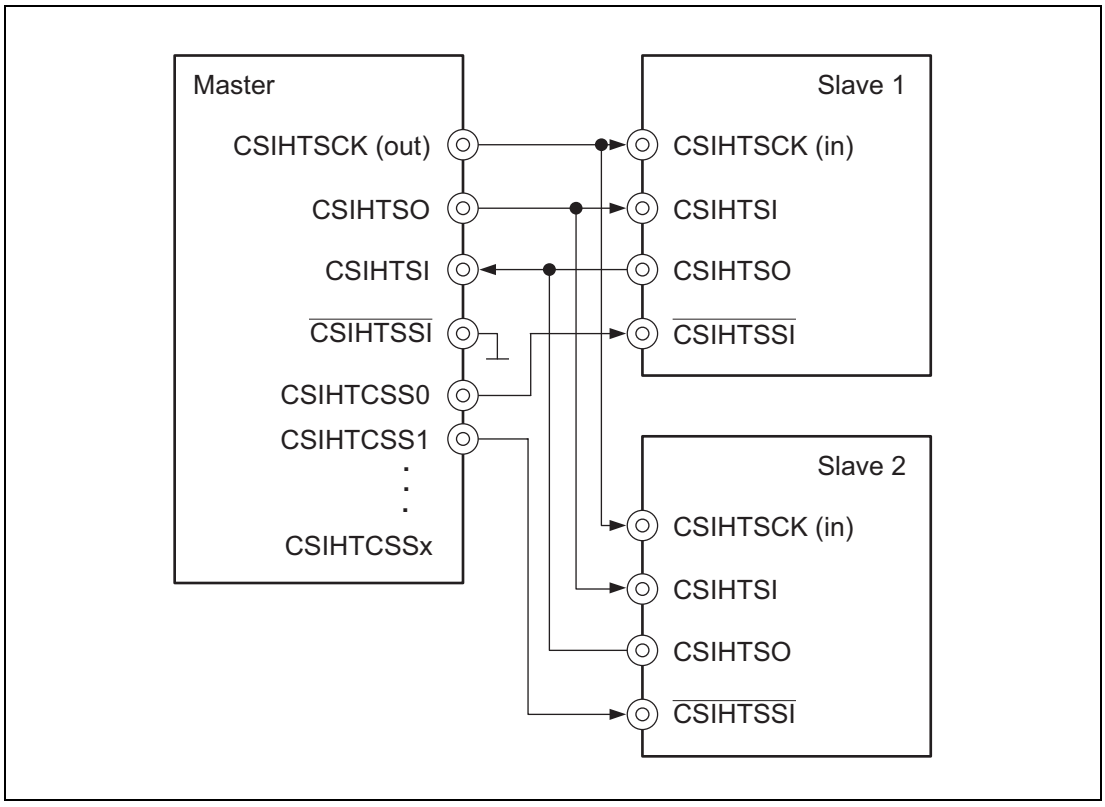


Figure 19.12 Connections between One Master and Multiple Slaves

By default, the chip select level is active low. That means, a slave is selected (enabled) as a CSIH slave when its $\overline{\text{CSIHTSSI}}$ signal has low level. However, to adapt the CS to other devices, the output level of each chip select signal can also be programmed to be active high.

If a slave is not selected, it will neither receive nor transmit data. In addition, its output CSIHTSO of a slave that is not selected is set to input mode in order to avoid interference with the output of another slave that was selected, in transmit-only mode or transmit/receive mode (CSIHnCTL0.CSIHnTXE = 1).

19.5.3 Chip Selection (CS) Features

The chip select signal, CSIHTCSSx can be used by the master to select one or several slaves for communication.

19.5.3.1 Configuration Registers

The parameters for each chip select signal CSIHTCSSx are defined in the corresponding configuration register CSIHnCFGx. The parameters include the communication protocol and additional CS parameters.

The communication protocol specifies:

- Data length: The number of bits to be sent or received. (CSIHnCFGx.CSIHnDLSx[3:0])
- Transfer direction: MSB or LSB first. (CSIHnCFGx.CSIHnDIRx)
- Parity usage: Odd, even, 0 parity or none. (CSIHnCFGx.CSIHnPSx[1:0])
- Clock phase and data phase. (CSIHnCFGx.CSIHnCKPx, CSIHnCFGx.CSIHnDAPx)

Additional parameters for each chip select signal that is only available in master mode are:

- Prescaler selection of the baud rate generator separately for each chip select signal (CSIH0CFGx.CSIH0BRSSx[1:0] for CSIH0, CSIH1CFGx.CSIH1PSCLx[1:0] for CSIH1)
- Chip select priority: Separates between “dominant” and “recessive” chip select signals. The priority applies if two or more chip selects signals with different configurations are simultaneously activated for message broadcasting. In this case, the configuration that is set as dominant is used. (CSIHnCFGx.CSIHnRCBx)

The principle is also called “Recessive Configuration for Broadcasting” (RCB).

CAUTION

It is forbidden to specify several chip select signals as dominant with different configurations unless all dominant chip selects have the same configuration.

- Chip select timing:

– Setup time T_{setup}: The time from setting the CS signal active to starting data output. (CSIHnCFGx.CSIHnSPx[3:0])
– Inter-data time T_{inter}: The time between one data and the next following data while the same CS signal is active. (CSIHnCFGx.CSIHnINx[3:0])
– Hold time T_{hold}: The time during which the CS signal remains active until CS is switched. (CSIHnCFGx.CSIHnHDx[3:0])
– Idle time T_{idle}: Inactive time after terminating a CS signal or after every data transfer to the same CSx. (CSIHnCFGx.CSIHnIDx[2:0])

The CS timings of the setup time, the inter-data time, the hold time, and the idle time are illustrated in the figure below. When CSIHnCFGx.CSIHnIDLx bit is set to 1, IDLE time is inserted for every transfer regardless of CS signal.

Figure 19.13 provides an example when the default CSIHTCSS1 and CSIHTCSS2 signals are active low (CSIHnCTL1.CSIHnCSL1 bit = 0, CSIHnCTL1.CSIHnCSL2 bit = 0). The active level can be specified individually for each CS.

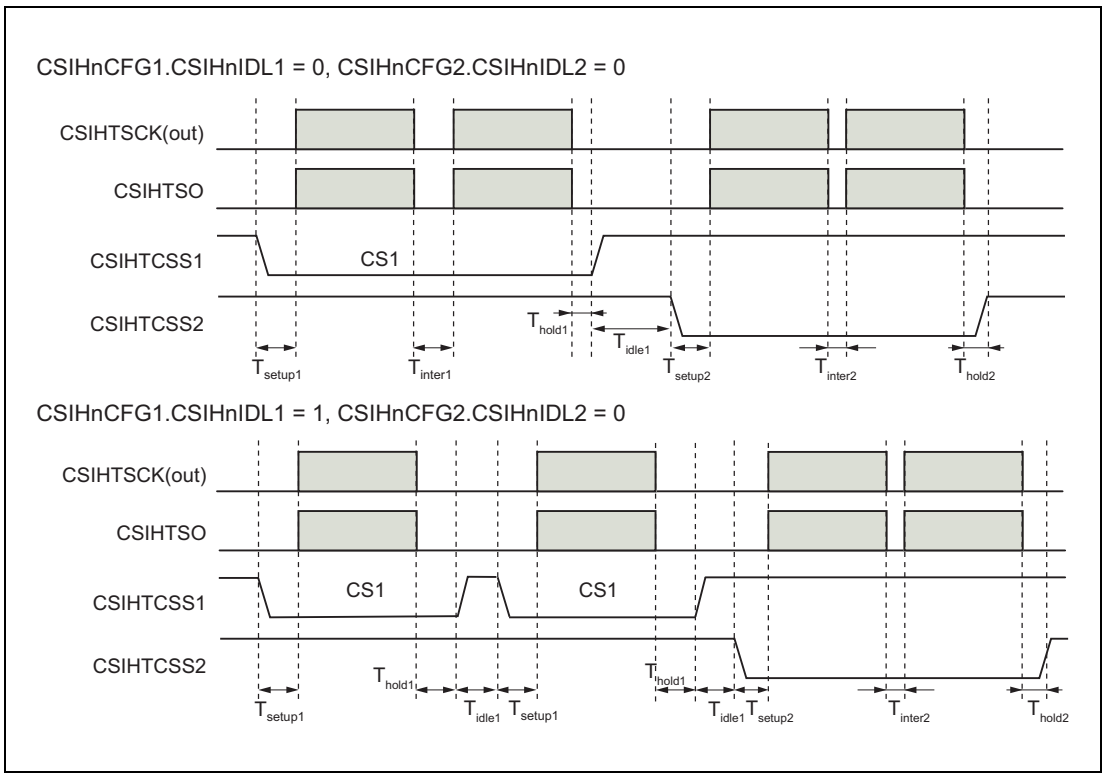


Figure 19.13 Chip Select Timings

Note that each CS signal can have a different value for setup, inter-data time, hold time, and idle time.

A particular chip select signal is activated by setting the appropriate bit in the transmission register CSIHnTX0W.CSIHnCSx.

CSIHnRX0W.CSIHnCSx in the reception register indicates the chip select signal associated with the received data.

CAUTION

For CSIH0 only:
When high priority communication function by CPU control is enabled

(CSIH0CTL1.CSIH0PHE = 1), IDLE state is inserted regardless of IDLn bit settings when priority communication mode is changed from low to high and from high to low.

19.5.3.2 CS Example

The following figure shows an example of two consecutive data transmissions.

The first communication uses CS0 to communicate with one single slave. The second enables CS0 and CS1 to broadcast a message to two slaves. The priority of CS0 is set to “recessive: low priority”, the priority of CS1 to “dominant: high priority”. Consequently, the second is conducted by using CS1 settings which are set in dominant.

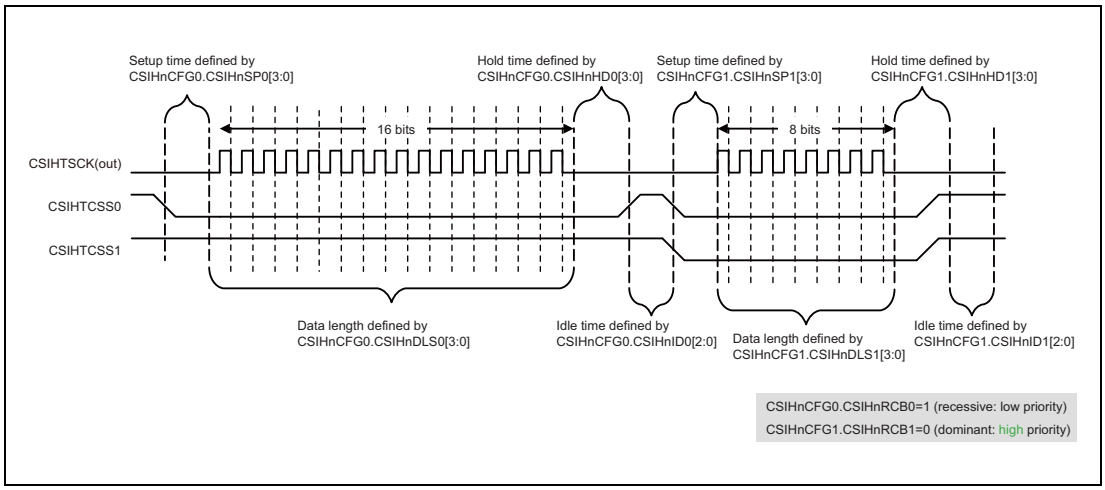


Figure 19.14 Chip Select and RCB Example

19.5.3.3 Job Concept

In terms of CSIH, a job consists of a number of data targeted for transfer.

Job mode enable

The job mode can only be enabled in master mode. The job mode is enabled and disabled by CSIHnCTL1.CSIHnJE, while the CSIH is disabled by CSIHnCTL0.CSIHnPWR = 0.

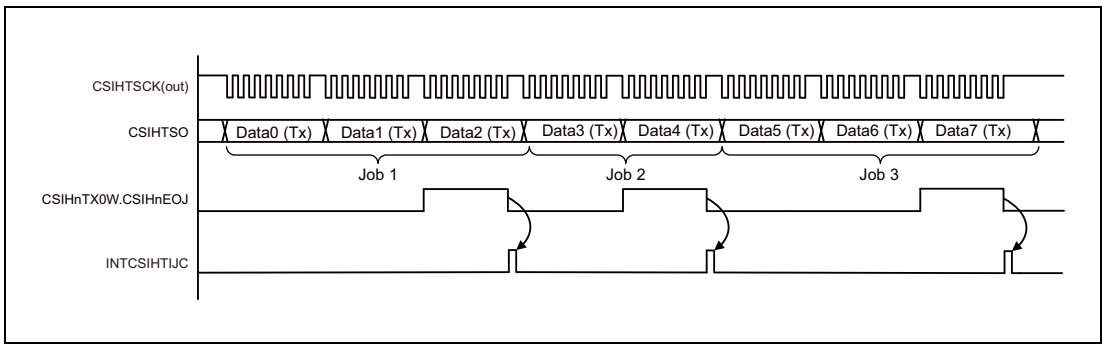


Figure 19.15 Job Examples

A job ends by transmitting data with CSIHnTX0W.CSIHnEOJ = 1.

A communication stop can be specified to occur after a job has finished. This is done by setting CSIHnCTL0.CSIHnJOBE. When CSIHnJOBE is set, the communication continues until data is sent, for which the CSIHnEOJ bit was set. After this data is sent, the communication is stopped and the interrupt INTCSIHTIJC is generated.

19.5.4 Details of Chip Select Timing

19.5.4.1 Changing the Clock Phase

The serial clock level specified by CSIHnCFGx.CSIHnCKPx can be changed while communication is stopped. The minimum value of an idle time is one period of transmission clock (CSIHTSCK(out)).

If the idle time is set to 0.5 transmission clock periods (in CSIHnCFGx.CSIHnIDx[2:0]) and two consecutive data is sent with different CSIHnCFGx.CSIHnCKPx configuration, the idle time is automatically extended to one period of CSIHTSCK(out).

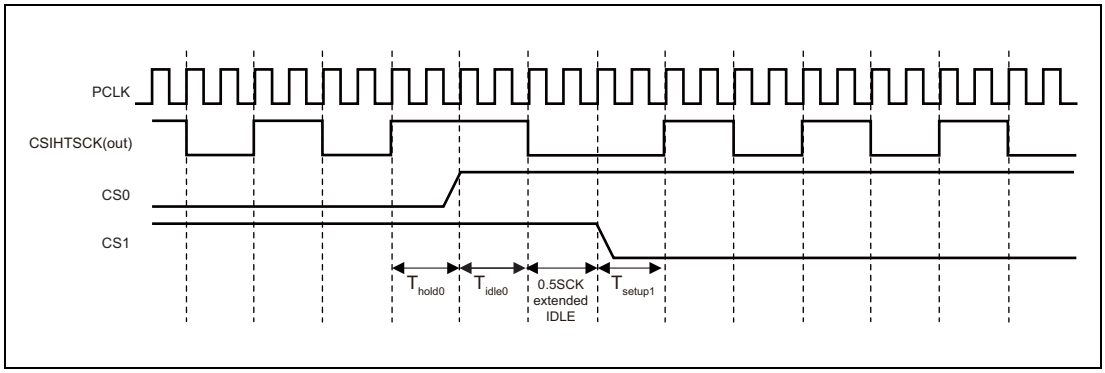


Figure 19.16 Clock Phase Timing with PCLK/4, $T_{hold}0 = T_{setup1}$ = 0.5CSIHTSCK, T_{idle0} = 0.5CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG1.CSIHnCKP1 = 1 (CSIHTCSS1)

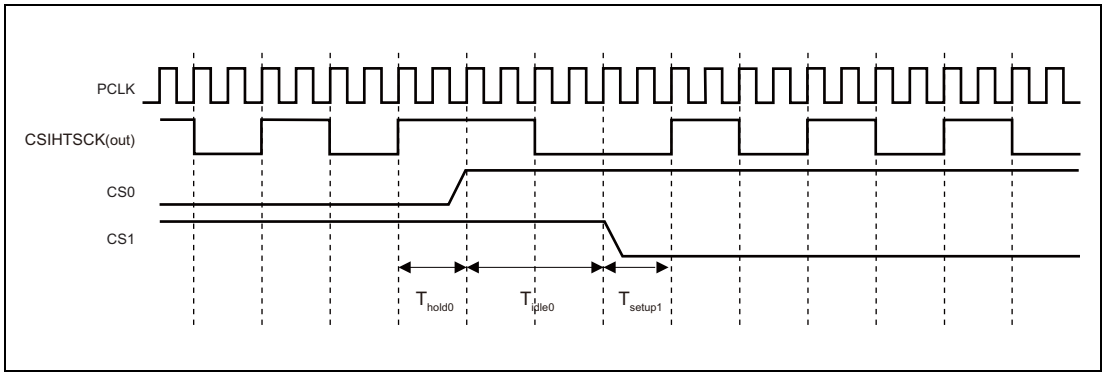


Figure 19.17 Clock Phase Timing with PCLK/4, $T_{hold}0 = T_{setup1}$ = 0.5CSIHTSCK, T_{idle0} = 1CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG1.CSIHnCKP1 = 1 (CSIHTCSS1)

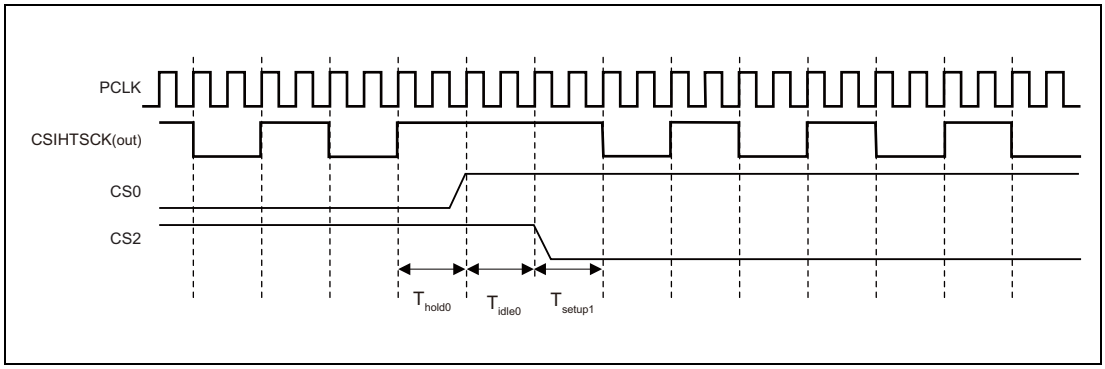


Figure 19.18 Clock Phase Timing with PCLK/4, $T_{hold}0 = T_{setup1}$ = 0.5CSIHTSCK, T_{idle0} = 0.5CSIHTSCK, CSIHnCFG0.CSIHnCKP0 = 0 (CSIHTCSS0) → CSIHnCFG2.CSIHnCKP2 = 0 (CSIHTCSS2)

19.5.4.2 Changing the Data Phase

The CSIHnCFGx.CSIHnDAPx bit defines the phase of the data bits relative to the clock.

The relation between the setting of the CSIHnCFGx.CSIHnDAPx bit and the hold and setup periods is described below.

Hold time is the period from the last edge of the serial clock (CSIHTSCK) until the signals on CSIHTCSS[7:0] change to the inactive level.

Setup time is the period from the signals on CSIHTCSS[7:0] changing to the active level until output (on CSIHTSO) of the data to be transmitted.

Accordingly, there is a gap of 0.5 cycles of CSIHTSCK until output of an edge of the serial clock signal (CSIHTSCK).

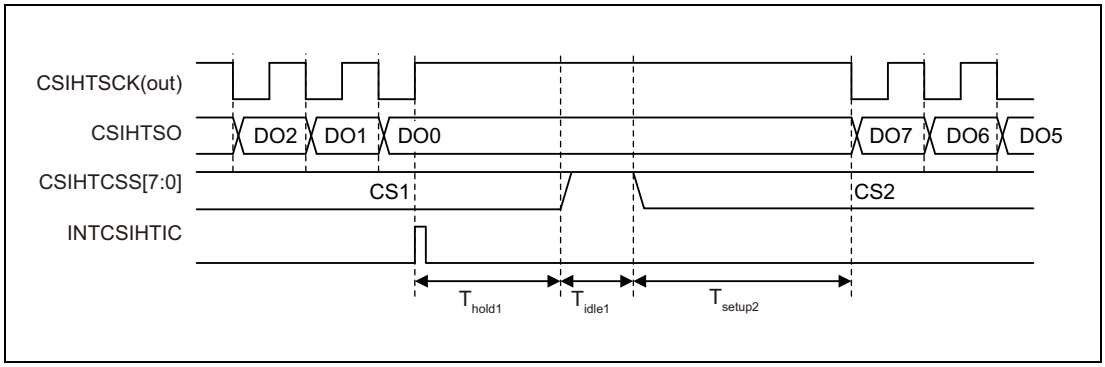


Figure 19.19 Data Phase Timing with CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 0

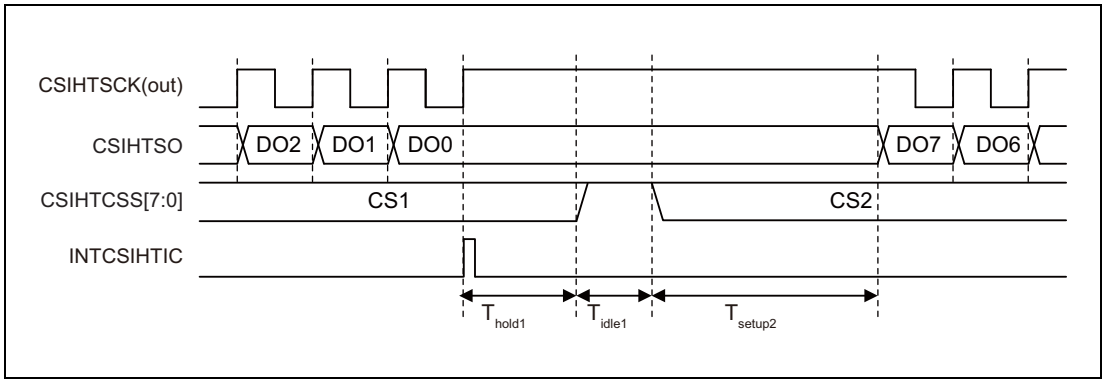


Figure 19.20 Data Phase Timing with CSIHnCFG1.CSIHnCKP1 = 1, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 1

19.5.5 Transmission Clock Selection

19.5.5.1 CSIH0 Transmission Clock Selection

In master mode, the transfer clock frequency is selectable using the following bits:

- CSIH0CTL2.CSIH0PRS[2:0]
- CSIH0BRSy.CSIH0BRS[11:0](y = 0 to 3)
- CSIH0CFGx.CSIH0BRSSx[1:0] (x = 0)

The transfer clock frequency of transmission clock CSIHTSCK is determined by the setting of the CSIH0CTL2.CSIH0PRS[2:0] bits and the setting of the CSIH0BRSy.CSIH0BRS[11:0] bits, but any one of CSIH0BRS3 to CSIH0BRS0 can be selected for each chip select signal with the CSIH0CFGx.CSIH0BRSSx[1:0] bits.

The following figure shows a block diagram of the baud rate generator.

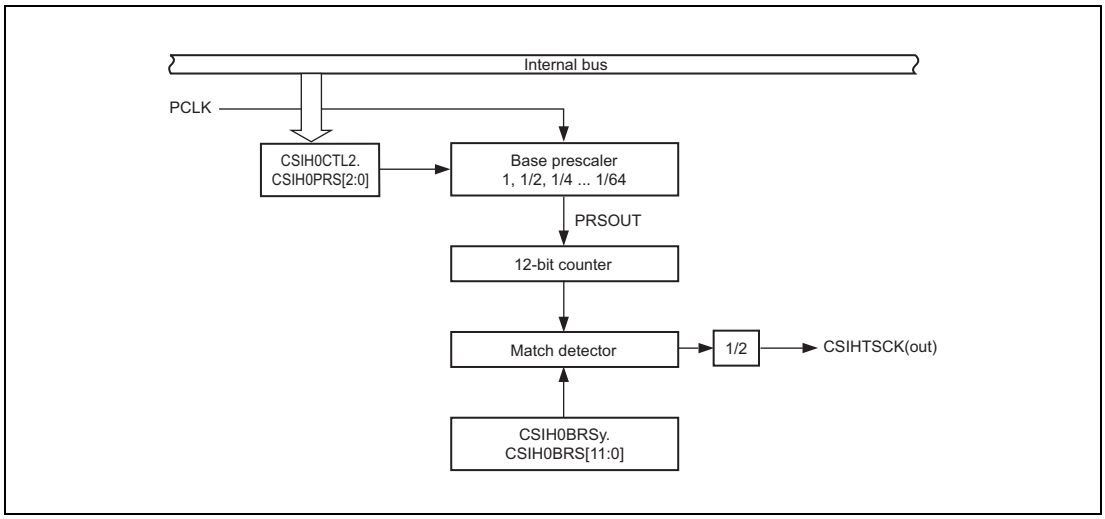


Figure 19.21 CSIH0 Baud Rate Generator Block Diagram

By setting CSIH0BRSy.CSIH0BRS[11:0] to 000_H, CSIH0BRSy.CSIH0BRS[11:0] disables the baud rate generator, and thus CSIHTSCK of the corresponding channel is stopped.

Transfer clock frequency calculation

The transfer clock frequency in master mode is calculated as:

Transfer clock frequency (CSIHTSCK) = PCLK / (division ratio of PCLK) = PCLK / (2α × k × 2),

where

α = CSIH0CTL2.CSIH0PRS[2:0] = 0 to 6

k = CSIH0BRS0.CSIH0BRS0[11:0] = 1 to 4095
(when CSIH0CFGx.CSIH0BRSSx[1:0] = 0)
CSIH0BRS1.CSIH0BRS1[11:0] = 1 to 4095
(when CSIH0CFGx.CSIH0BRSSx[1:0] = 1)
CSIH0BRS2.CSIH0BRS2[11:0] = 1 to 4095
(when CSIH0CFGx.CSIH0BRSSx[1:0] = 2)
CSIH0BRS3.CSIH0BRS3[11:0] = 1 to 4095
(when CSIH0CFGx.CSIH0BRSSx[1:0] = 3)

Transfer clock frequency upper and lower limits

When setting the transfer clock frequency, please note the followings.

- The minimum transfer clock frequency in master mode and slave mode is PCLK/524160.
- The maximum transfer clock frequency is as follows:
  - In master mode: up to PCLK/4
  - In slave mode: up to PCLK/6

19.5.5.2 CSIH1 Transmission Clock Selection

In master mode, the transfer clock frequency is selectable using the following bits:

- CSIH1CTL2.CSIH1PRS[2:0]
- CSIH1CTL2.CSIH1BRS[11:0]
- CSIH1CFGx.CSIH1PSCLx[1:0] (x = 0)

The clock frequency of base transmission clock CSIHBCLK is determined by the setting of the CSIH1CTL2.CSIH1PRS[2:0] bits and the setting of the CSIH1CTL2.CSIH1BRS[11:0] bits.

Additional prescalers CSIH1CFGx.CSIH1PSCLx[1:0], each dedicated to a chip select x, selects a divisor 1/1, 1/2, 1/4 or 1/8, that determines the transmission clock CSIHTSCKx for the chips select x.

The following figure shows a block diagram of the CSIH1 baud rate generator.

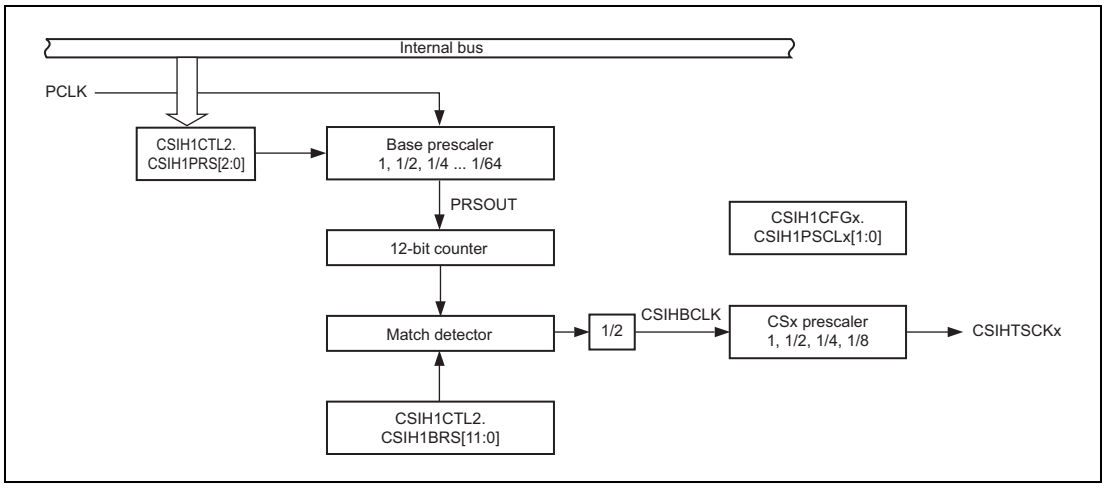


Figure 19.22 CSIH1 Baud Rate Generator Block Diagram

Setting CSIH1CTL2.CSIH1BRS[11:0] to 000_H disables the baud rate generator, and thus CSIHTSCK of the corresponding channel is stopped.

Transfer clock frequency calculation

The transmission clock frequency CSIHTSCKx in master mode is calculated as:

$$CSIHTSCKx = PCLK / (2^{\alpha} \times k \times 2 \times 2^{j})$$

where

α = CSIH1CTL2.CSIH1PRS[2:0] = 0 to 6

k = CSIH1CTL2.CSIH1BRS[11:0] = 1 to 4095

j = CSIH1CFGx.CSIH1PSCLx[1:0] = 0 to 3

Transfer clock frequency upper and lower limits

When setting the transfer clock frequency, please note the followings.

- The minimum transfer clock frequency in master mode and slave mode is PCLK/524160.
- The maximum transfer clock frequency is as follows:

– In master mode: up to PCLK/4
– In slave mode: up to PCLK/6

19.5.6 CSIH Buffer Memory

The CSIH has a configurable RAM that can be used for buffered I/O. One word is comprised of 32 bits data plus 7 bits ECC.

The following configurations are available:

Mode	CSIHnCTL0. CSIHnMBS	CSIHnMCTL0. CSIHnMMS[1:0]
FIFO mode	0	00_B
Dual buffer mode		01_B
Transmit-only buffer mode		10_B
Direct access mode	1	X

19.5.6.1 FIFO Mode

In FIFO mode, data can be written to the CSIHnTX0W register without waiting for completion of the transmission, and data can be received without reading the CSIHnRX0W register immediately, provided the FIFO is not full.

Data to be transmitted is stored to the FIFO memory. Transmission and reception occur simultaneously – one data is sent, one data is received. That means, received data overwrites the transmitted data in the FIFO.

The CSIH automatically updates the respective FIFO memory pointers when data is written to or read from the FIFO memory, or data is transmitted to or received from the FIFO memory:

Table 19.39 FIFO Mode

Pointer Description	Control Bit*1	Range
Number of unsent words	CCSIHnSTR0.CSIHnSPF[7:0]	0 to 128
Number of words received and stored in the FIFO	CSIHnSTR0.CSIHnSRP[7:0]	0 to 128
Address for write/read of transmit data	CSIHnMRWP0.CSIHnTRWA[6:0]	0000_H to $01FC_H$
Address for read of received data	CSIHnMRWP0.CSIHnRRA[6:0]	0000_H to $01FC_H$
Address to be sent	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $01FC_H$

Note 1. The values are automatically updated after each read/write or data transmit/receive operation.

The CSIH status register contains also two FIFO status flags:

- CSIHnSTR0.CSIHnFLF: FIFO full
- CSIHnSTR0.CSIHnEMF: FIFO empty

When this mode is started, bit CSIHnSTCR0.CSIHnPCT must be set. By doing this, only CSIHnSTR0.CSIHnEMF is set, but not reset.

All FIFO pointers and FIFO flags excluding CSIHnSTR0.CSIHnEMF are reset and CSIHnSTR0.CSIHnEMF is set.

19.5.6.2 Dual Buffer Mode

In this mode, the memory is divided into two parts of equal size – this means 64 words for transmit data and 64 words for received data. In dual buffer mode, the respective buffer pointers indicate the values shown in the following table.

Table 19.40 Dual Buffer Mode

Pointer Description	Pointer*1	Range
Address of data written to and read from transmit buffer	CSIHnMRWP0.CSIHnTRWA[6:0]	0000_H to $00FC_H$
Address of data read from receive buffer	CSIHnMRWP0.CSIHnRRA[6:0]	0000_H to $00FC_H$
The number of transmit data remained in the transmit buffer	CSIHnMCTL2.CSIHnND[7:0]	0 to 64
Address to which data is sent to	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $00FC_H$

Note 1. Both pointers are automatically incremented after each read/write.

19.5.6.3 Transmit-Only Buffer Mode

In this mode, the entire memory is used to save transmission data.

Received data must be read directly from CSIHnRX0W/H.

In transmit-only buffer mode, the respective buffer pointer indicates the values shown in the following table.

Table 19.41 Transmit-Only Buffer Mode

Pointer Description	Pointer*1	Range
Address of data written to and read from transmit buffer	CSIHnMRWP0.CSIHnTRWA[6:0]	0000_H to $01FC_H$
The number of transmit data remained in the transmit buffer	CSIHnMCTL2.CSIHnND[7:0]	0 to 128
Address to which data is sent to	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $01FC_H$

Note 1. Pointers are automatically incremented after each read/write.

19.5.6.4 Direct Access Mode

In direct access mode, the CSIH memory is completely bypassed:

- Transmission data provided by the CPU to the transmission register CSIHnTX0W or CSIHnTX0H is directly copied to the shift register.
- Reception data is directly copied from the shift register to the reception register CSIHnRX0W or CSIHnRX0H.

19.5.7 Data Transfer Modes

19.5.7.1 Transmit-Only Mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 0 puts the CSIH in transmit-only mode. Start of transmission depends on the memory mode:

- In case of FIFO or direct access mode, transmission starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of dual buffer or transmit-only buffer mode, transmission starts when the CSIHnMCTL2.CSIHnBTST bit is set.

19.5.7.2 Receive-Only Mode

Setting CSIHnCTL0.CSIHnTXE = 0 and CSIHnCTL0.CSIHnRXE = 1 puts the CSIH in receive-only mode.

In master mode, start of reception depends on the memory mode:

- In case of FIFO or direct access mode, reception starts when dummy data is written in the CSIHnTX0W or CSIHnTX0H register.

In slave mode, reception starts as soon as the CSIHTSCK transmission clock from the master is received. It is not necessary to write data to the CSIHnTX0W or CSIHnTX0H register of the slave.

- In case of dual buffer mode or transmit-only buffer mode, reception starts when the CSIHnMCTL2.CSIHnBTST bit is set.

19.5.7.3 Transmit/Receive Mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 1 puts the CSIH in transmit/receive mode.

Start of communication (transmission and reception) depends on the memory mode:

- In case of FIFO or direct access mode, communication starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of dual buffer or transmit-only buffer mode, communication starts when the CSIHnMCTL2.CSIHnBTST bit is set..

19.5.7.4 Summary

The following table summarizes this section. It shows how data transfer is started in the various memory, operating, and transfer modes.

Table 19.42 Start of Data Transfer

Memory and Operating Mode		Transfer Mode	
		Transmit-Only Transmit/Receive	Receive-Only
FIFO, direct access	Master	Writing to the CSIHnTX0W register or the CSIHnTX0H register	Writing to the CSIHnTX0W register on the CSIHnTX0H register
	Slave	Incoming clock from master	Incoming clock from the master
Transmit-only buffer, dual buffer	Master	CSIHnMCTL2.CSIHnBTST = 1	CSIHnMCTL2.CSIHnBTST = 1
	Slave	Incoming clock from master	Incoming clock from master

19.5.8 Data Length Selection

19.5.8.1 Data Length from 2 to 16 Bits

The length of a data packet is selectable for each chip select signal from

- 2 to 16 bits for CSIH0
- 7 to 16 bits for CSIH1

using CSIHnCFGx.CSIHnDLSx[3:0]. The examples below show the communication with MSB first (CSIHnCFGx.CSIHnDIRx = 0).

Data length = 16 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 0000_B):

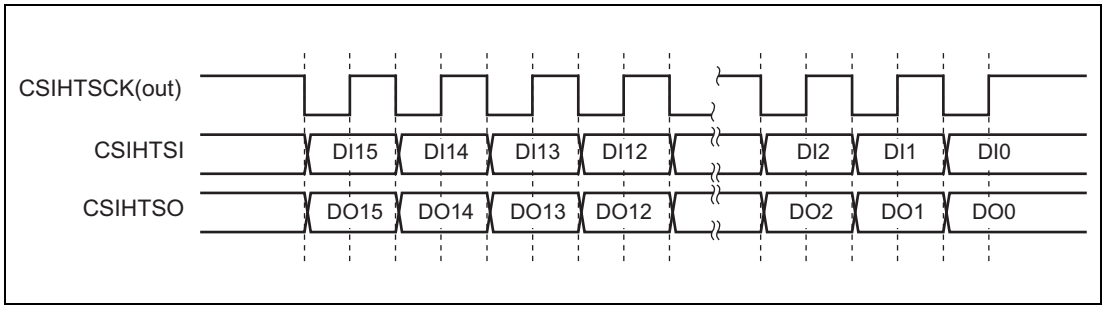


Figure 19.23 16 Bit Data Length, MSB First

Data length = 14 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1110_B):

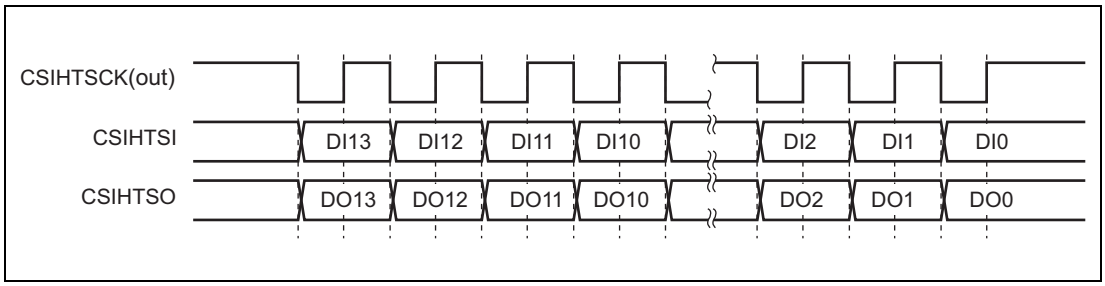


Figure 19.24 14 Bit Data Length, MSB First

19.5.8.2 Data Length Greater than 16 Bits

If the data to be sent/received exceeds 16 bits, the extended data length (EDL) feature can be used.

EDL function is enabled by setting the CSIHnCTL1.CSIHnEDLE bit to 1.

EDL function works as follows:

- The data has to be broken into 16-bit blocks plus remainder. For example, data of 42 bits would be broken into two 16-bit blocks plus 10 bits.
- The bit length of the remainder is set as “data length” in CSIHnCFGx.CSIHnDLSx[3:0].
- For transmitting the 16-bit blocks, CSIHnTX0W.CSIHnEDL must be set to 1. In this case, the data written to CSIHnTX0W is sent as a 16-bit data length regardless of the CSIHnCFGx.CSIHnDLSx[3:0] bit setting.
- The transfer is complete after a block with the specified data length (the remainder of data specified with CSIHnTX0W.CSIHnEDL = 0) has been sent.

Example

Example for sending 40-bit data ($123456789A_H$) to CS0:

40 bits are split into two blocks of 16 bits plus 8 bits.

- Initialize CSIHnCFG0.CSIHnDLS0[3:0] = 8.
- To send $123456789A_H$ with MSB first, write the following sequence to CSIHnTX0W:
  - $20FE\ 1234_H$ (CSIHnTX0W.CSIHnEDL = 1)
  - $20FE\ 5678_H$ (CSIHnTX0W.CSIHnEDL = 1)
  - $00FE\ 009A_H$ (CSIHnTX0W.CSIHnEDL = 0)

The following figure illustrates the timing.

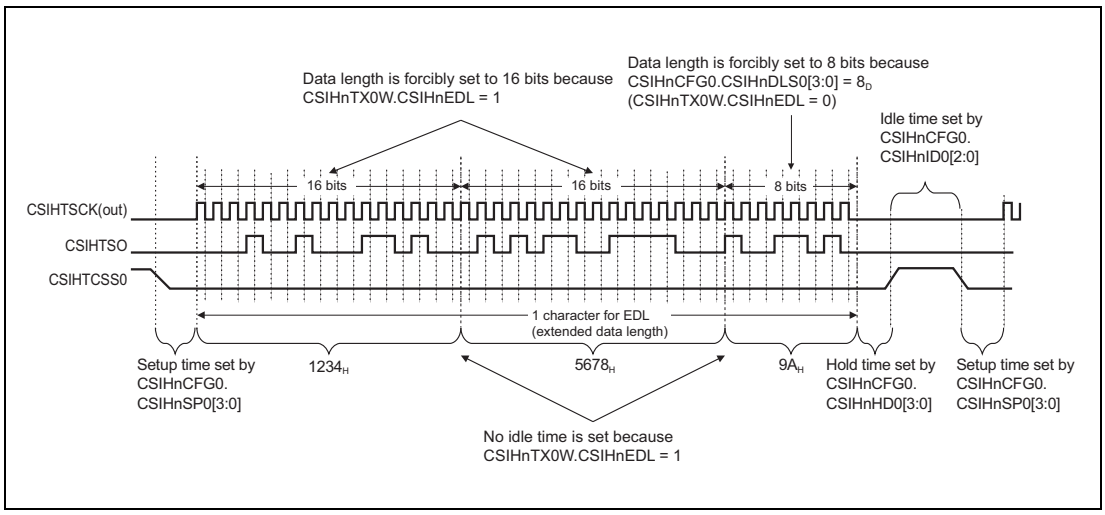


Figure 19.25 EDL Timing Diagram

NOTES

1. CSIH0:
   Data lengths settings of 1 bit is only permitted in combination with EDL mode.
   It is not possible to send two consecutive data with a data length of 1 bit.
2. CSIH1:
   Data lengths settings lower than 7 bits are only permitted in combination with EDL mode.
   It is not possible to send two consecutive data with a data length of less than 7 bits.
3. If parity is enabled, the parity bit is added after the last bit.
4. When data is sent using extended data length (EDL) function, use the same chip select signal.
5. To consider the data direction, pay attention to the following example:
   - Data to be sent:123456_H
   - MSB first:
     Set CSIHnCFGx.CSIHnDIRx = 0
     Write CSIHnTX0W = $20FE\ 1234_H$ (EDL bit = 1)
     Write CSIHnTX0W = $00FE\ 0056_H$ (EDL bit = 0)
   - LSB first:
     Set CSIHnCFGx.CSIHnDIRx = 1
     Write CSIHnTX0W = $20FE\ 3456_H$ (EDL bit = 1)

Write CSIHnTX0W = 00FE 0012$_H$ (EDL bit = 0)

6. Operation is not guaranteed if CSIHnTX0W.CSIHnEOJ and CSIHnTX0W.CSIHnEDL are simultaneously set to “1” while CSIHnCTL1.CSIHnJE = 1 and CSIHnCTL1.CSIHnEDLE = 1.
7. EDL mode cannot be used in receive-only mode of slave mode. (CSIHnCTL2.CSIHnPRS[2:0] = 111$_B$, CSIHnCTL0.CSIHnTXE = 0, CSIHnCTL0.CSIHnRXE = 1)

19.5.9 Serial Data Direction Selection

The serial data direction is selectable for each chip select signal using the CSIHnDIRx bit in the CSIHnCFGx register.

The examples below show communication for a data length of 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000$_B$).

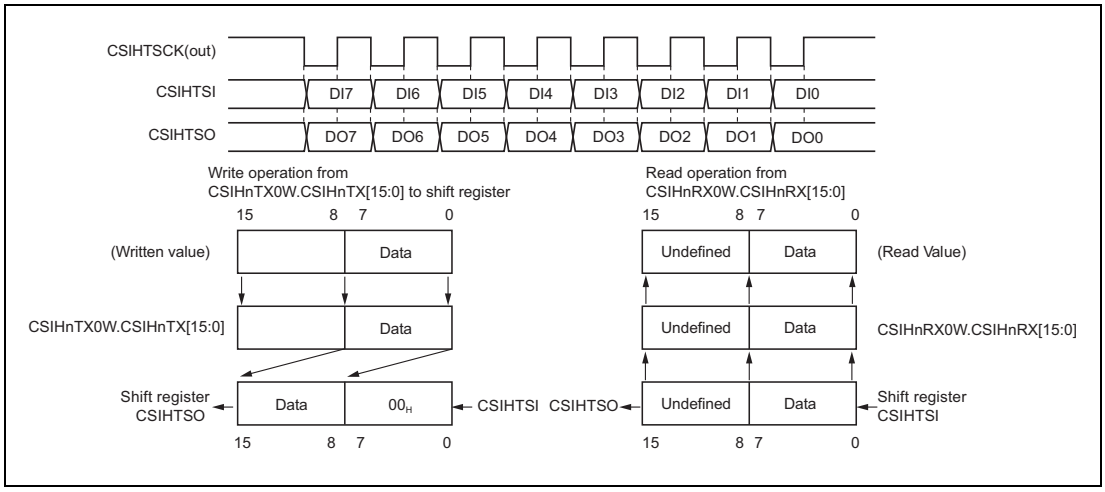


Figure 19.26 Serial Data Direction Select Function - MSB First (CSIHnDIRx = 0)

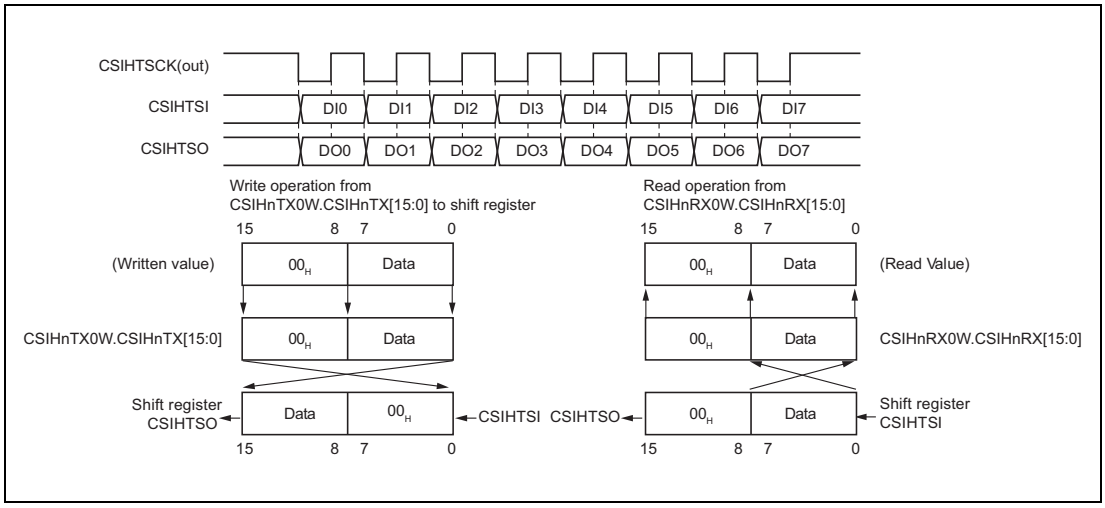


Figure 19.27 Serial Data Direction Select Function - LSB First (CSIHnDIRx = 1)

19.5.10 Slave Select (SS) Function

The Slave Select (SS) function realizes communication between one master and multiple slaves.

In master mode, the master device outputs the salve select signal (CSIHTCSSx) to select a single slave. Communication by a device in slave mode is enabled when the slave input select signal (CSIHTSSI) is at the low level.

See the Section 19.5.2, Master/Slave Connections, for an example of a connection using the SS function.

19.5.10.1 Communication Timing Using SS Function

The following figure illustrates the communication signal using the SS function and timings.

In slave mode, the data transfer configuration is determined by the CSIHnCFG0 register.

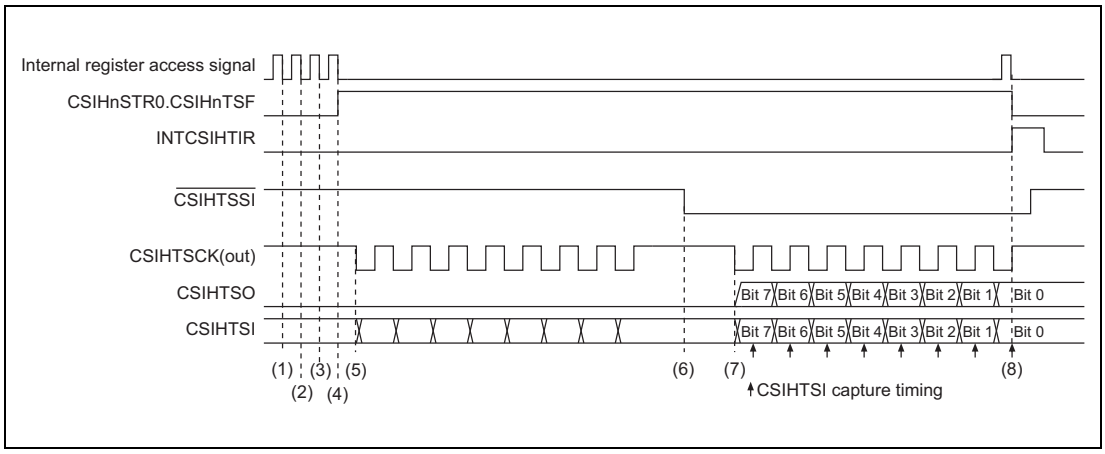


Figure 19.28 Tx/Rx Timing of Communication Using SS Function

(1) CSIH is put into slave mode by setting CSIHnCTL2.CSIHnPRS[2:0] = 111_B. CSIHnCFG0.CSIHnCKP0 and CSIHnCFG0.CSIHnDAP0 are 0.

(2) The data length is 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000_B). The data direction is MSB first (CSIHnCFG0.CSIHnDIR0 = 0).

(3) The transmit/receive mode is set (CSIHnCTL0.CSIHnTXE = 1, CSIHnCTL0.CSIHnRXE = 1, and CSIHnCTL0.CSIHnPWR = 1). Communication start is permitted.

(4) The transfer status flag CSIHnSTR0.CSIHnTSF is automatically set when transfer data is written to the CSIHnTX0W or CSIHnTX0H transmission register during direct access mode or FIFO mode.

(5) As long as signal $\overline{\text{CSIHTSSI}}$ is at the high level, transmission/reception is not started, even if an external transmission clock CSIHTSCK is applied. Input at CSIHTSI is ignored.

(6) As soon as $\overline{\text{CSIHTSSI}}$ falls to low level, indicating that CSIHTSO is enabled and ready for transmission.

(7) Now, as soon as the external clock signal CSIHTSCK is detected, the slave transmits data to CSIHTSO and simultaneously captures data from CSIHTSI.

(8) Interrupt INTCSIHTIR indicates when the reception is complete. The CSIHnRX0W/H register can be read.

19.5.10.2 CSIHTSSO Operation

CSIHnPWR	CSIHnTXE	CSIHnRXE	CSIHnSSE	CHISTSSO
0	—	—	—	H
1	—	—	0	H
	0		1	H
	1		1	Reversed value of $\overline{\text{CSIHTSSI}}$ level

The CSIHTSSO pin is a signal to control the I/O function of the chip's SO pin in case of using the SS function.

The CSIHTSO pin is enabled when the CSIHTSSO pin is "High" (the chip's SO pin is being driven).

The CSIHTSO pin is disabled when the CSIHTSSO pin is "Low" (the chip's SO pin is not being driven).

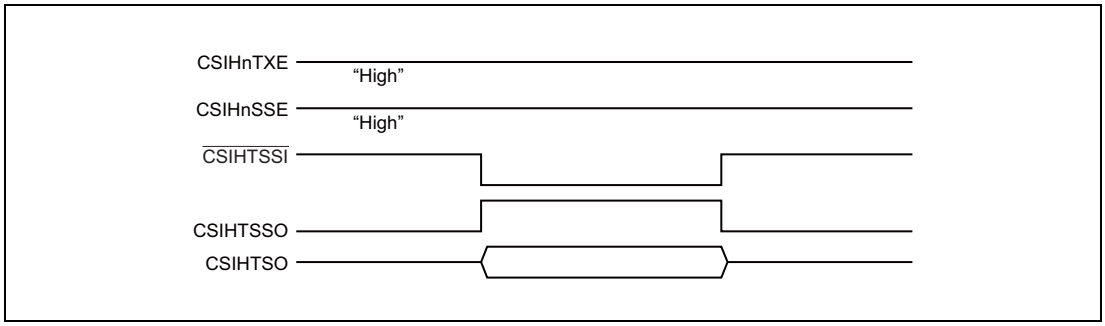


Figure 19.29 Operation of CSIHTSSO

CAUTION

If $\overline{\text{CSIHTSSI}}$ pin is changed during communication (CSIHnSTR0.CSIHnTSF = 1), current communication is not assured.

19.5.11 Handshake Function

CSIH features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by the CSIHnCTL1.CSIHnHSE bit. For handshake, the signals CSIHTRYI and CSIHTRYO are used.

The busy timing depends on the data phase selection bit CSIHnCFGx.CSIHnDAPx.

19.5.11.1 Slave Mode

When CSIHnCTL1.CSIHnHSE = 1 and the slave is busy, the CSIHTRYO signal outputs low level (0). This can happen in two cases:

1. When the next data to be sent is not ready:
   When the slave is in transmit-only mode or transmit/receive mode (CSIHnCTL0.CSIHnTXE = 1) and is in the states listed below, the CSIHTRYO output indicates the busy state (is at the low level).

Table 19.43 Memory Mode and Slave Transfer State

Memory Mode	Slave Transfer State
Direct access mode	When there is no more data to be sent
FIFO mode	When there is no more data to be sent (CSIHnSTR0.CSIHnEMF = 1)
Dual buffer mode	When CSIHnMCTL2.CSIHnBTST is not set to 1
Transmit-only buffer mode	

The example below is on the assumption of an eight-bit data length.

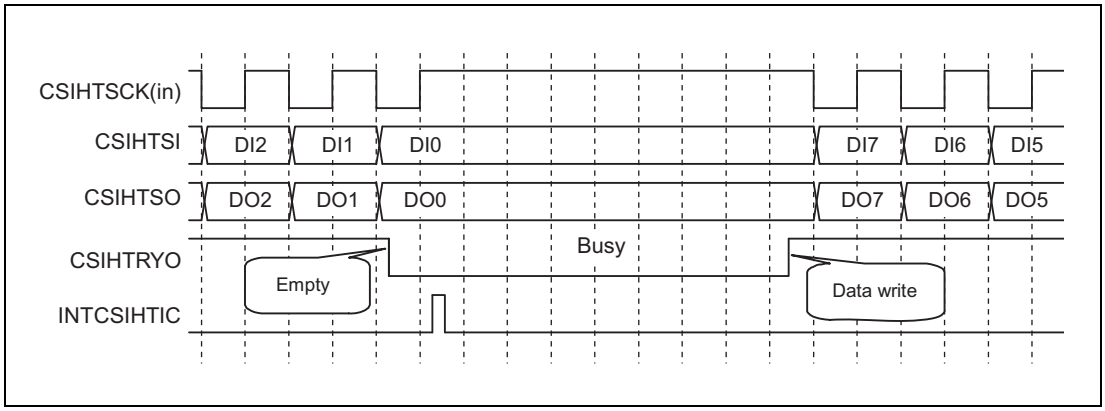


Figure 19.30 Busy Signal from the Slave (FIFO Mode; CSIHnCFGx.CSIHnDAPx = 0)

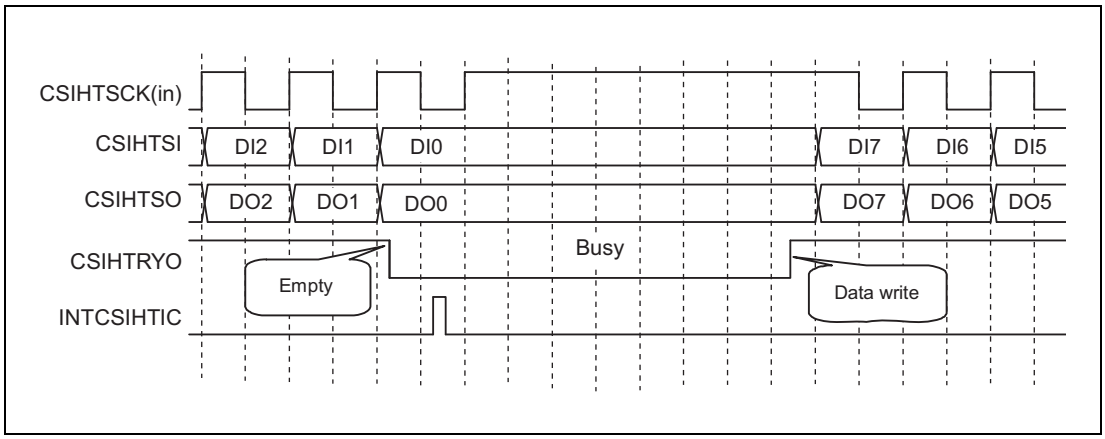


Figure 19.31 Busy Signal from the Slave (FIFO Mode; CSIHnCFGx.CSIHnDAPx = 1)

2. When receive register is full:
   When slave is set in receive-only mode or transmit/receive mode (CSIHnCTL0.CSIHnRXE = 1), and new data cannot be copied from a shift register to CSIHnRX0W/H (CSIHnRX0W/H is full) because the previously received data is still in the CSIHnRX0W/H register.
   When CSIHnCTL0.CSIHnRXE is 1 and is in the following states, CSIHTRYO outputs busy state (low level).

Table 19.44 Memory Mode and Slave Reception State

Memory Mode	Slave Reception State
Direct access mode	When CSIHnRX0W or CSIHnRX0H is full
FIFO mode	When receive data is remained in buffer (CSIHnSTR0.CSIHnFLF = 1)
Dual buffer mode	No applicable case
Transmit-only buffer mode	When CSIHnRX0W or CSIHnRX0H is full

The example below is on the assumption of an eight-bit data length.

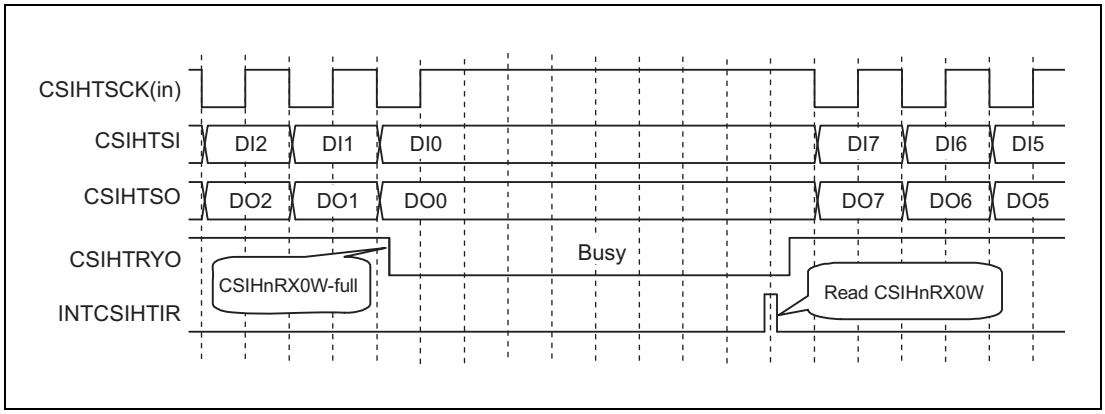


Figure 19.32 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 0)

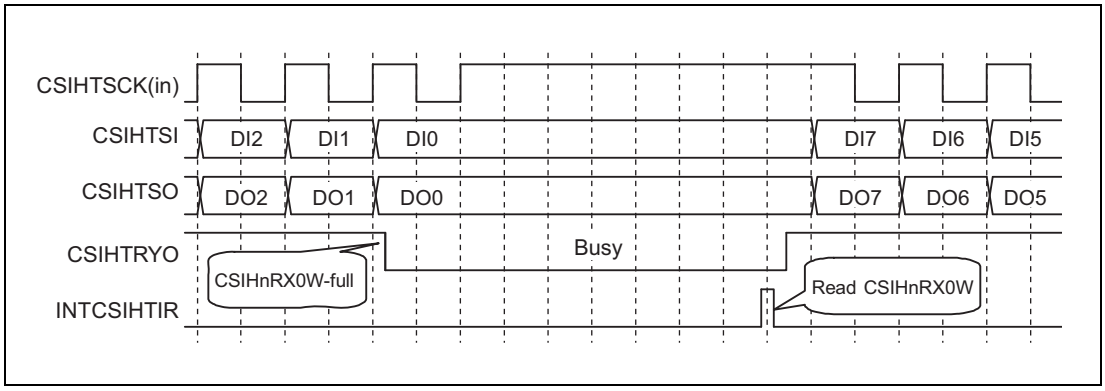


Figure 19.33 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 1)

19.5.11.2 Master Mode

When the master detects CSIHTRYI = 0 while CSIHnCTL1.CSIHnHSE = 1, the following transfer is put on hold, and the master goes into wait status. It suspends the CSIHTSCK clock.

The CSIHTRYI level is checked at each half clock cycle of CSIHTSCK.

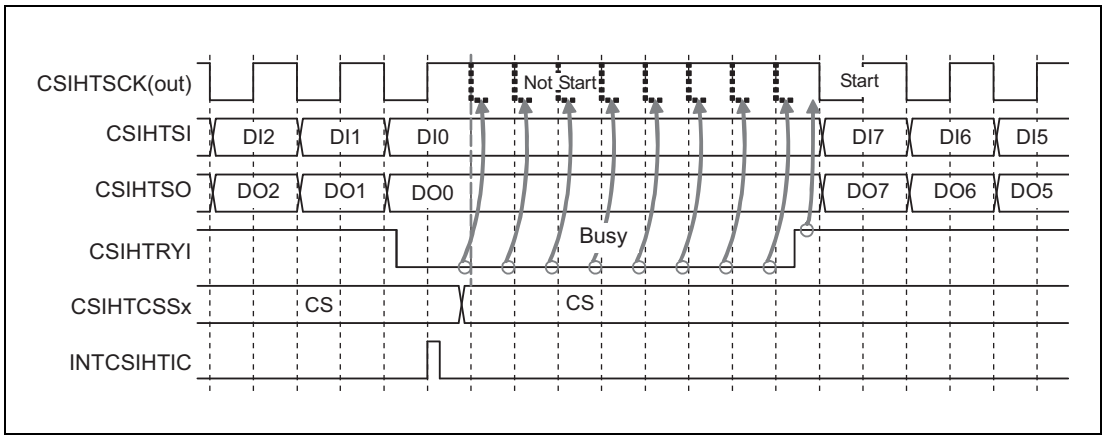


Figure 19.34 Master's Reaction on CSIHTRYI (CSIHnCFGx.CSIHnDAPx = 0)

The CSIHTRYI signal must be pulled down by the slave before the next transfer starts. If this is done while data transfer is in progress, the serial clock from the master is suspended after the transfer is complete.

The master resumes the communication as soon as CSIHTRYI becomes high (the slave is "ready").

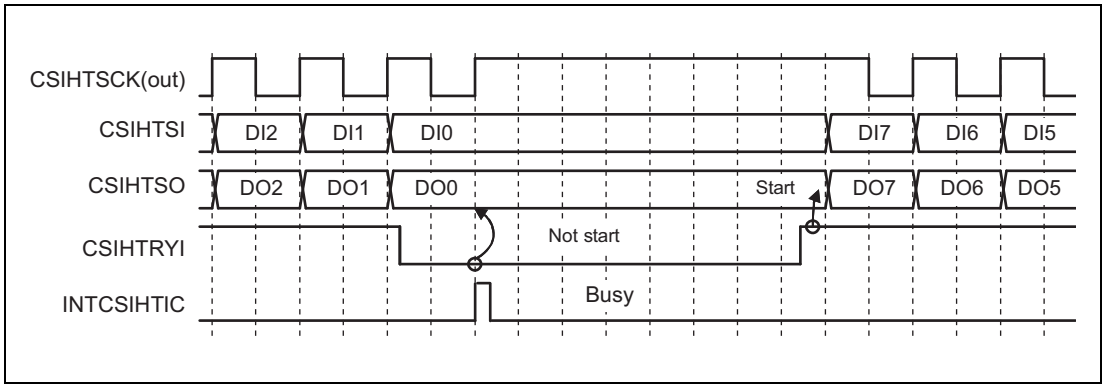


Figure 19.35 Master's Reaction on CSIHTRYI (CSIHnCFGx.CSIHnDAPx = 1)

CAUTIONS

1. **If multiple slaves are connected, the master must only detect the CSIHTRYI signal of the slave it has selected for communication.**
2. **Even when the CSIHTRYI pin of the master detects a CSIHTRYO signal from the slave during data transfer, the communication is not made to wait but continues until the data transfer is completed.**

19.5.12 Error Detection

CSIH can detect five error types:

- Data consistency error (transmission data)
- Parity error (received data)
- Overrun error (received data)
- Time-out error (in FIFO mode)
- Overflow error (in FIFO mode)

Check for parity, data consistency and time-out errors can be enabled/disabled individually.

If one of these errors is detected, the interrupt request, INTCSIHTIRE is generated and the corresponding flags are set.

19.5.12.1 Data Consistency Check

The purpose of the data consistency check is to ensure that the data physically sent as output signal is identical to the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by the CSIHnCTL1.CSIHnDCS bit (when checking data consistency, make sure that PIPCn.PIPCn_m = 1). It is not active if data transmission is disabled (CSIHnCTL0.CSIHnTXE = 0).

When the data consistency check is active, the data transferred from CSIHnTX0W or CSIHnTX0H to the shift register is copied to a separate register. In addition, the physical levels at CSIHTSO are read back via the CSIHTDCS signal into an own shift register.

After completion of the transmission, the sent data is compared with the original transmission data.

Mismatch is considered as a data consistency error:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnDCE bit is set.

Additionally, CSIHnRX0W.CSIHnTDCE of data that contains the error is set.

The data consistency check function is illustrated in the following block diagram.

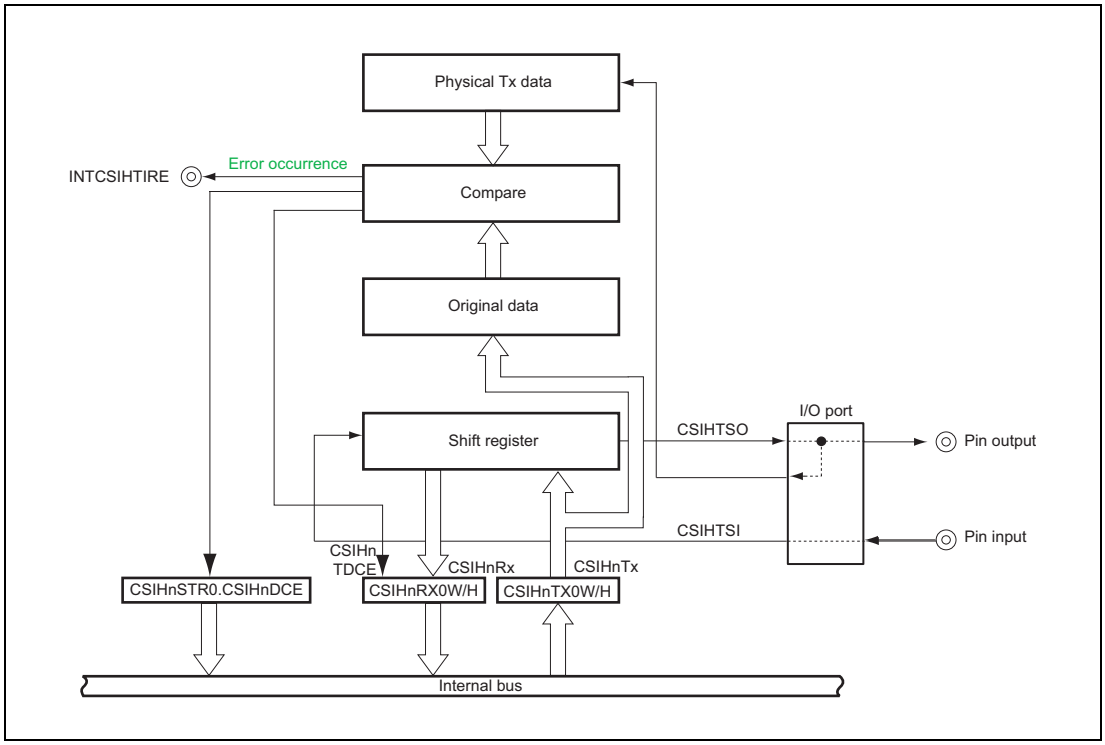


Figure 19.36 Data Consistency Check Functional Block Diagram

19.5.12.2 Parity Check

CSIH can append a parity bit to the last data bit (even if extended data length is used).

The use and type of parity is specified in CSIHnCFGx.CSIHnPSx[1:0].

Parity check is enabled if CSIHnCFGx.CSIHnPSx[1] = 1.

The parity bit is checked after a reception is complete. In case of parity error:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnPE bit is set.

Additionally, CSIHnRX0W.CSIHnRPE of data that contains the error is set.

The figure below shows an example.

- Data length is 8 bits.
- The data to be transmitted is 05_H and 35_H.
- Data direction is LSB first.
- Parity type is odd.

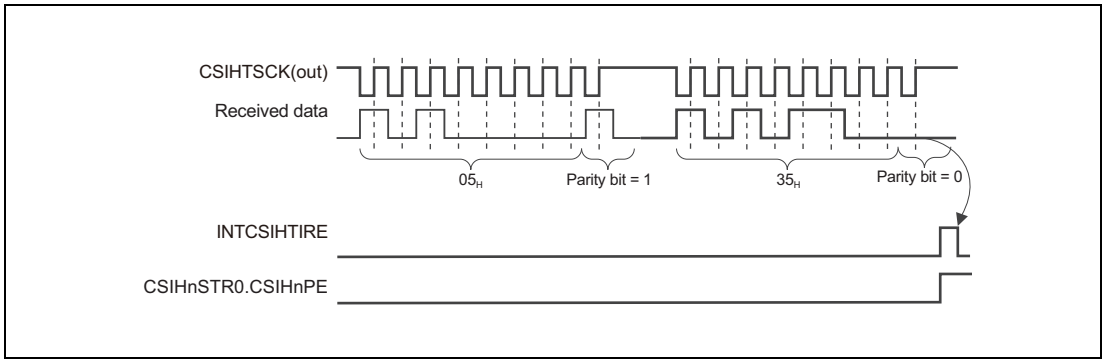


Figure 19.37 Parity Check Example

The parity bit of the first data is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

The parity bit of the second data is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

If the EDL (extended data length) function is used, the parity bit is added after the last bit of the data.

19.5.12.3 Time-Out Error

Time-out errors can be checked only in slave FIFO mode.

This error occurs if neither of the following occurred within a certain period of time:

- Received data in FIFO is read
- FIFO receives data from CSIHTSI

The time is defined in CSIHnMCTL0.CSIHnTO[4:0] in multiples of 8 times the transmission clock, CSIHTSCK. A time-out error occurs when the specified time is exceeded (The time-out time is not detected when CSIHnMCTL0.CSIHnTO[4:0] = 00000_B).

A dedicated time-out counter measures the time between the last and the next read operation.

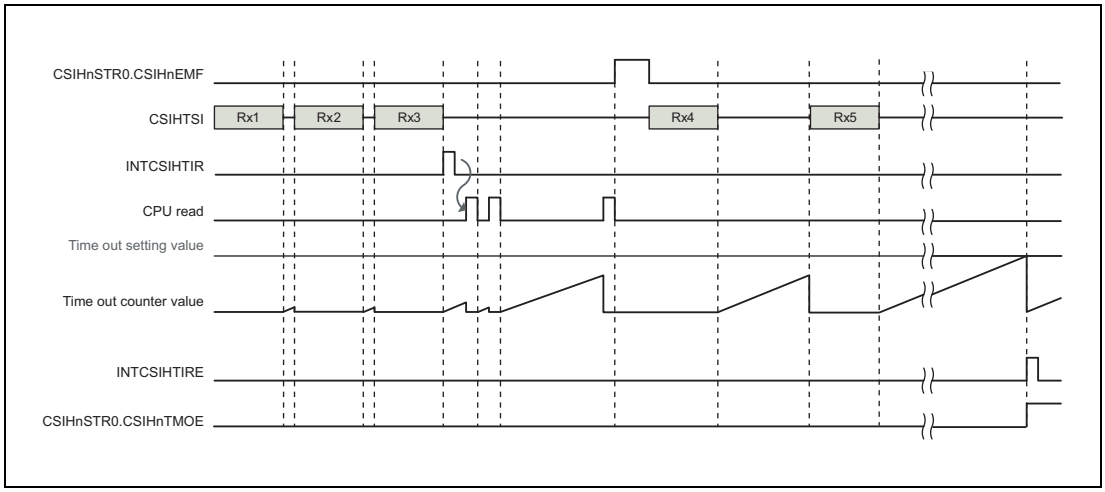


Figure 19.38 Time-Out Check Functional Timing Diagram

The start timing of the time-out counter is as follows:

- When reception is completed,

- When data read from the CPU completes,
  (The counter does not start if the buffer is empty.)
- When a time-out error is detected,

After a time-out error is detected, if data is still available in FIFO, the time-out counter restarts.

If the value set by bit CSIHnMCTL0.CSIHnTO[4:0] is reached again, the INTCSIHTIRE interrupt is output again.

The timeout counter continues to count until received data is read. To stop the counter, read all received data or set CSIHnSTCR0.CSIHnPCT to 1. Note that the pointer is cleared if you perform the latter.

The counter is reset at the following timing:

- Data is read once.
- A new data item is received.
- A timeout error is detected.
- The CSIHnSTR0.CSIHnPCT bit is set.

If a timeout error occurs, the following occurs:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnTMOE bit is set.

The dedicated time-out counter of CSIH0 is set by the CSIHnCTL2.CSIHnPRS[2:0] and CSIH0BRSy.CSIH0BRS[11:0] bits. If the value of the CSIH0BRSy.CSIH0BRS[11:0] bits is left as 000_H, the dedicated time-out counter of CSIH0 does not operate.

The dedicated time-out of CSIH1 counter is set by the CSIHnCTL2.CSIHnPRS[2:0] and CSIH1CTL2.CSIH1BRS[11:0] bits. If the value of the CSIH1CTL2.CSIH1BRS[11:0] bits is left as 000_H, the dedicated time-out counter of CSIH1 does not operate.

19.5.12.4 Overflow Error

An overflow error can happen in FIFO mode. It occurs when transmission data is written to the CSIHnTX0W register while the FIFO buffer is filled with received data.

Example

100 data have been transmitted. That means, the FIFO contains 100 received data. The application starts to read the received data.

While the read operation is in progress, the application begins to write another set of 50 transmission data to the FIFO. However, only 10 received data have been read up to now, 90 are still in the FIFO.

In this case, only 38 cells are available for new transmission data packets. When the CPU tries to write the 39th data, an overflow error happens.

This is illustrated in the following figure.

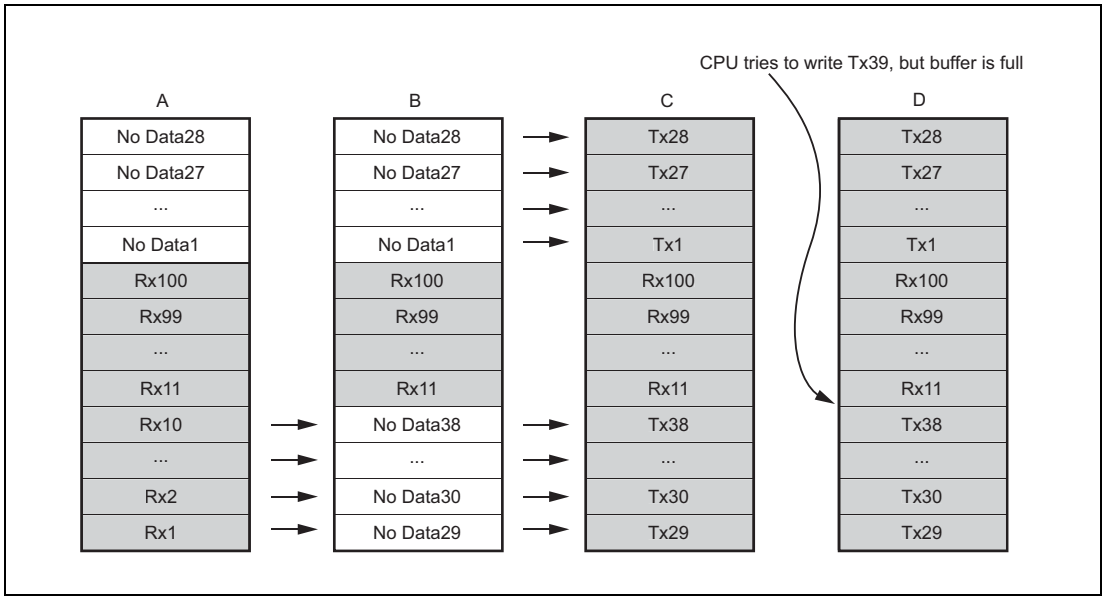


Figure 19.39 FIFO Overview

The 39th and subsequent data are discarded. The figure below shows the overflow timing.

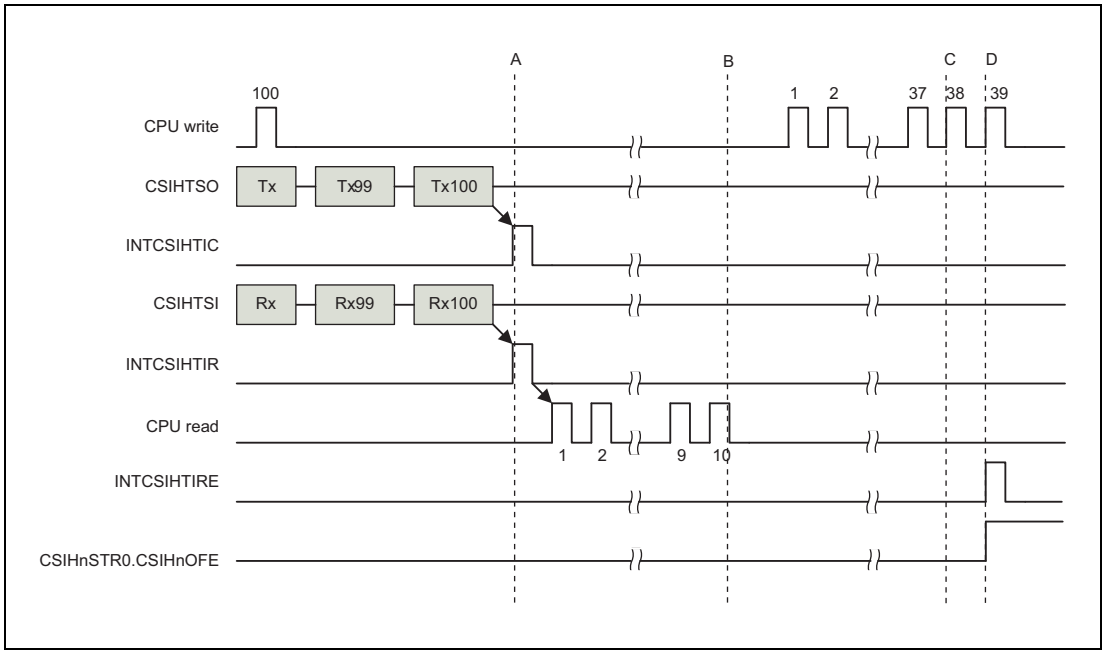


Figure 19.40 FIFO Overflow Timing

In case of overflow error:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnOFE bit is set.

19.5.12.5 Overrun Error

An overrun error can happen in direct access, transmit-only buffer, and FIFO modes. It cannot happen in dual buffer mode. The overrun error is not generated if data reception is disabled (CSIHnCTL0.CSIHnRXE = 0).

There are two conditions for overrun errors.

Condition for errors 1

- In FIFO mode, while the number of received data is 0 and CPU reads the CSIHnRX0W/H register

Condition for errors 2

- In slave mode, when CSIHnCTL1.CSIHnHSE = 0 (handshake function disabled):
  - In direct access mode or transmit-only buffer mode, when reception is completed while the previous received data is remained in the CSIHnRX0W/H register.
  - In FIFO mode, when FIFO buffer completes receiving data in the full state, an interrupt is generated, and communication continues.

(1) Direct access/transmit-only buffer

In direct access and transmit-only buffer mode, this error occurs when newly received data cannot be transferred from the shift register to the reception register CSIHnRX0W/H. This happens when CSIHnRX0W/H was not read and therefore contains previous reception data.

The following figure illustrates the function.

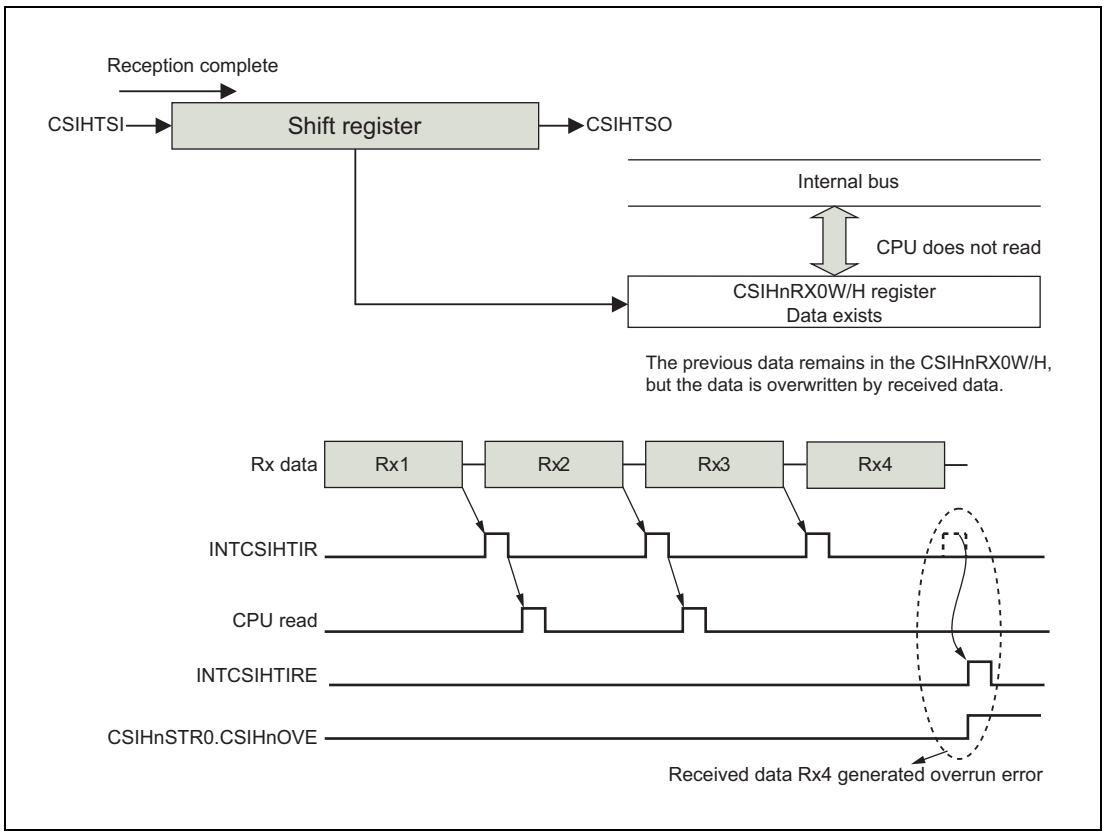


Figure 19.41 Overrun Error Detection in Direct Access and Transmit-Only Buffer Mode

NOTE

An overrun error can be avoided in slave mode by using handshake function. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver has read its reception register and is ready again.

(2) FIFO mode

In FIFO mode, this error occurs if:

1. Newly received data cannot be transferred from the shift register to the FIFO because the FIFO is full.

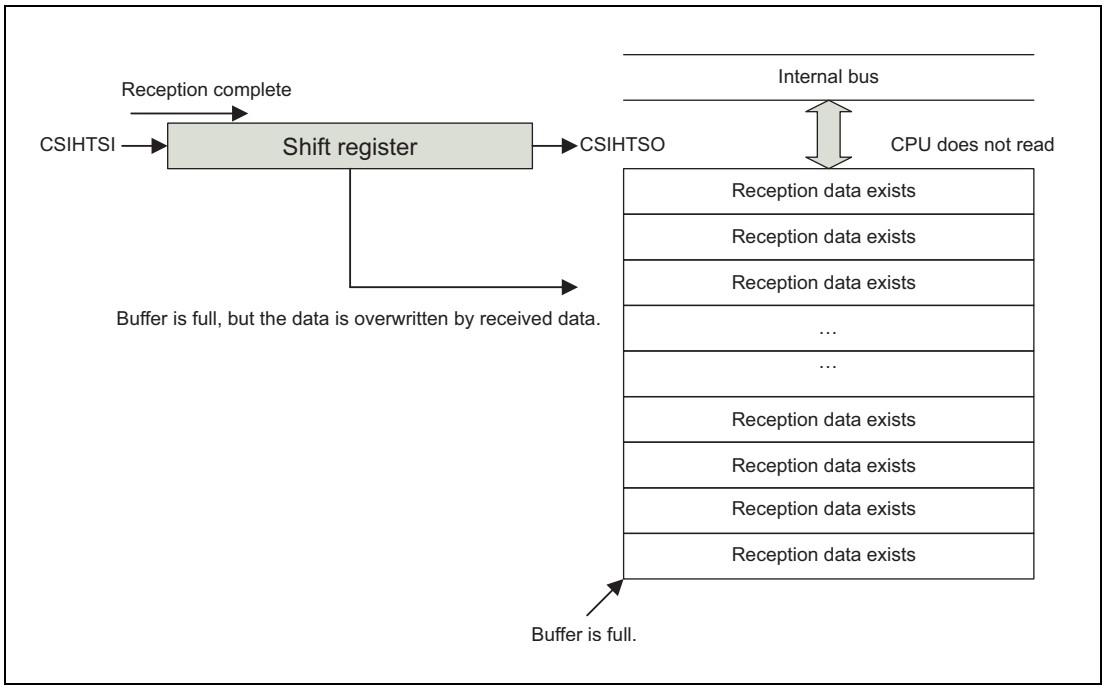


Figure 19.42 Overrun Error Detection in FIFO Mode (FIFO Full)

In FIFO mode, an overrun error is generated only when using received-only mode.

NOTE

An overrun error can be avoided in slave mode by using the handshake function. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver has read its reception register and is ready again.

2. The CPU attempts to read non existing reception data.

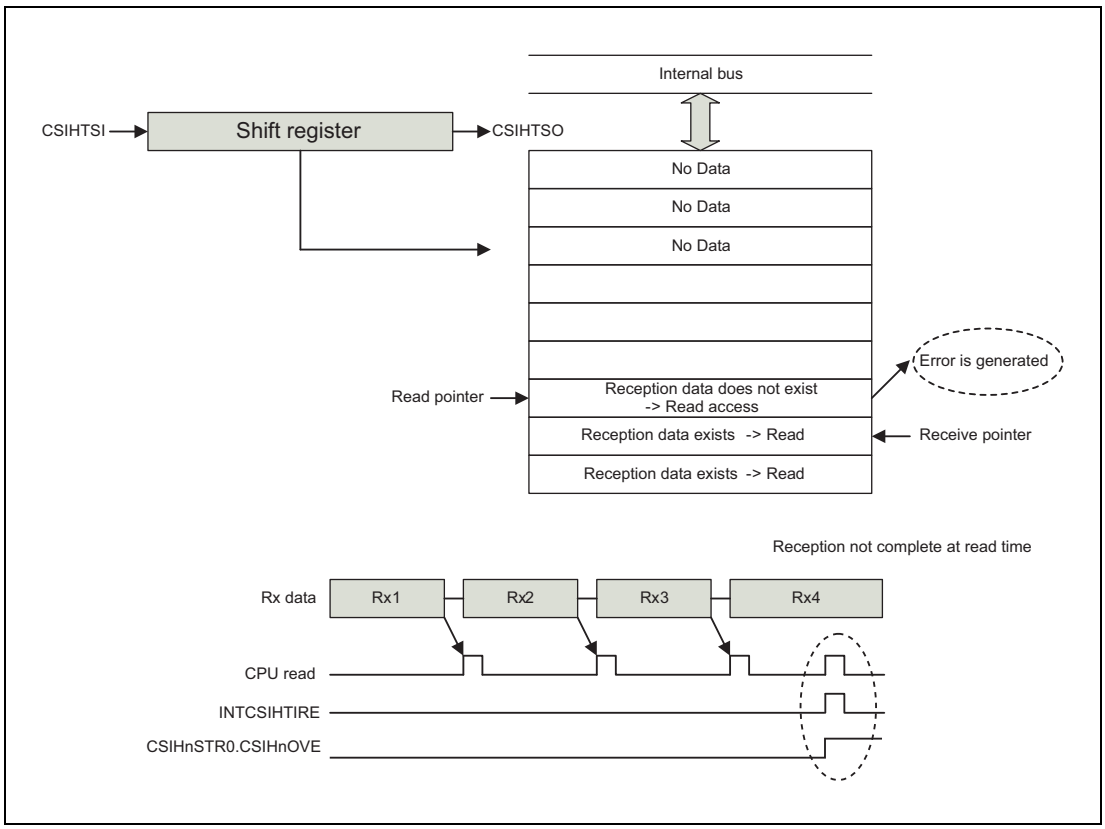


Figure 19.43 Overrun Error Detection in FIFO Mode (No Data)

In case of overrun error:

- Interrupt INTCSIHTIRE is generated.
- The CSIHnSTR0.CSIHnOVE bit is set.
- Received data is overwritten and the communication continues.
  (When the CPU tries to read non-existent data, the CPU starts reading again after a wait until reception is completed.)

For details see Section 19.5.11, Handshake Function.

19.5.13 Loop-Back Mode

Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active (CSIHnCTL1.CSIHnLBM = 1), CSIHTCSSx is fixed to the inactive level (the active level is defined by the CSIHnCTL1.CSIHnCSLx value). The transmit and receive signals are internally connected, as shown in the figures below.

The signals CSIHTSCK, CSIHTSO, and CSIHTSI are disconnected from the ports. In addition, the CSIHTSO output level is fixed to low, and CSIHTSCK is set to reset level (High) regardless of the value of the CSIHnCFGx.CSIHnCKPx. The rest of CSIH works as in normal operation.

In order to test CSIH, put it in loop-back mode and carry out normal transfer operations. Then check that the received data is the same as the transmitted data. Any connected device remains unaffected by the loop-back test.

Table 19.45 Pin Output Level in Loop-Back Mode

Pin Name	Output level
CSIHTSCK(out)	High level
CSIHTCSS[7:0]	Inactive level
CSIHTSO	Low level (not dependent on the previous value)
Interrupt	Normal function
CSIHTRYO	Normal function (Low level)

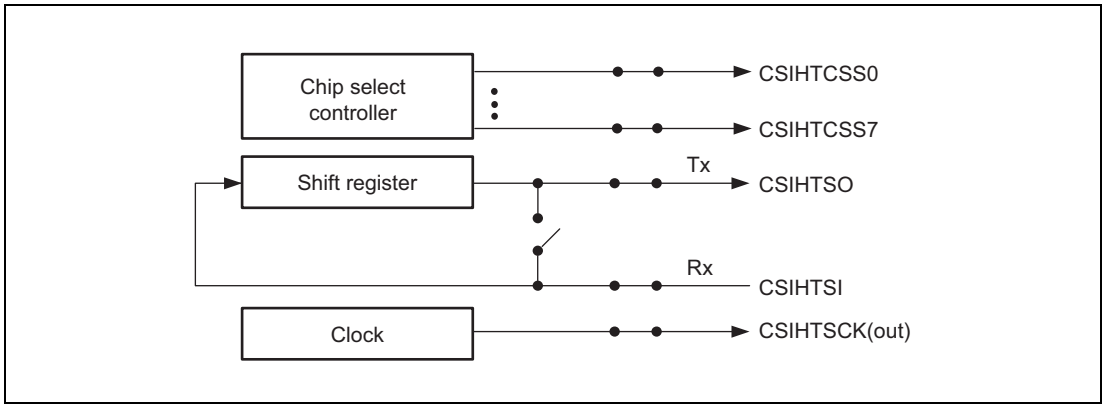


Figure 19.44 Normal Operation

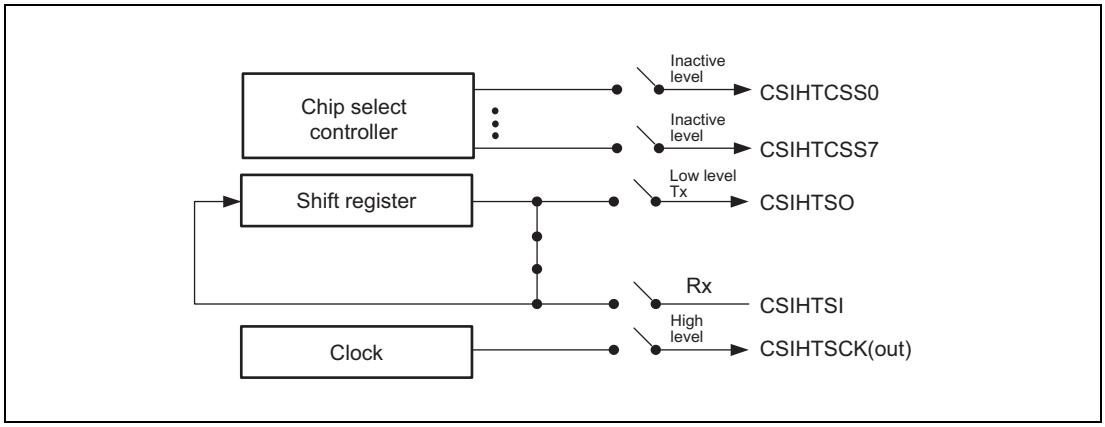
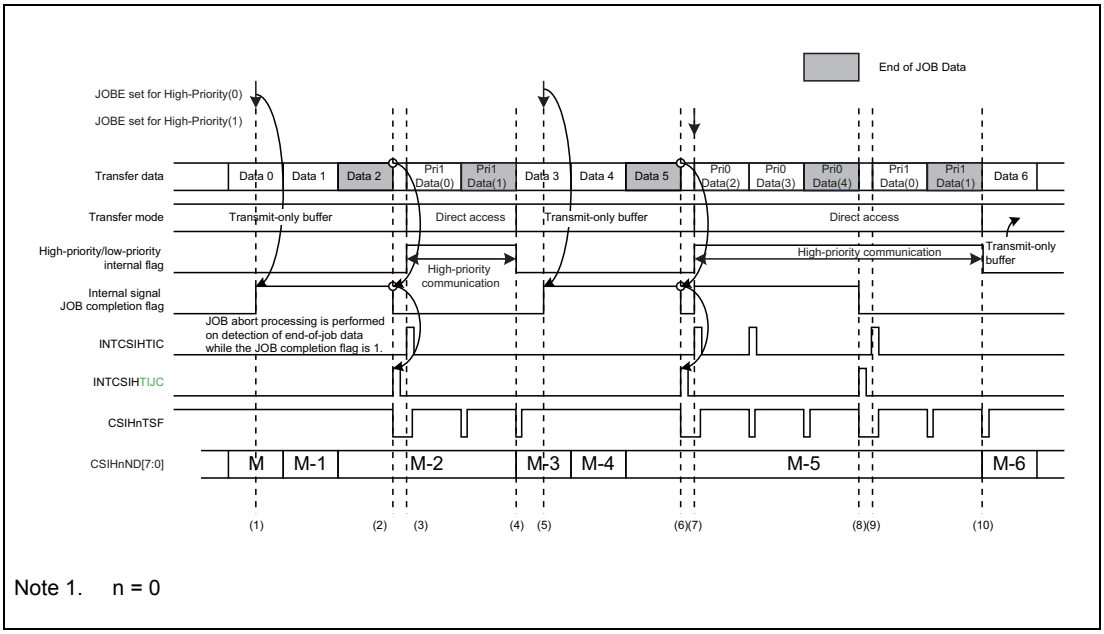


Figure 19.45 Loop-Back Mode Operation

19.5.14 CPU-Controlled High Priority Communication Function (CSIH0 only)

CSIH has a function to abort low priority communication to perform high priority communication if it receives a high-priority communication request from the CPU while low-priority communication is being used. This function supports transmit-only buffer mode as low priority communication and direct access mode as high-priority communication only. To enable this function, CSIH0CTL1.CSIH0PHE and CSIH0CTL1.CSIH0JE must be set to 1.

The following figure illustrates CPU-controlled high-priority communication.



Note 1. n = 0

Figure 19.46 Example of CPU-Controlled High-Priority Communication

(1) By setting CSIH0CTL0.CSIH0JOBE = 1 during low-priority communication, start of high-priority communication following end-of-job data is notified, and the internal signal flag is set.

(2) When end-of-job data is detected, the current low-priority communication is aborted and the INTCSIH0TIJS interrupt occurs. An internal signal, the JOB completion flag is cleared due to the abort of communication, and memory mode is automatically switched to direct access mode for the subsequent high-priority communication.

(3) The CPU detects the interrupt and starts communication by writing the first transmission data of high-priority communication to CSIH0TX0W or CSIH0TX0H.

(4) When end-of-job data is detected, communication is aborted. At this time, because the internal signal, end-of-job flag is set to 0, the CSIH determines that the next communication is low-priority and switches memory mode to transmit-only buffer mode automatically, and then resumes the aborted low-priority communication.

(5) Same as (1) above.

(6) Same as (2) above.

(7) The CPU detects an interrupt and starts communication by writing the first transmission data of high-priority communication to CSIH0TX0W or CSIH0TX0H. The CPU sets CSIH0CTL0.CSIH0JOBE = 1 again to notify that the next communication is high-priority.

(8) When end-of-job data is detected, communication is aborted and the INT_CSIH0TIJC interrupt is generated. At this time, the CPU determines that the subsequent communication is high-priority because the internal signal JOB completion flag is 1, and waits for communication to start.

(9) Same as (3) above.

(10) Same as (4) above.

CAUTION

Memory mode is switched automatically when communication is changed from low priority to high priority (switch from transmit-only buffer mode to direct access mode) and from high priority to low priority (switch from direct access mode to transmit-only buffer mode).

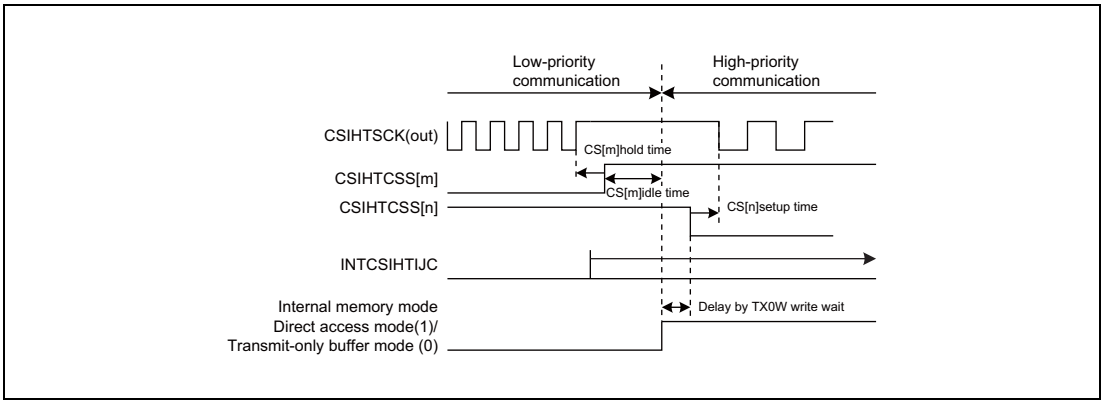


Figure 19.47 Transition from Low-Priority Mode to High-Priority Mode

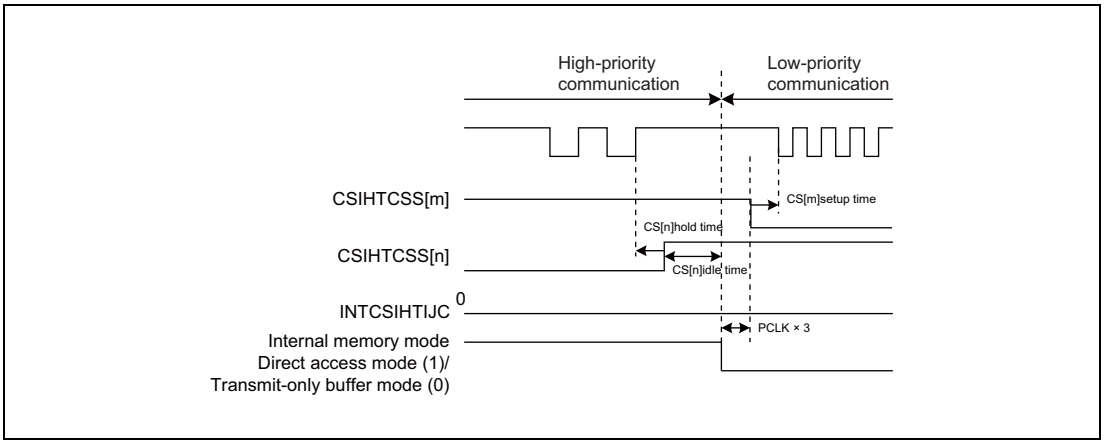


Figure 19.48 Transition from High-Priority Mode to Low-Priority Mode

Do not conduct write operation of communication data or CSIH0CTL0.CSIH0JOBE bit operation during setting prohibit period to switch low and high priority communication mode correctly.

CSIH0TX0W register write inhibited period:

- Period from when CSIH0JOBE bit is set for switching to high priority communication mode to when CSIHTIJC interrupt is detected.
- Period from when the last data of high priority communication (End of JOB data) is written to when the CSIH0HPST state = 0 is detected.

CSIH0JOBE register write inhibited period:

- Period from when CSIH0JOBE bit is set for switching to high priority communication mode to when INTCSIHTIJC interrupt is detected.

During high communication mode period, there is no setting prohibit period for CSIH0JOBE bit. It is possible to set CSIH0JOBE bit before writing communication data. For example, to communicate multiple JOB data in high priority mode, it is possible to set CSIH0JOBE bit before writing the first communication data.

CAUTION

For CSIH0 only:
When CSIH0JOBE bit is set right before the last communication of high priority communication ends, different operations are required depending on the internal detection timing of CSIH0JOBE bit setting. When CSIH0JOBE bit setting is detected before the last bit communication, high priority communication mode continues.

When setting of the CSIH0JOBE bit is detected after the transfer of the last bit is completed, the mode temporarily returns to low priority communications. After detection of End of JOB data in low priority communications, the mode changes back to high priority communications.

19.5.15 Enforced Chip Select Idle Setting

This macro is able to insert an idle state between the two consecutive transfer data by the setting of CSIHnCFGx.CSIHnIDLx. Detail is as follows.

1. When CSIHnCFGx.CSIHnIDLx = 0
   If a next CSIHTCSSx is the same as the previous one, an idle state is not inserted and an inter-data time is inserted.
   If a next CSIHTCSSx is different from the previous one, an idle state is inserted.
2. When CSIHnCFGx.CSIHnIDLx = 1
   An idle state is always inserted even if a next CSIHTCSSx is not different from the previous one.

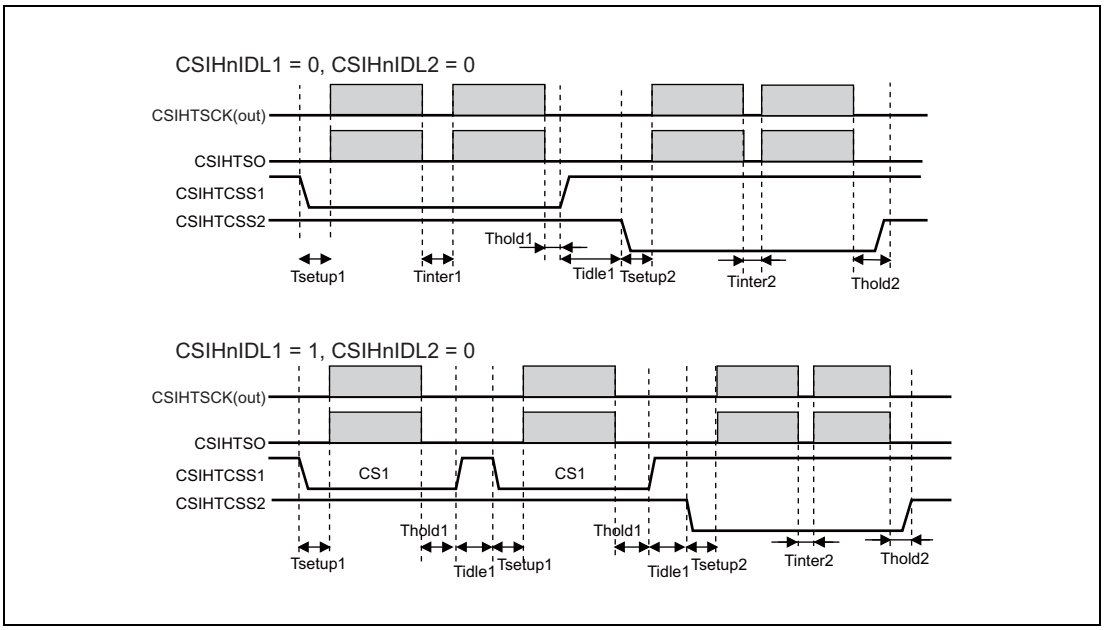


Figure 19.49 Enforced Chip Select Idle Example

CAUTION

CSIH0 only: If high priority communication function controlled by CPU is validated (CSIHnCTL1.CSIHnPHE = 1), when switch from low priority communication mode to high priority communication mode or switch from high priority communication mode to low priority communication mode, IDLE state is inserted regardless of the setting of IDLx bit.

19.6 Operating Procedures

The examples and procedures below are described according to the memory mode in the following order:

- Direct access mode
- Transmit-only buffer mode
- Dual buffer mode
- FIFO mode

19.6.1 Procedures in Direct Access Mode

Two examples for a master are provided, one with job mode disabled, and the other one with job mode enabled.

19.6.1.1 Transmit/Receive in Master Mode when Job Mode is Disabled

The procedures below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = $1000_{B)}$.
- Transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)

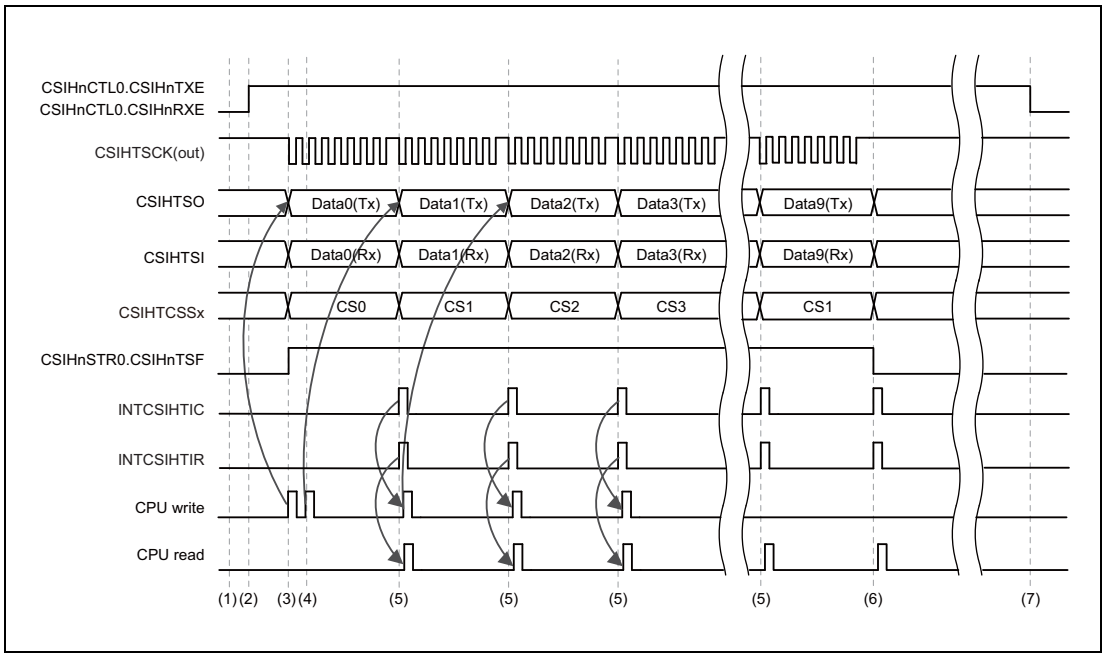


Figure 19.50 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 0

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
   Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), CSIHnRXE = 1 (permits reception), and CSIHnMBS = 1 (Direct access mode selection).
3. Write the first data to be sent to the transmission register, CSIHnTX0W. Within the same write operation, activate CS0. Transmission starts automatically when the first data becomes available.
4. Write the second data to CSIHnTX0W. If required, you can change the CS to address a different device. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After every transmission of a data the interrupts INTCSIHTIC and INTCSIHTIR are generated:
   – INTCSIHTIC indicates that the next data can be written to CSIHnTX0W.
   – INTCSIHTIR indicates that the reception register, CSIHnRX0W must be read.
6. No more write action is required after completion of data 8. data 9 (the last data) has been written in advance.
   However, reception register CSIHnRX0W must be read after completion of writing data 8 and 9.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

19.6.1.2 Transmit/Receive in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- Transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)
- Two jobs, each of them sends three data.

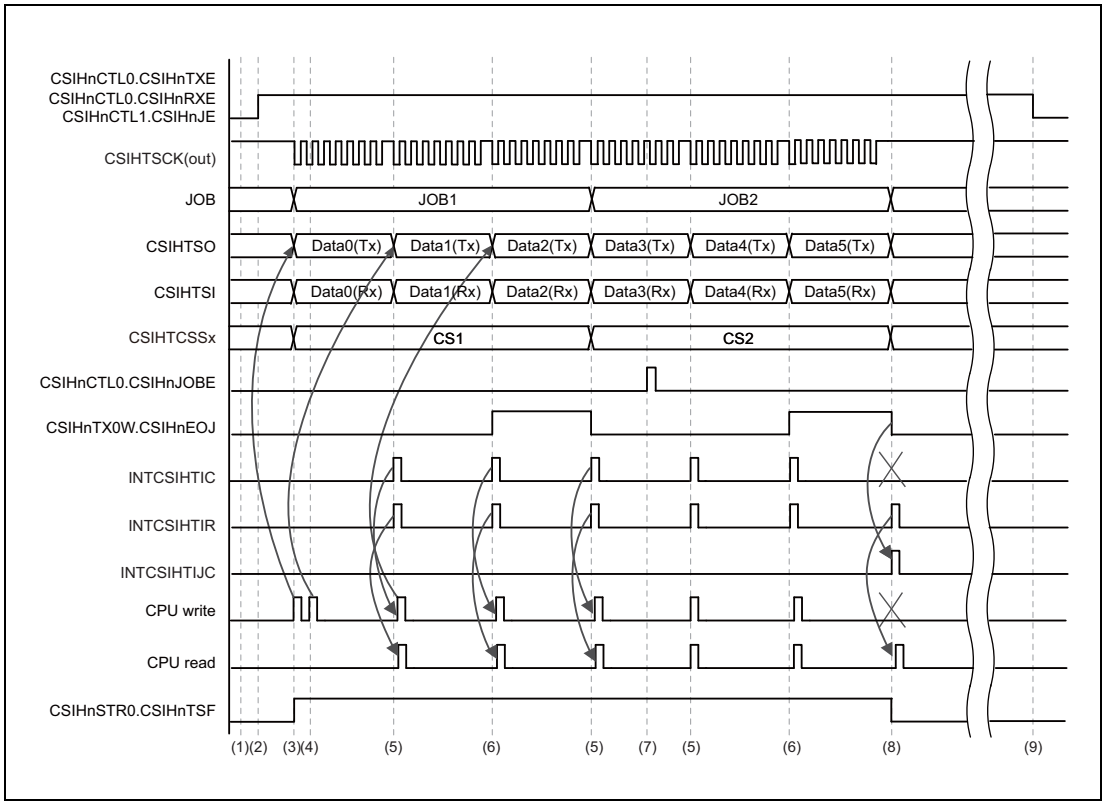


Figure 19.51 Master in Direct Access Mode, CSIHnCTL1.CSIHnJE = 1

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CS1 to CS2.
   Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
2. In the CSIHnCTL0 register, set the bits CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), CSIHnRXE = 1 (permits the reception), and CSIHnMBS = 1 (selects direct access mode).
3. Write the first data to be sent to the transmission register CSIHnTX0W. Transmission starts automatically when the first data becomes available.
   The CSIHnSTR0.CSIHnTSF flag indicates that communication is in progress.
4. Write the second data to CSIHnTX0W. Writing the second data immediately after the first one avoids unnecessary delays between the data.
5. After every data transmission/reception, the interrupt requests INTCSIHTIC and INTCSIHTIR are generated.
   - INTCSIHTIC indicates that the next data can be written to CSIHnTX0W.
   - INTCSIHTIR indicates that the reception register, CSIHnRX0W must be read.
6. Setting CSIHnTX0W.CSIHnEOJ = 1 indicates that the last data of the current job is sent. After that, the next job may begin.
7. By setting CSIHnCTL0.CSIHnJOBE = 1, communication is forced to stop at the end of the current job (JOB2).

8. After the forced stop of communication, the interrupt request, INTCSIHTIC is replaced by INTCSIHTIJC. INTCSIHTIR is generated as usual.
   The interrupt request, INTCSIHTIJC indicates a forced stop of communication at the end of the current job.
   The interrupt request, INTCSIHTIC is not generated. Additionally, the transmission data available in the CSIHnTX0W register is not sent.
9. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.
   To start another transmission without stopping communication, perform steps 3 and later.

19.6.2 Procedures in Transmit-Only Buffer Mode

Two examples for a master is provided, one with job mode disabled, and the other one with job mode enabled.

19.6.2.1 Transmit/Receive in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0,CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)

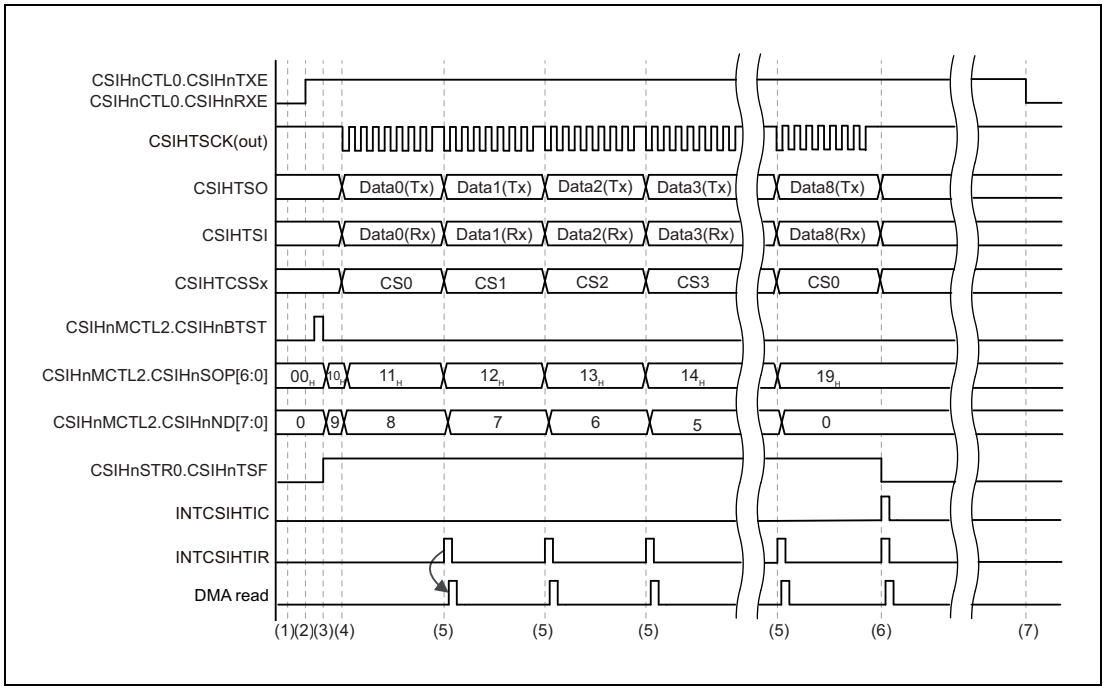


Figure 19.52 Master in Transmit-Only Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The procedure of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
   Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 10_B (transmit-only buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure the send pointer and the number of data by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission/reception is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. After every data reception, the interrupt request, INTCSIHTIR is generated. INTCSIHTIR indicates that the reception register, CSIHnRX0W must be read.
6. When all transmissions are complete, the interrupt request, INTCSIHTIC is generated.

7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

19.6.2.2 Transmit/Receive in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- The number of data is 8 (CSIHnMCTL2.CSIHnND[7:0] = 08_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0,CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)

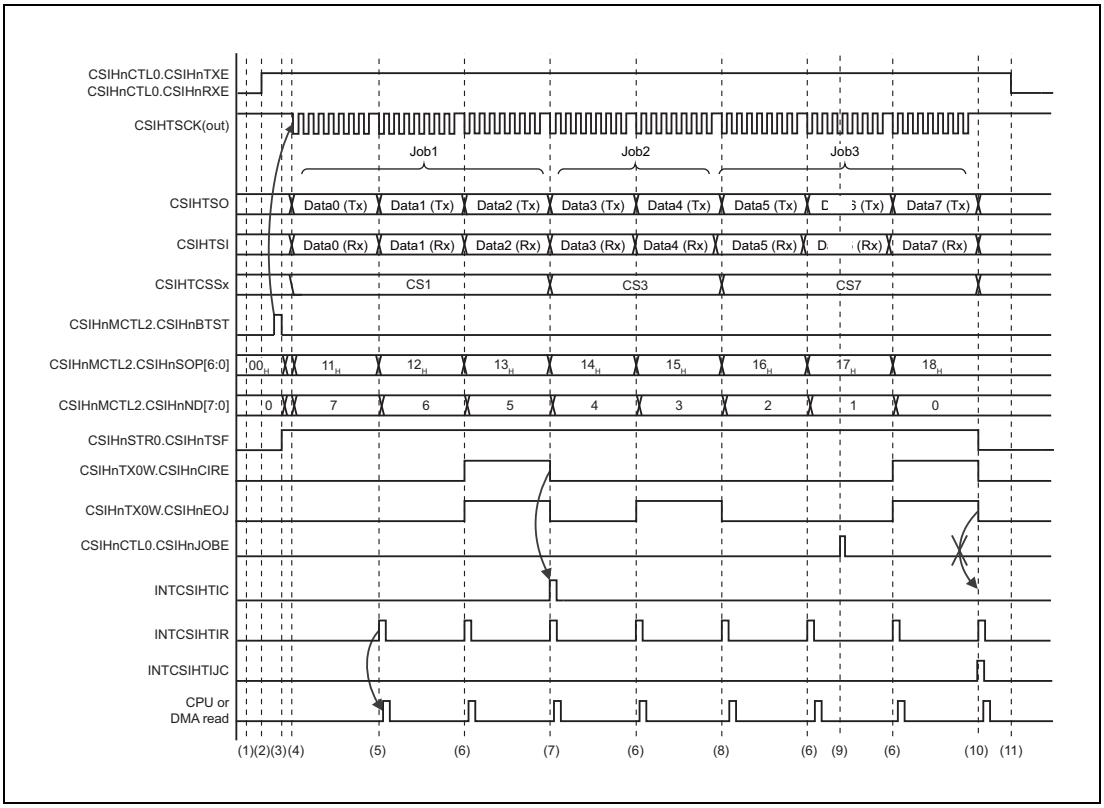


Figure 19.53 Master in Transmit-Only Buffer Mode, CSIHnCTL1.CSIHnJE = 1

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0, CSIHTCSS3, and CSIHTCSS7.
   Specify the transfer mode and job mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 10_B (transmit -only buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure the send pointer and the number of data by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. After every data reception, the interrupt request, INTCSIHTIR is generated. INTCSIHTIR indicates that the reception register, CSIHnRX0W must be read.
6. The CSIHnTX0W.CSIHnEOJ = 1 setting indicates that the last data of the current job is sent.
7. The interrupt request INTCSIHTIC is generated. INTCSIHTIC indicates that the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
8. The INTCSIHTIC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CHABnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
9. By setting CSIHnCTL0.CSIHnJOBE = 1, communication is forced to stop at the end of JOB3.
10. After the forced stop of communication, interrupt requests INTCSIHTIJC and INTCSIHTIR are generated at the end of job3.
    The INTCSIHTIJC interrupt request indicates a forced stop of communication at the end of the current job.
    The INTCSIHTIC interrupt request is not generated because the INTCSIHTIJC interrupt request is generated instead of the INTCSIHTIC interrupt request. Additionally, the transmission data available in the CSIHnTX0W register is not sent.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

19.6.3 Procedures in Dual Buffer Mode

Examples when job mode is enabled in master mode, disabled in master mode, and disabled in slave mode are provided below.

19.6.3.1 Transmit/Receive in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CHABnCFGx.CSIHnDIRx = 0).
- Default clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)

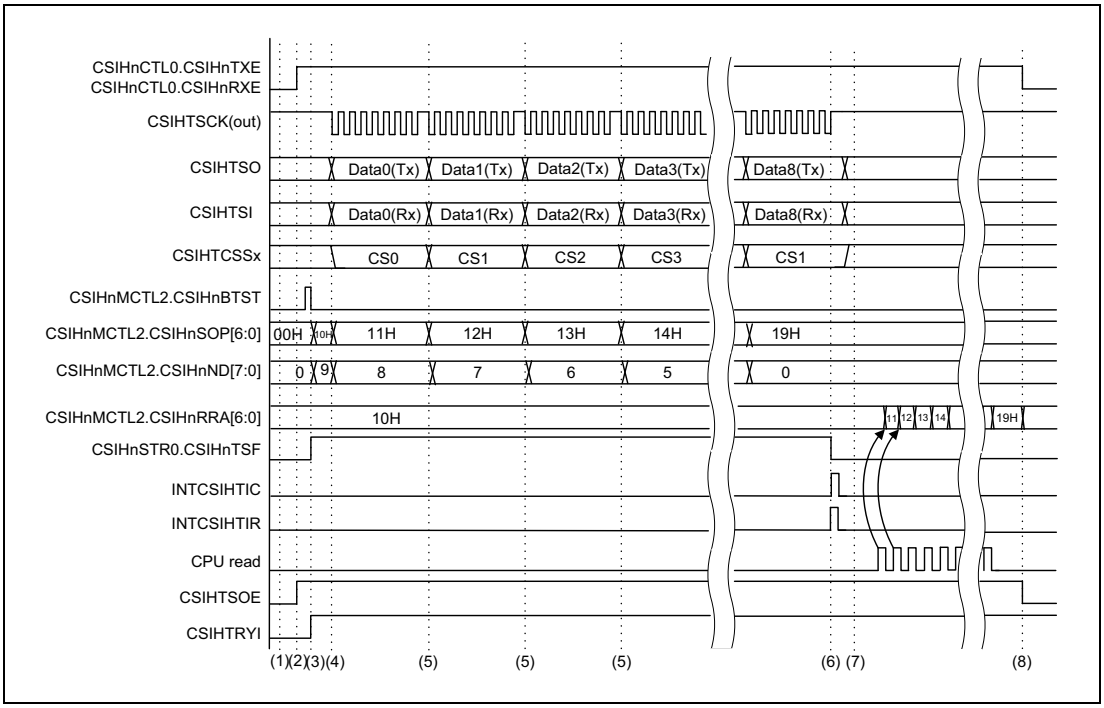


Figure 19.54 Master in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
   Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0]. Set CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits the reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure communication by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Permit buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started. TheCSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. This is repeated until the last data is transmitted/received.
   The interrupt requests, INTCSIHTIC and INTCSIHTIR are not generated.
6. When the last data is transmitted/received, the interrupt requests, INTCSIHTIC and INTCSIHTIR are generated.
   The CPU starts to read the received data from the receive buffer.
7. The start address of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0] (CSIHnRRA[6:0] is set to 10_H by the software in this figure). These bits are incremented after the reading of every data.
8. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

19.6.3.2 Transmit/Receive in Master Mode when Job Mode is Enabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- The number of data is 8 (CSIHnMCTL2.CSIHnND[7:0] = 08_H).
- The transfer start address is 00_H (CSIHnMCTL2.CSIHnSOP[6:0] = 00_H).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)

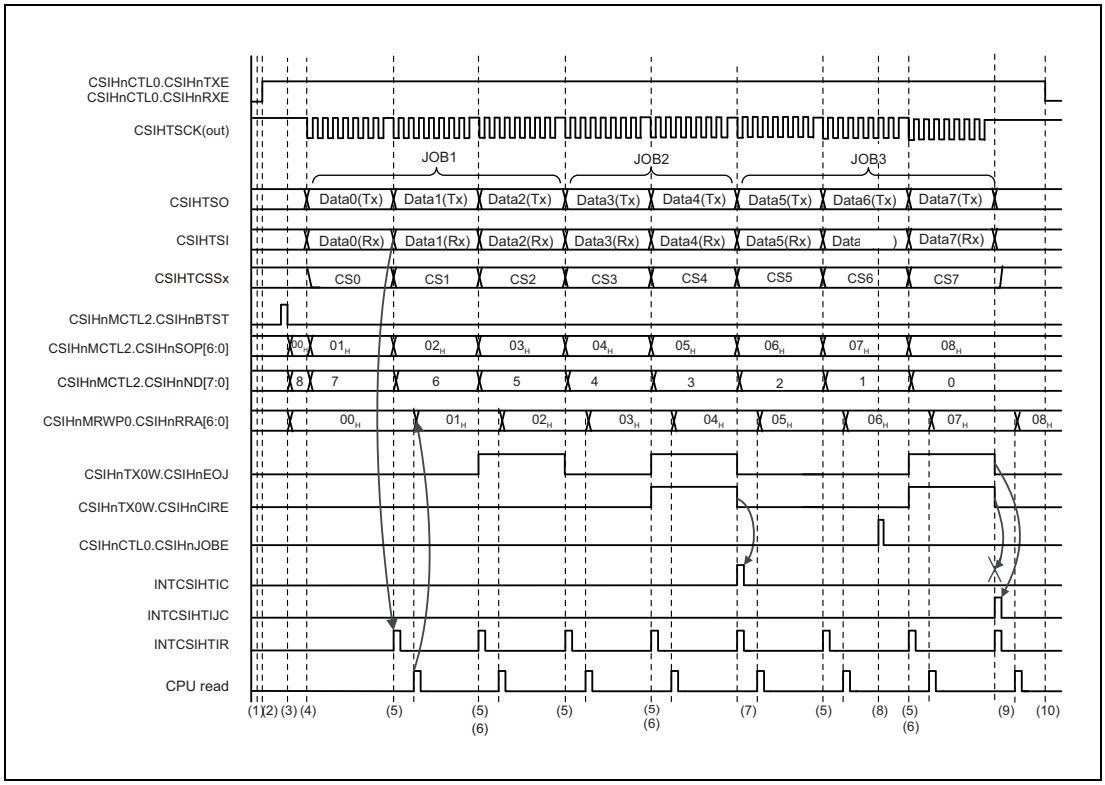


Figure 19.55 Master in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 1

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. This example uses chip select signals CSIHTCSS0 to CSIHTCSS7.
   Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits the reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Configure communication by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Start the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started.
   TheCSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission. This is repeated until the last data is transmitted/received.

5. The INTCSIHTIR interrupt request is generated everytime a data is received.
   The INTCSIHTIC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
6. CSIHnTX0W.CSIHnEOJ = 1 indicates that the last data of the current job is sent.
7. The INTCSIHTIC interrupt request is generated. INTCSIHTIC indicates that the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
8. By setting CSIHnCTL0.CSIHnJOBE = 1, communication is forced to stop at the end of JOB3.
9. After the forced stop of communication, interrupt requests, INTCSIHTIJC and INTCSIHTIR are generated at the end of JOB3.
   The INTCSIHTIJC interrupt request indicates a forced stop of communication at the end of the current job.
   The INTCSIHTIC interrupt request is not generated because the INTCSIHTIJC interrupt request is generated instead of the INTCSIHTIC interrupt request. Additionally, the transmission data available in register CSIHnTX0W is not sent.
10. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

19.6.3.3 Transmit/Receive in Slave Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCTL1.CSIHnCKR = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- The number of data is 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H).
- The transfer start address is 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H).
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)
- Handshake function enable (CSIHnCTL1.CSIHnHSE = 1)
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

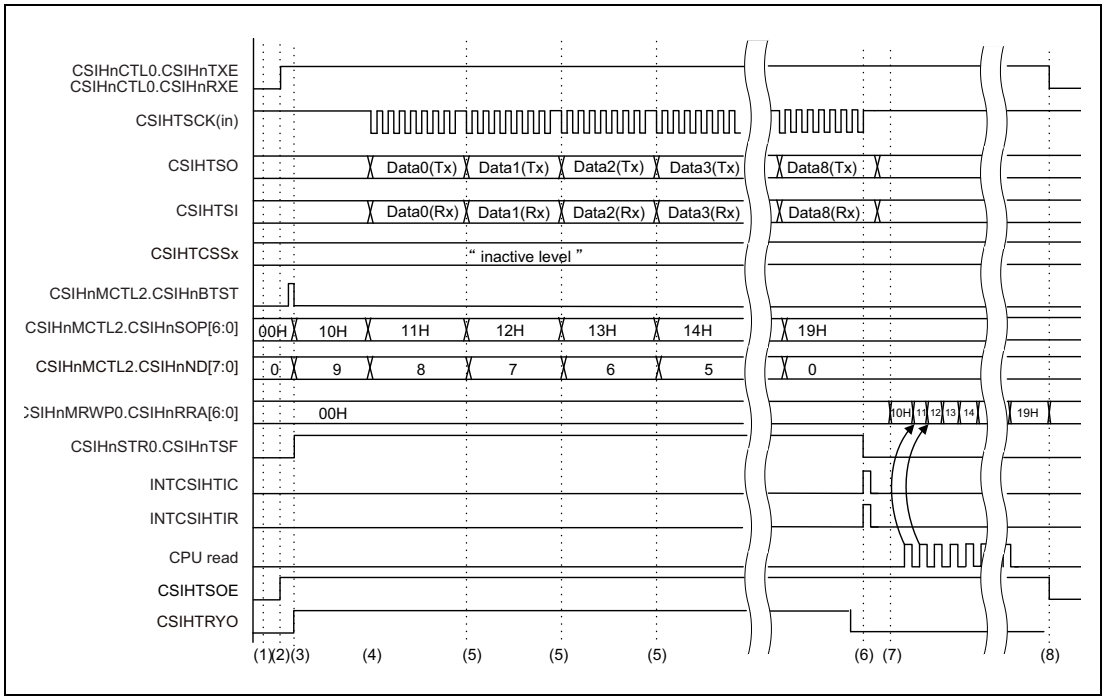


Figure 19.56 Slave in Dual Buffer Mode, CSIHnCTL1.CSIHnJE = 0

NOTE

The process of writing the data into the buffer is not described.

Procedure:

1. Configure the communication protocol in register CSIHnCFG0.
   Specify the transfer and operating modes by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2.
   Configure the memory mode in CSIHnMCTL0.CSIHnMMS[1:0].
   Set CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (dual buffer mode).
2. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits the reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
3. Specify the transfer start address by setting the CSIHnMCTL2.CSIHnSOP[6:0] bits and the number of data by setting the CSIHnMCTL2.CSIHnND[7:0] bits. Permit the buffer transfer by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission is started when the input clock from the master is received. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission.
5. This is repeated until the last data is transmitted/received.
   The interrupt requests, INTCSIHTIC and INTCSIHTIR are not generated because transmission data is sent from the buffer, and received data is stored in the buffer.
6. When the last data is transmitted/received, the interrupt requests, INTCSIHTIC and INTCSIHTIR are generated.
   The CPU starts to read the received data that is stored in the receive buffer.

7. The start address of the read access is specified in CSIHnMRWP0.CSIHnRRA[6:0] (CSIHnRRA[6:0] is set to 10_H by the software in this figure). These bits are incremented after the reading of every data.
8. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

19.6.4 Procedures in FIFO Mode

Two examples for a master is provided, one with job mode disabled, the other one with job mode enabled.

19.6.4.1 Transmit/Receive in Master Mode when Job Mode is Disabled

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0).
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is disabled (CSIHnCTL1.CSIHnJE = 0).
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)

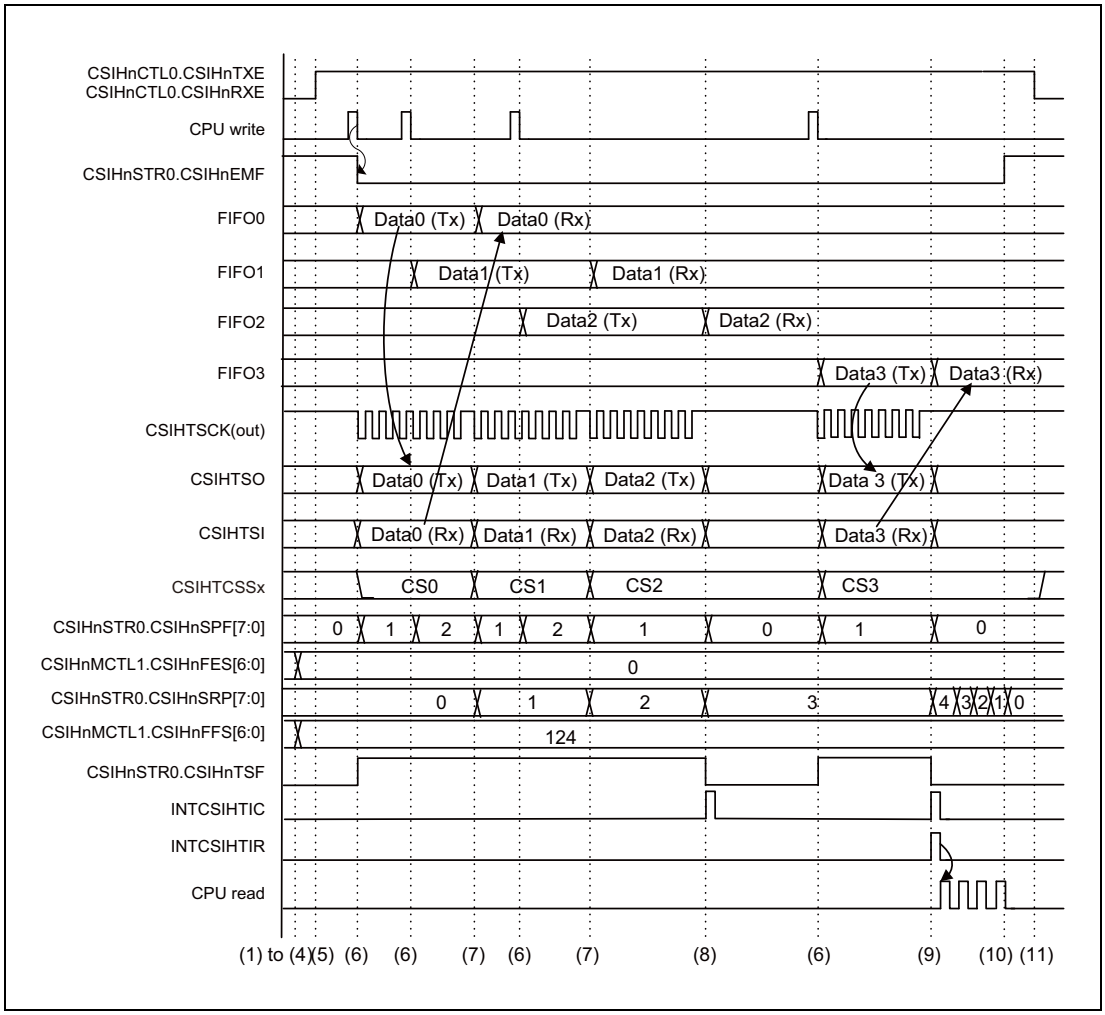


Figure 19.57 Master in FIFO Mode, CSIHnCTL1.CSIHnJE = 0

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. Specify the job mode disable and master mode by setting the corresponding bits in registers CSIHnCTL1 and CSIHnCTL2. Specify the FIFO mode by CSIHnMCTL0.CSIHnMMS[1:0] = 00B. This example uses chip select signals CSIHTCSS0 to CSIHTCSS3.
2. Set CSIHnSTCR0.CSIHnPCT = 1 to clear all buffer pointers.
3. Check that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00_H.
4. The CSIHnMCTL1.CSIHnFES [6:0] bits specify the condition for the INTCSIHTIC interrupt output.
   The CSIHnFFS[6:0] bits in the same register specify the condition for the INTCSIHTIR interrupt.
5. Set CSIHnCTL0.CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.

6. Write the first transmit data to the transmit register CSIHnTX0W. Transmission starts automatically when the first data becomes available.
   Check that CSIHnSTR0.CSIHnEMF = 0.
7. The current transmission is completed. As the CSIHnFES[6:0] bits are not the same as the CSIHnSPF[7:0] bits, the interrupt request INTCSIHTIC is not generated.
8. As the CSIHnFES[6:0] bits are the same as the CSIHnSPF[7:0] bits, the interrupt request INTCSIHTIC is generated.
9. When CSIHnFFS[6:0] = 128 - CSIHnSRP[7:0], the interrupt request INTCSIHTIR is generated. Since CSIHnFES[6:0] is CSIHnSPF[7:0], the interrupt request INTCSIHTIC is generated. After the generation of an interrupt, the CPU starts reading received data that is stored in the receive buffer.
10. When the CPU completes reading the received data that is stored in the receive buffer, CSIHnSTR0.CSIHnEMF is set to 1 and the FIFO buffer will be empty.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. In addition, if communication is not performed, set CSIHnCTL0.CSIHnPWR = 0 to minimum the power consumption of CSIHn.

19.6.4.2 Transmit/Receive Mode when Job Mode is Enabled in Master Mode

The procedure below is based on the assumption that:

- The transmission data length is 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B).
- The transmission direction is MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No interrupt delay (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
- JOB1 consists of four data, JOB2 consists of three data, and JOB3 consists of five data.
- Normal INTCSIHTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)

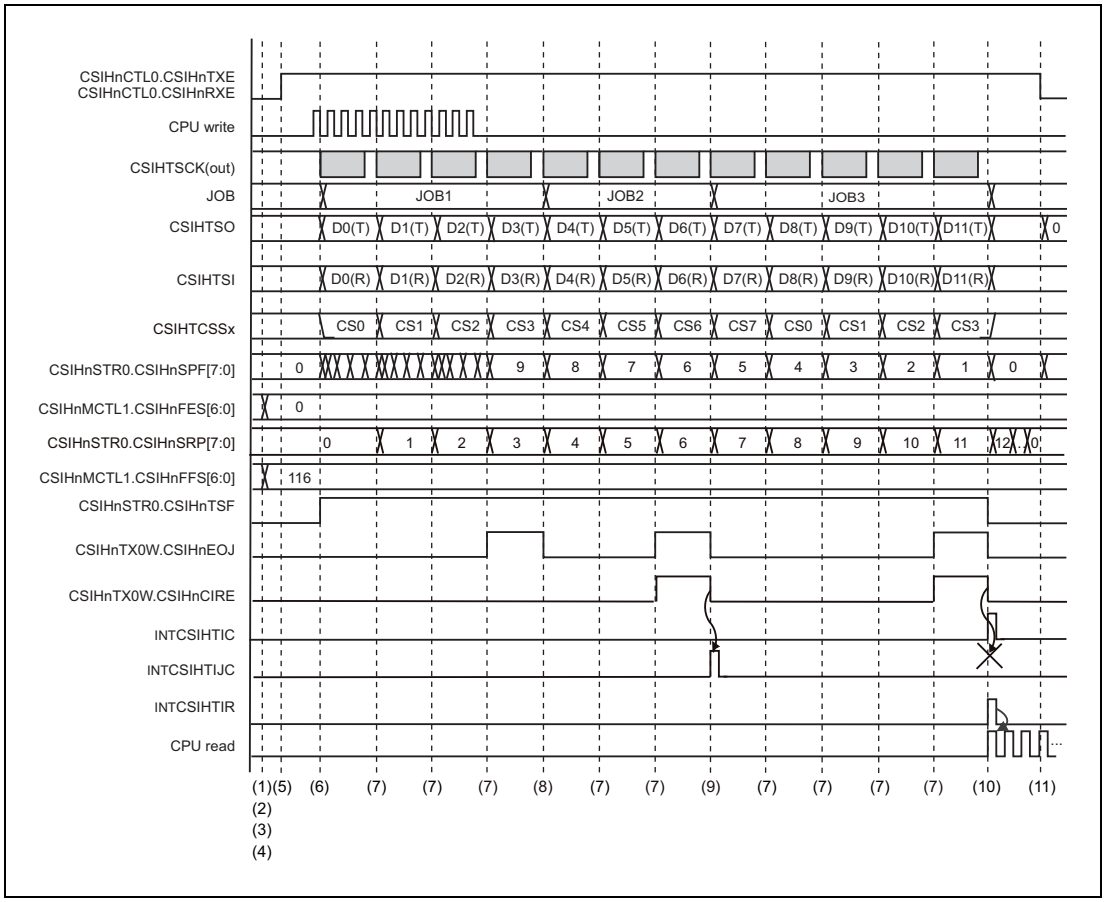


Figure 19.58 Master in FIFO Mode, CSIHnCTL1.CSIHnJE = 1

Procedure:

1. Configure the communication protocol in the CSIHnCFGx register. Set job mode disenable and master mode in the bits corresponding to CSIHnCTL1 and CSIHnCTL2 registers. Set FIFO mode by setting CSIHnMCTL0.CSIHnMMS[1:0] to 00_B. This example uses chip select signals CSIHTCSS0 to CSIHTCSS7.
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all buffer pointers.
3. Make sure that CSIHnSTR0.CSIHnFLF is set to 0, CSIHnSTR0.CSIHnEMF is set to 1, and CSIHnSTR0.CSIHnSPF[7:0] is set to 00_H.
4. With CSIHnMCTL1.CSIHnFES[6:0], specify the conditions for generating the INTCSIHTIC interrupt request. With CSIHnMCTL1.CSIHnFFS[6:0], specify the conditions for generating the INTCSIHTIR interrupt request.
5. In the CSIHnCTL0 register, set CSIHnPWR = 1 (enables the clock), CSIHnTXE = 1 (permits transmission), and CSIHnRXE = 1 (permits reception). The CSIHnCTL0.CSIHnMBS bit must be cleared.
6. Write the first data to be sent to the CSIHnTX0W transmission register. Transmission starts automatically when the first data becomes available.
   Make sure CSIHnSTR0.CSIHnEMF is set to 0.

7. The current transmission is completed. Since CSIHnFES[6:0] is not CSIHnSPF[7:0], the interrupt request INTCSIHTIC is not generated.
8. The INTCSIHTIJC interrupt request is not generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 0.
9. The INTCSIHTIJC interrupt request is generated because the last data of the current job (CSIHnTX0W.CSIHnEOJ = 1) was sent with CSIHnTX0W.CSIHnCIRE = 1.
10. The INTCSIHTIC interrupt request is generated because CSIHnFES[6:0] is CSIHnSPF[7:0]. INTCSIHTIC is generated so that INTCSIHTIJC is not generated.
    When CSIHnFFS[6:0] is 128 - CSIHnSRP[7:0], the interrupt request INTCSIHTIR is generated. After the generation of the INTCSIHTIR interrupt, CPU starts reading the received data stored in received buffer.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to prohibit the transmit/receive operation. When no communication is conducted, set CSIHnCTL0.CSIHnPWR to 0 to minimize power consumption of the CSIHn.

Section 20 LIN/UART Interface (RLIN3)

This section contains a generic description of the LIN/UART interface (RLIN3).

The first part of this section describes all RH850/D1L/D1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RLIN3.

20.1 Features of RH850/D1L/D1M RLIN3

20.1.1 Number of Units and Channels

This microcontroller has the following number of RLIN3 units.

RLIN3 unit has one channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 20.1 Number of Units

Product Name	All products
Number of units	4
Name	RLIN3n (n = 0 to 3)

Table 20.2 Unit Configurations and Channels

Unit Name (Channel Name) RLIN3n	Channels per Unit	All products
RLIN30	1	√
RLIN31	1	√
RLIN32	1	√
RLIN33	1	√

The channel names are same as those of the corresponding units.

Table 20.3 Index

Index	Meaning
n	Throughout this section, the individual RLIN3 units are identified by the index “n” (n = 0 to 3): for example, RLN3nLCUC is the LIN control register.
b	Throughout this section, the individual transmit/receive data buffers of RLIN3n are identified by the index “b” (b = 1 to 8): for example, RLN3nLDBRb is the first stage data buffer register.

The following lists the index corresponding to each product.

Table 20.4 Index Correspondence of Each Product

Index Correspondence of Each Product
All products
b = 1 to 8

20.1.2 Register Base Address

RLIN3 base addresses are listed in the following table.

RLIN3 register addresses are given as offsets from the base addresses in general.

Table 20.5 Register Base Address

Base Address Name	Base Address
<RLIN30_base>	FFCE 0000_H
<RLIN31_base>	FFCE 1000_H
<RLIN32_base>	FFCE 2000_H
<RLIN33_base>	FFCE 3000_H

20.1.3 Clock Supply

The RLIN3 clock supply is shown in the following table.

Table 20.6 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RLIN3n	PCLK	Clock Controller C_ISO_PCLK
	LIN communication clock sources	Clock Controller C_ISO_RLIN

CAUTION

The frequency of the C_ISO_RLIN clock must not exceed the C_ISO_PCLK clock frequency. Thus the clock frequencies must fulfill following condition:

$$f_{C\_ISO\_RLIN} < f_{C\_ISO\_PCLK}$$

20.1.4 Interrupt Request

RLIN3 interrupt requests are listed in the following table.

Table 20.7 Interrupt Requests (1/2)

RLIN3n signals	Function	Connected to
RLIN30		
INTRLIN30	RLIN30 interrupt	not connected
INTRLIN30UR0	RLIN30 transmit interrupt	Interrupt Controller INTRLIN30UR0 DMA Controller trigger ID 87
INTRLIN30UR1	RLIN30 receive completion interrupt	Interrupt Controller INTRLIN30UR1 DMA Controller trigger ID 88
INTRLIN30UR2	RLIN30 status interrupt	Interrupt Controller INTRLIN30UR2
RLIN31		
INTRLIN31	RLIN31 interrupt	not connected
INTRLIN31UR0	RLIN31 transmit interrupt	Interrupt Controller INTRLIN31UR0 DMA Controller trigger ID 89
INTRLIN31UR1	RLIN31 receive completion interrupt	Interrupt Controller INTRLIN31UR1 DMA Controller trigger ID 90
INTRLIN31UR2	RLIN31 status interrupt	Interrupt Controller INTRLIN31UR2

Table 20.7 Interrupt Requests (2/2)

RLIN3n signals	Function	Connected to
RLIN32		
INTRLIN32	RLIN32 interrupt	not connected
INTRLIN32UR0	RLIN32 transmit interrupt	Interrupt Controller INTRLIN32UR0 DMA Controller trigger ID 91
INTRLIN32UR1	RLIN32 receive completion interrupt	Interrupt Controller INTRLIN32UR1 DMA Controller trigger ID 92
INTRLIN32UR2	RLIN32 status interrupt	Interrupt Controller INTRLIN32UR2
RLIN33		
INTRLIN33	RLIN33 interrupt	not connected
INTRLIN33UR0	RLIN33 transmit interrupt	Interrupt Controller INTRLIN33UR0 DMA Controller trigger ID 93
INTRLIN33UR1	RLIN33 receive completion interrupt	Interrupt Controller INTRLIN33UR1 DMA Controller trigger ID 94
INTRLIN33UR2	RLIN33 status interrupt	Interrupt Controller INTRLIN33UR2

20.1.5 Reset Sources

RLIN3 reset sources are listed in the following table. RLIN3 is initialized by these reset sources.

Table 20.8 Reset Sources

Unit Name	Reset Source
RLIN3n	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

20.1.6 External Input/output Signals

External input/output signals of RLIN3 are listed below.

Table 20.9 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
RLIN30		
RLIN30RX	RLIN30 receive data input	RLIN30RX
RLIN30TX	RLIN30 transmit data output	RLIN30TX
RLIN31		
RLIN31RX	RLIN31 receive data input	RLIN31RX
RLIN31TX	RLIN31 transmit data output	RLIN31TX
RLIN32		
RLIN32RX	RLIN32 receive data input	RLIN32RX
RLIN32TX	RLIN32 transmit data output	RLIN32TX
RLIN33		
RLIN33RX	RLIN33 receive data input	RLIN33RX
RLIN33TX	RLIN33 transmit data output	RLIN33TX

20.2 Overview

20.2.1 Functional Overview

The LIN/UART interface is a hardware LIN communication controller that supports LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602, and automatically performs frame communication and error determination.

The LIN/UART interface is provided with UART mode and can also be used as a UART.

The appropriate mode should be used for the LIN/UART interface according to the application: LIN master or UART.

LIN master

- LIN reset mode
- LIN mode (LIN master mode)
  - LIN wake-up mode
  - LIN operation mode
- LIN self-test mode

UART

- LIN reset mode
- UART mode

Table 20.10 gives the LIN/UART interface specifications.

Table 20.10 LIN/UART Interface Specifications (1/3)

Item		Specifications
	Channel count	4 channels

Table 20.10 LIN/UART Interface Specifications (2/3)

Item		Specifications	
LIN communication function	Protocol	LIN Specification Package Revision 1.3, 2.0, 2.1, 2.2, and SAE J2602	
	Variable frame structure	Master	• Break transmission width: 13 to 28 Tbits • Break delimiter transmission width: 1 to 4 Tbits • Transmission inter-byte space width (header): 0 to 7 Tbits (space between Sync field and ID field)*1 • Transmission response space width: 0 to 7 Tbits*1 • Transmission inter-byte space width: 0 to 3 Tbits (space between data bytes in response area) • Transmission wake-up width: 1 to 16 Tbits
	Checksum	• Automatic operation for both transmission and reception • Classic or enhanced selectable (for each frame)	
	Response field data byte count	Variable from 0 to 8 bytes Multi-byte (9 or more bytes) response transmission and reception also possible	
	Frame communication modes	Master	• Mode in which header transmission and response transmission/reception is started with a single transmission start request • Mode in which header transmission and response transmission are started with separate transmission start requests (frame separate mode)
	Wake-up transmission and reception	LIN wake-up mode provided • Wake-up transmission (1 to 16 Tbits) • Wake-up reception Low-level width of input signals measured	
	Status	Master	• Successful frame/wake-up transmission • Successful header transmission • Successful frame/wake-up reception*2 • Successful data 1 reception • Error detection • Operation mode (LIN reset mode, LIN wake-up mode, LIN operation mode, LIN self-test mode)
	Error status	Master	• Bit error • Checksum error • Frame timeout error/response timeout error • Physical bus error • Framing error • Response preparation error
	Baud rate selection	Baud rate conforming to the LIN specifications generated using baud rate generator	
	Test mode	Self-test mode for user evaluation	
	Interrupt function	Master	• Successful header/frame/wake-up transmission • Successful frame/wake-up reception*2 • Error detection

Table 20.10 LIN/UART Interface Specifications (3/3)

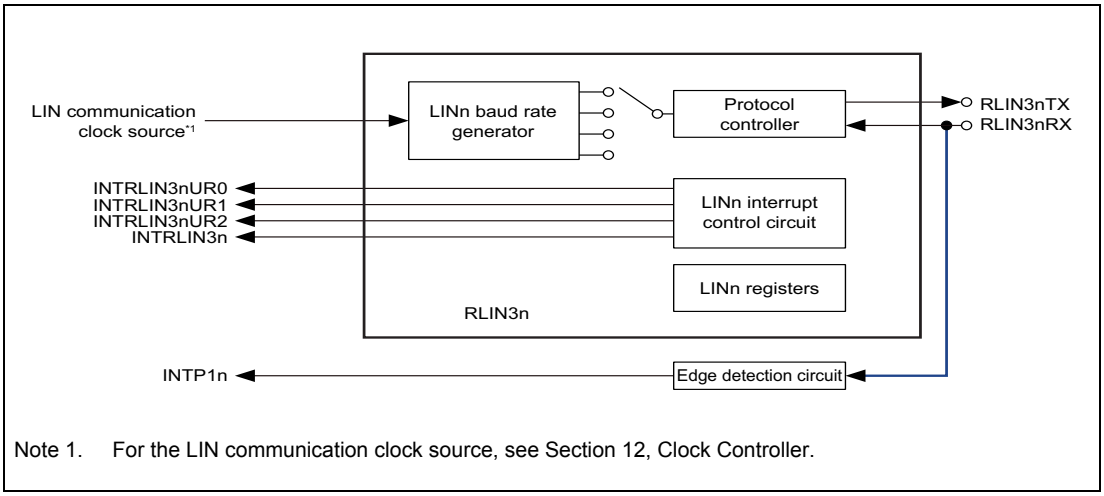
Item		Specifications
UART communication function	Data buffer	• Transmission data buffer/transmission data buffer for wait (exclusively for transmission; data length of 1. Character length of 7, 8, and 9 bits supported) • UART buffer (exclusively for transmission; variable data length from 1 to 9 bits; character length of 7 and 8 bits supported) • Reception data buffer (exclusively for reception; data length of 1. Character length of 7, 8, and 9 bits supported)
	Data format	Character length: 7 or 8 bits 9 bits including the expansion bit supported.
		Transmission stop bit: 1 or 2 bits
		Parity function: odd, even, 0, or none
		LSB- or MSB-first transfer selectable
		Reverse input/output of transmission/reception data
	Status	• Transmission status • Reception status • Successful UART buffer transmission • Error detection • Expansion bit detection • ID match • Reset mode status
	Error status	• Bit error • Framing error • Parity error • Overrun error
	Baud rate selection	With the baud rate generator incorporated, any baud rate can be set.
	When a certain expansion bit is at the expected level, the data received can be compared to the 8-bit data preset in the register.	
	The stop bit received is guaranteed (start of transmission can be delayed when start of transmission is attempted during reception of the stop bit).	
	Interrupt function	• Transmission start/complete • Reception complete • Status/error detection

Note 1. Since the same register is used for setting, the inter-byte space (header) = response space.

Note 2. For wake-up reception, the low level width of the input signal is indicated.

20.2.2 Block Diagram

Figure 20.1 shows a block diagram of the LIN/UART interface.



Note 1. For the LIN communication clock source, see Section 12, Clock Controller.

Figure 20.1 LIN/UART Interface Block Diagram

20.2.3 Description of Blocks

- RLIN3nTX, RLIN3nRX: LIN/UART interface I/O pins
- LINn baud rate generator: Generates the LIN/UART interface communication clock signal.
- LINn registers: LIN/UART interface registers
- LINn interrupt controller circuit: Controls interrupt requests generated by the LIN/UART interface

20.3 Registers

20.3.1 List of Registers

RLIN3 registers are listed in the following table.

For <RLIN3n_base>, see Section 20.1.2, Register Base Address.

Table 20.11 Registers (1/2)

Module	Register	Symbol	Address	LIN Master	UART
RLN3n	LIN wake-up baud rate selector register	RLN3nLWBR	<RLN3n_base> + 01_H	√	√
RLN3n	UART baud rate prescaler 01 register	RLN3nLBRP01	<RLN3n_base> + 02_H	—	√
RLN3n	LIN/ UART baud rate prescaler 0 register	RLN3nLBRP0	<RLN3n_base> + 02_H	√	√
RLN3n	LIN/ UART baud rate prescaler 1 register	RLN3nLBRP1	<RLN3n_base> + 03_H	√	√
RLN3n	LIN self-test control register	RLN3nLSTC	<RLN3n_base> + 04_H	√	—
RLN3n	LIN/ UART mode register	RLN3nLMD	<RLN3n_base> + 08_H	√	√
RLN3n	LIN break field configuration register/ UART configuration register	RLN3nLBFC	<RLN3n_base> + 09_H	√	√
RLN3n	LIN / UART space configuration register	RLN3nLSC	<RLN3n_base> + $0A_H$	√	√
RLN3n	LIN wake-up configuration register	RLN3nLWUP	<RLN3n_base> + $0B_H$	√	—
RLN3n	LIN interrupt enable register	RLN3nLIE	<RLN3n_base> + $0C_H$	√	—
RLN3n	LIN / UART error detection enable register	RLN3nLEDE	<RLN3n_base> + $0D_H$	√	√
RLN3n	LIN/ UART control register	RLN3nLCUC	<RLN3n_base> + $0E_H$	√	√
RLN3n	LIN / UART transmission control register	RLN3nLTRC	<RLN3n_base> + 10_H	√	√
RLN3n	LIN/ UART mode status register	RLN3nLMST	<RLN3n_base> + 11_H	√	√
RLN3n	LIN / UART status register	RLN3nLST	<RLN3n_base> + 12_H	√	√
RLN3n	LIN/ UART error status register	RLN3nLEST	<RLN3n_base> + 13_H	√	√
RLN3n	LIN data field configuration register	RLN3nLDFC	<RLN3n_base> + 14_H	√	√
RLN3n	LIN ID buffer register	RLN3nLIDB	<RLN3n_base> + 15_H	√	√
RLN3n	LIN checksum buffer register	RLN3nLCBR	<RLN3n_base> + 16_H	√	—
RLN3n	UART data 0 buffer register	RLN3nLUDB0	<RLN3n_base> + 17_H	—	√
RLN3n	LIN / UART data buffer 1 register	RLN3nLDBR1	<RLN3n_base> + 18_H	√	√
RLN3n	LIN/ UART data buffer 2 register	RLN3nLDBR2	<RLN3n_base> + 19_H	√	√
RLN3n	LIN/ UART data buffer 3 register	RLN3nLDBR3	<RLN3n_base> + $1A_H$	√	√
RLN3n	LIN / UART data buffer 4 register	RLN3nLDBR4	<RLN3n_base> + $1B_H$	√	√
RLN3n	LIN / UART data buffer 5 register	RLN3nLDBR5	<RLN3n_base> + $1C_H$	√	√
RLN3n	LIN / UART data buffer 6 register	RLN3nLDBR6	<RLN3n_base> + $1D_H$	√	√
RLN3n	LIN / UART data buffer 7 register	RLN3nLDBR7	<RLN3n_base> + $1E_H$	√	√
RLN3n	LIN/ UART data buffer 8 register	RLN3nLDBR8	<RLN3n_base> + $1F_H$	√	√
RLN3n	UART operation enable register	RLN3nLUOER	<RLN3n_base> + 20_H	—	√
RLN3n	UART option register 1	RLN3nLUOR1	<RLN3n_base> + 21_H	—	√
RLN3n	UART transmission data register	RLN3nLUTDR	<RLN3n_base> + 24_H	—	√
RLN3n	UART transmission data register L	RLN3nLUTDRL	<RLN3n_base> + 24_H	—	√
RLN3n	UART transmission data register H	RLN3nLUTDRH	<RLN3n_base> + 25_H	—	√
RLN3n	UART reception data register	RLN3nLURDR	<RLN3n_base> + 26_H	—	√
RLN3n	UART reception data register L	RLN3nLURDRL	<RLN3n_base> + 26_H	—	√

Table 20.11 Registers (2/2)

Module	Register	Symbol	Address	LIN Master	UART
RLN3n	UART reception data register H	RLN3nLURDRH	<RLN3n_base> + 27_H	—	√
RLN3n	UART wait transmission data register	RLN3nLUWTDR	<RLN3n_base> + 28_H	—	√
RLN3n	UART wait transmission data register L	RLN3nLUWTDRL	<RLN3n_base> + 28_H	—	√
RLN3n	UART wait transmission data register H	RLN3nLUWTDRH	<RLN3n_base> + 29_H	—	√

Note: √: Used, —: Not used
When writing to a register not used, write the value after reset.

20.3.2 LIN Master Related Registers

20.3.2.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			LWBR0
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.12 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 sampling 1 1 1 1: 16 sampling Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	LWBR0	Wake-up Baud Rate Select 0: In LIN wake-up mode, the clock specified in the LCKS[1:0] bit of the RLN3nLMD register is used. (for LIN1.3) 1: In LIN wake-up mode, the clock fa is used regardless of the setting in the LCKS[1:0] bit of the RLN3nLMD register. (for LIN2.x)

Set the RLN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

These bits select the number of sampling in one Tbit (reciprocal of the baud rate).
In LIN master mode (LIN/UART mode select bits in LIN mode register = 00_B), set these bits to 0000_B or 1111_B (16 sampling).

LPRS[2:0] Bits (Prescaler Clock Select)

These bits select the frequency division ratio for the prescaler.
The LIN communication clock source is divided by this prescaler.

LWBR0 Bit (Wake-up Baud Rate Select)

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. This allows the 2.5-Tbit or longer low-level width of the input signal to be measured.
When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 selects fa as the LIN system clock (fLIN) during LIN wake-up mode regardless of the setting of the

RLN3nLMD.LCKS[1:0] bit (the LCKS[1:0] bit is not changed). This allows the 2.5-Tbit or longer low-level width of the input signal to be measured.

Setting the baud rate to 19200 bps while fa is selected allows the 130 µs or longer low-level width of the input signal to be detected during LIN wake-up mode regardless of the setting of the RLN3nLMD.LCKS[1:0] bit.

20.3.2.2 RLN3nLBRP0 — LIN Baud Rate Prescaler 0 Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 02_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP0[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.13 RLN3nLBRP0 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP0[7:0]	Assuming that the value set in this register is N (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by N + 1. Setting range: 00_H to FF_H

Set the RLN3nLBRP0 register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock sources fa, fb, and fc.

Assuming that the value set in this register is N, baud rate prescaler 0 divides the frequency of the clock that is selected by the LPRS[2:0] bits by N + 1.

20.3.2.3 RLN3nLBRP1 — LIN Baud Rate Prescaler 1 Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 03_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LBRP1[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.14 RLN3nLBRP1 Register Contents

Bit Position	Bit Name	Function
7 to 0	LBRP1[7:0]	Assuming that the value set in this register is M (0 to 255), the baud rate prescaler divides the frequency of the prescaler clock by M + 1. Setting range: 00_H to FF_H

Set the RLN3nLBRP1 register when the OMM0 bit in the RLN3LMST register is 0_B (in LIN reset mode).

The value set in this register is used to control the frequency of baud rate clock source fd.

Assuming that the value set in this register is M, baud rate prescaler 1 divides the frequency of the clock that is selected by the LPRS[2:0] bits (prescaler clock select bits) by M+1.

20.3.2.4 RLN3nLSTC — LIN Self-Test Control Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 04_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LSTM
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.15 RLN3nLSTC Register Contents

Bit Position	Bit Name	Function
7 to 0	—	Writing $A7_H$, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.
0	LSTM	LIN Self-Test Mode 0: The module is not in LIN self-test mode. 1: The module is in LIN self-test mode.

The RLN3nLSTC register cancels protection of LIN self-test mode.

Set the RLN3nLSTC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Writing $A7_H$, 58_H, and 01_H successively to the RLN3nLSTC register places the module into LIN self-test mode.

When successive writing is completed thus placing LIN self-test mode to be entered, the LSTM bit is set to 1.

Do not write any other value during successive writing.

For making transition to LIN self-test mode, see Section 20.9, LIN Self-Test Mode.

When read, bits 6 to 1 return "000000_B", and bit 7 returns an undefined value.

LSTM Bit (LIN Self-Test Mode)

When transition to LIN self-test mode is completed, the LSTM bit is set to 1.
For leaving LIN self-test mode, see Section 20.9, LIN Self-Test Mode.
Writing 1 to this bit does not affect the value of the RLN3nLSTC register if it is not a part of successive writing of $A7_H$, 58_H, and 01_H.

20.3.2.5 RLN3nLMD — LIN Mode Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base>+ 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	LIOS	LCKS[1:0]		LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.16 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filtering Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4	LIOS	LIN Interrupt Output Select 0: RLIN3n interrupt is used. 1: RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are used.
3, 2	LCKS[1:0]	LIN System Clock Select b3 b2 0 0: fa (Clock generated by baud rate prescaler 0) 0 1: fb (1/2 clock generated by baud rate prescaler 0) 1 0: fc (1/8 clock generated by baud rate prescaler 0) 1 1: fd (1/2 clock generated by baud rate prescaler 1)
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 0 0: LIN master mode

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode)

LRDNFS Bit (LIN Reception Data Noise Filtering Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

LIOS Bit (LIN Interrupt Output Select)

The LIOS bit selects the number of interrupt outputs from the LIN/UART interface.
With 0 set, the RLIN3 interrupt is generated from the LIN/UART interface.
With 1 set, the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, and RLIN3n status interrupt are generated from the LIN/UART interface.
For each interrupt source, see Section 20.4, Interrupt Sources.

LCKS[1:0] Bits (LIN System Clock Select)

The LCKS[1:0] bits select the clock to be input to the protocol controller.
With 00_B set, the protocol controller is provided with fa (clock generated by baud rate prescaler 0).
With 01_B set, the protocol controller is provided with fb (1/2 clock generated by baud rate prescaler 0).

With 10_B set, the protocol controller is provided with fc (1/8 clock generated by baud rate prescaler 0).
With 11_B set, the protocol controller is provided with fd (1/2 clock generated by baud rate prescaler 1).

With 1_B is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x is used), and the RLN3nLMST register is 01_H (LIN wake-up mode), the protocol controller is provided with fa regardless of the setting of the bit (the LCKS[1:0] bit is not changed)

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD[1:0] bits select the LIN/UART interface mode.
To use the LIN/UART interface as an LIN master, set these bits to 00_B.

20.3.2.6 RLN3nLBFC — LIN Break Field Configuration Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	BDT[1:0]		BLT[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.17 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5, 4	BDT[1:0]	Transmission Break Delimiter (High Level) Width Select b5 b4 0 0: 1 Tbit 0 1: 2 Tbits 1 0: 3 Tbits 1 1: 4 Tbits
3 to 0	BLT[3:0]	Transmission Break (Low Level) Width Select b3 b0 0 0 0 0: 13 Tbits 0 0 0 1: 14 Tbits 0 0 1 0: 15 Tbits : 1 1 1 0: 27 Tbits 1 1 1 1: 28 Tbits

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Some combinations of the set values result in the length of a frame exceeding the timeout time. Set the appropriate values in this register.

BDT[1:0] Bits (Transmission Break Delimiter (High Level) Width Select)

The BDT[1:0] bits set the break delimiter (high level) width of transmission frame header.
1 Tbit to 4 Tbits can be set.

BLT[3:0] Bits (Transmission Break (Low Level) Width Select)

This BLT[3:0] bits set the break low level width of transmission frame header.
13 Tbits to 28 Tbits can be set.

20.3.2.7 RLN3nLSC — LIN Space Configuration Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base>+ $0A_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	IBHS[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W

Table 20.18 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
2 to 0	IBHS[2:0]	Inter-Byte Space (Header)/Response Space Select b2 b0 0 0 0: 0 Tbit 0 0 1: 1 Tbit 0 1 0: 2 Tbits 0 1 1: 3 Tbits 1 0 0: 4 Tbits 1 0 1: 5 Tbits 1 1 0: 6 Tbits 1 1 1: 7 Tbits

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Some combinations of the set values result in the length of a frame or a response exceeding the timeout time. Set the appropriate values in this register.

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS[1:0] bits set the width of the inter-byte space of the transmission frame response field.
0 Tbit to 3 Tbits can be set.
These bits are enabled only during response transmission; these are disabled during response reception.

IBHS[2:0] Bits (Inter-Byte Space (Header)/Response Space Select)

The IBHS[2:0] bits set the width of the inter-byte space (header) of the transmission frame header field and the response space.
0 Tbit to 7 Tbits can be set.
The response space setting is enabled only during response transmission; setting is disabled during response reception.
The inter-byte space (header) is equal to the response space.

20.3.2.8 RLN3nLWUP — LIN Wake-Up Configuration Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base>+ $0B_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	WUTL[3:0]				—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 20.19 RLN3nLWUP Register Contents

Bit Position	Bit Name	Function
7 to 4	WUTL[3:0]	Wake-up Transmission Low Level Width Select b7 b4 0 0 0 0: 1 Tbit 0 0 0 1: 2 Tbits 0 0 1 0: 3 Tbits 0 0 1 1: 4 Tbits : 1 1 0 0: 13 Tbits 1 1 0 1: 14 Tbits 1 1 1 0: 15 Tbits 1 1 1 1: 16 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

Set the RLN3nLWUP register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

WUTL[3:0] Bits (Wake-up Transmission Low Level Width Select)

The WUTL[3:0] bits set the low level width of the wake-up signal transmission.
1 Tbit to 16 Tbits can be set.

With 1 is set in the LWBR0 bit in the RLN3nLWBR register (LIN 2.x is used), fa is selected as the LIN system clock (fLIN) regardless of the setting of the RLN3nLMD.LCKS[1:0] bit (the LCKS[1:0] bit is not changed).

20.3.2.9 RLN3nLIE — LIN Interrupt Enable Register

Occurrence factors are bit errors, physical bus errors, frame/response timeout errors, framing errors, checksum errors, and response preparation errors.

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + $0C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SHIE	ERRIE	FRCIE	FTCIE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 20.20 RLN3nLIE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
3	SHIE	Successful Header Transmission Interrupt Request Enable 0: Disables successful header transmission interrupt request. 1: Enables successful header transmission interrupt request.
2	ERRIE	Error Detection Interrupt Request Enable 0: Disables error detection interrupt request. 1: Enables error detection interrupt request.
1	FRCIE	Successful Frame/Wake-up Reception Interrupt Request Enable 0: Disables successful frame/wake-up reception interrupt request. 1: Enables successful frame/wake-up reception interrupt request.
0	FTCIE	Successful Frame/Wake-up Transmission Interrupt Request Enable 0: Disables successful frame/wake-up transmission interrupt request. 1: Enables successful frame/wake-up transmission interrupt request.

Set the RLN3nLIE register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

SHIE Bit (Successful Header Transmission Interrupt Request Enable)

The SHIE bit enables or disables interrupt request upon successful transmission of a header.
With 0 set, the interrupt request for RLIN3n transmission is not generated when the HTRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n transmission is generated when the HTRC flag in the RLN3nLST register is set to 1.

ERRIE Bit (Error Detection Interrupt Request Enable)

The ERRIE bit enables or disables an interrupt request upon detection of an error.
With 0 set, the interrupt request for RLIN3n status not generated when the ERR flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n status is generated when the ERR flag in the RLN3nLST register is set to 1.
Occurrence factors are bit errors, physical bus errors, frame/response timeout errors, framing errors, checksum errors, and response preparation errors.
Detection of the bit error, physical bus error, frame/response timeout error, and framing error can be enabled or disabled using the RLN3nLEDE register.

FRCIE Bit (Successful Frame/Wake-up Reception Interrupt Request Enable)

The FRCIE bit enables or disables an interrupt request upon successful reception of a frame or a wake-up signal (counting of low level width of the input signal).
With 0 set, the interrupt request for successful RLIN3n reception is not generated when the FRC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for successful RLIN3n reception is generated when the FRC flag in the RLN3nLST register is set to 1.

FTCIE Bit (Successful Frame/Wake-up Transmission Interrupt Request Enable)

The FTCIE bit enables or disables an interrupt request upon successful transmission of a frame or a wake-up signal.
With 0 set, the interrupt request for RLIN3n transmission is not generated when the FTC flag in the RLN3nLST register is set to 1.
With 1 set, the interrupt request for RLIN3n transmission is generated when the FTC flag in the RLN3nLST register is set to 1.

20.3.2.10 RLN3nLEDE — LIN Error Detection Enable Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + $0D_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LTES	—	—	—	FERE	FTERE	PBERE	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Table 20.21 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7	LTES	Timeout Error Select 0: Frame timeout error 1: Response timeout error
6 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	FTERE	Timeout Error Detection Enable 0: Disables frame/response timeout error detection. 1: Enables frame/response timeout error detection.
1	PBERE	Physical Bus Error Detection Enable 0: Disables physical bus error detection. 1: Enables physical bus error detection.
0	BERE	Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection.

Note: Set FERE bit and BERE bit to 1.

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode)

LTES Bit (Timeout Error Select)

The LTES bit selects the timeout function to be used.
With 0 set, the timeout function applies to frame timeout.
With 1 set, the timeout function applies to response timeout.
For details on the timeout error, see Section 20.7.6, Error Status.

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.
With 0 set, the framing error is not detected.
With 1 set, the framing error is detected.
When this bit is set to 1, the detection result is indicated in the FER flag in the RLN3nLEST register.
For details on the framing error, see Section 20.7.6, Error Status.

FTERE Bit (Timeout Error Detection Enable)

The FTERE bit enables or disables detection of the frame timeout error or the response timeout error.
With 0 set, the frame timeout error or response timeout error is not detected.
With 1 set, the frame timeout error or response timeout error is detected.
When this bit is set to 1, the detection result is indicated in the FTER flag in the RLN3nLEST register.
With the LTES bit, either the frame timeout error or response timeout error can be selected.
Do not use the timeout error if response data of 9 bytes or more is to be transmitted or received.
For details on the timeout error, see Section 20.7.6, Error Status.

PBERE Bit (Physical Bus Error Detection Enable)

The PBERE bit enables or disables detection of the physical bus error.
With 0 set, the physical bus error is not detected.
With 1 set, the physical bus error is detected.
When this bit is set to 1, the detection result is indicated in the PBER flag in the RLN3nLEST register.
For details on the physical bus error, see Section 20.7.6, Error Status.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.
With 0 set, the bit error is not detected.
With 1 set, the bit error is detected.
When this bit is set to 1, the detection result is indicated in the BER flag in the RLN3nLEST register.
For details on the bit error, see Section 20.7.6, Error Status.

20.3.2.11 RLN3nLCUC — LIN Control Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + $0E_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OM1	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 20.22 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	OM1	LIN Mode Select 0: LIN wake-up mode is caused. 1: LIN operation mode is caused.
0	OM0	LIN Reset 0: LIN reset mode is caused. 1: LIN reset mode is canceled.

Set the RLN3nLCUC register to 01_H to cause a transition to LIN wake-up mode after canceling LIN reset mode, and set the register to 03_H to cause a transition to LIN operation mode.

In LIN self-test mode, set the RLN3nLCUC register to 03_H after a transition to LIN self-test mode is completed.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM1 Bit (LIN Mode Select)

The OM1 bit selects the specific operation mode (either LIN wake-up mode or LIN operation mode) after canceling LIN reset mode.
With 0 set, LIN wake-up mode is caused.
With 1 set, LIN operation mode is caused.
This bit is valid only when the OMM0 bit in the RLN3nLMST register is 1.
Writing a value to this bit is disabled while the FTS bit in the RLN3nLTRC register is 1.

OM0 Bit (LIN Reset)

The OM0 bit selects either causing a transition to LIN reset mode or canceling LIN reset mode.
With 0 set, LIN reset mode is caused.
With 1 set, LIN reset mode is canceled.

20.3.2.12 RLN3nLTRC — LIN Transmission Control Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	FTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 20.23 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	RTS	Response Transmission/Reception Start 0: Response transmission/reception is stopped in frame separate mode. 1: Response transmission/reception is started in frame separate mode.
0	FTS	Frame Transmission/Wake-up Transmission /Reception Start 0: Frame Transmission/wake-up transmission/reception is stopped. 1: Frame Transmission/wake-up transmission/reception is started.

RTS Bit (Response Transmission/Reception Start)

Set the RTS bit to 1 in frame separate mode after header transmission is started (FTS bit is 1) and response transmission data is ready. Once set, this bit is automatically cleared to 0 upon completion of frame communication (including error detection) or transition to LIN reset mode.
Only 1 can be written to this bit; 0 cannot be written.
To write 1 to this bit, write 02_H to the RLN3nLTRC register using the store instruction.
Writing a value to this bit is disabled when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).
Writing a value to this bit is disabled when the FTS bit is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, set this bit to 1 each time a data group (variable from 0 to 8 bytes) is transmitted or received. Once set, this bit is automatically cleared to 0 at the end of communication or transition to LIN reset mode.

FTS Bit (Frame Transmission/Wake-up Transmission/Reception Start)

Set the FTS bit to 1 to start frame transmission and reception.
Also set this bit to 1 to allow wake-up transmission and reception (counting of the low level width of the input signal).
Only 1 can be written to this bit; 0 cannot be written.
Writing a value to this bit is disabled when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

This bit is set to 0 upon completion of frame or wake-up communication (including error detection) and transition to LIN reset mode.

20.3.2.13 RLN3nLMST — LIN Mode Status Register

Access: This register can only be read in 8-bit units

Address: <RLIN3n_base> + 11_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OMM1	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 20.24 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	OMM1	LIN Mode Status Monitor 0: The module is in LIN wake-up mode. 1: The module is in LIN operation mode.
0	OMM0	LIN Reset Status Monitor 0: The module is in LIN reset mode. 1: The module is not in LIN reset mode.

OMM1 Bit (LIN Mode Status Monitor)

The OMM1 bit indicates the current operating mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

20.3.2.14 RLN3nLST — LIN Status Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 12_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	HTRC	D1RC	—	—	ERR	—	FRC	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W

Table 20.25 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7	HTRC	Successful Header Transmission Flag 0: Header transmission has not been completed. 1: Header transmission has been completed.
6	D1RC	Successful Data 1 Reception Flag 0: Data 1 reception has not been completed. 1: Data 1 reception has been completed.
5, 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
1	FRC	Frame/Wake-up Reception Complete Flag 0: Frame or wake-up reception has not been completed. 1: Frame or wake-up reception has been completed.
0	FTC	Frame/Wake-up Transmission Complete Flag 0: Frame or wake-up transmission has not been completed. 1: Frame or wake-up transmission has been completed.

The RLN3nLST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

HTRC Flag (Successful Header Transmission Flag)

Only 0 can be written to the HTRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The HTRC flag is set to 1 upon completion of header transmission. Here, an interrupt request for RLN3n transmission is generated if the SHIE bit in the RLN3nLIE register is 1 (interrupt is enabled).
To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

D1RC Flag (Successful Data 1 Reception Flag)

Only 0 can be written to the D1RC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The D1RC flag is set to 1 upon completion of data 1 reception. Here, an interrupt request is not

generated. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time data 1 of a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

ERR Flag (Error Detection Flag)

The ERR flag is set to 1 upon detection of an error (when the value of at least one of the flags of the RLN3nLEST register is set to 1). Here, an interrupt request for RLN3n status is generated if the ERRIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the RPER, CSER, FER, FTER, PBER, and BER flags in the RLN3nLEST register in LIN operation mode or LIN wake-up mode. This clears the ERR flag to 0.

FRC Flag (Frame/Wake-up Reception Complete Flag)

Only 0 can be written to the FRC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FRC flag is set to 1 upon completion of frame or wake-up reception. Here, an interrupt request for RLN3n reception complete is generated if the FRCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.
When response data of 9 bytes or more is to be received, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is received. Write 0 before starting reception of the next data group.

FTC Flag (Frame/Wake-up Transmission Complete Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FTC flag is set to 1 upon completion of frame or wake-up transmission. Here, an interrupt request for RLN3n transmission is generated if the FTCIE bit in the RLN3nLIE register is 1 (interrupt is enabled). To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.
When response data of 9 bytes or more is to be transmitted, this bit is set to 1 each time a data group (variable from 0 to 8 bytes) is transmitted. Write 0 before starting transmission of the next data group.

20.3.2.15 RLN3nLEST — LIN Error Status Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RPER	—	CSER	—	FER	FTER	PBER	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Table 20.26 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	RPER	Response Preparation Error Flag 0: Response preparation error has not been detected. 1: Response preparation error has been detected.
6	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
5	CSER	Checksum Error Flag 0: Checksum error has not been detected. 1: checksum error has been detected.
4	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	FTER	Timeout Error Flag 0: Frame/response timeout error has not been detected. 1: Frame/response timeout error has been detected.
1	PBER	Physical Bus Error Flag 0: Physical bus error has not been detected. 1: Physical bus error has been detected.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H upon transition to LIN reset mode and start of the next communication (when the FTS bit of the RLN3nLTRC register is 1).

In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains 00_H.

When the FTS bit in the RLN3nLTRC register is 1 (frame transmission or wake-up transmission/reception is started), do not write a value to this register.

To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

RPER Flag (Response Preparation Error Flag)

Only 0 can be written to the RPER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The RPER flag is set to 1 upon response preparation error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

CSER Flag (Checksum Error Flag)

Only 0 can be written to the CSER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The CSER flag is set to 1 upon checksum error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

FER Flag (Framing Error Flag)

Only 0 can be written to the FER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the value of the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled), the FER flag is set to 1 upon framing error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

FTER Flag (Timeout Error Flag)

Only 0 can be written to the FTER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the FTERE bit of the RLN3nLEDE register is 1 (frame/response timeout error detection enabled), the FTER flag is set to 1 upon frame timeout error or response timeout error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode.

PBER Flag (Physical Bus Error Flag)

Only 0 can be written to the PBER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the PBERE bit of the RLN3nLEDE register is 1 (physical bus error detection enabled), the PBER flag is set to 1 upon physical bus error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled), the BER flag is set to 1 upon bit error detection. To clear the bit to 0 before the next communication (when the FTS bit of the RLN3nLTRC register is 1), write 0 to the bit in LIN operation mode or LIN wake-up mode.

20.3.2.16 RLN3nLDFC — LIN Data Field Configuration Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LSS	FSM	CSM	RFT	RFDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.27 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7	LSS	Transmission/Reception Continuation Select 0: The data group to be transmitted/received next is the last one. 1: The data group to be transmitted/received next is not the last one. (Checksum is not included.)
6	FSM	Frame Separate Mode Select 0: Frame separate mode is not set. 1: Frame separate mode is set.
5	CSM	Checksum Select 0: Classic checksum mode 1: Enhanced checksum mode
4	RFT	Response Field Communication Direction Select 0: Reception 1: Transmission
3 to 0	RFDL[3:0]	Response Field Length Select b3 b0 0 0 0 0: 0 byte (+ checksum) 0 0 0 1: 1 byte (+ checksum) 0 0 1 0: 2 bytes (+ checksum) : 0 1 1 1: 7 bytes (+ checksum) 1 0 0 0: 8 bytes (+ checksum) Settings other than the above are prohibited.

LSS Bit (Transmission/Reception Continuation Select)

The LSS bit indicates that the data group to be transmitted or received next is not the last data group when response data of 9 bytes or more is to be transmitted or received.
With 0 set, data and checksum are transmitted or received because the next data group to be transmitted or received is the last one.
With 1 set, only data is transmitted or received, and the checksum is not included because the next data group to be transmitted or received is not the last one.
Set the LSS bit only when the FSM bit is 1 (frame separate mode) and response data of 9 bytes or more is to be transmitted or received.

Set the LSS bit only when the RTS bit in the RLN3nLTRC is 0 (response transmit/receive is stopped).

FSM Bit (Frame Separate Mode Select)

The FSM bit sets the response communication mode.
With 0 set, frame separate mode is not selected. In this case, after header transmission is started (the FTS bit in the RLN3nLTRC register is 1), response is transmitted/received without the RTS bit in the RLN3nLTRC register being set.

With 1 set, frame separate mode is selected. If the RTS bit of the RLN3nLTRC register is set to 1 during header transmission, response transmission is executed after header transmission is completed.

For response reception which is 8 bytes or less (the RFT bit is 0), set the FSM bit to 0.
When causing a transition to LIN self-test mode, set this bit to 0 before transition.
For details on frame separate mode, see Section 20.7.3.1, Transmission of LIN Frames.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, set the FSM bit to 1.

CSM Bit (Checksum Select)

The CSM bit sets the checksum mode.
With 0 set, classic checksum mode is selected.
With 1 set, enhanced checksum mode is selected.
When the timeout error is used (the FTERE bit in the RLN3nLEDE register is 1), the specific timeout time depends on the setting of this bit. For details on the bit error, see Section 20.7.6, Error Status.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, do not change the CSM bit setting after the first data group through the last data group.
During communication of response data of 9 bytes or more, only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

RFT Bit (Response Field Communication Direction Select)

The RFT bits set the direction of the response field/wake-up signal communication.
With 0 set, reception is performed in the response field. In LIN wake-up mode, wake-up reception is performed (low level width of the input signal is counted).
With 1 set, transmission is performed in the response field. In LIN wake-up mode, wake-up transmission is performed.
Set this bit when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).
When response data of 9 bytes or more is to be transmitted or received, do not change the RFT bit setting after the first data group through the last data group.

RFDL[3:0] Bits (Response Field Length Select)

The RFDL[3:0] bits set the length of the response field data.
The data length can be 0 to 8 bytes excluding the checksum size.
To transmit response data with the FSM bit set to 0 (not frame separate mode), set the RFDL[3:0] bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).
To transmit response data with the FSM bit set to 1 (frame separate mode), set the RFDL[3:0] bits before response transmission (the RTS bit in the RLN3nLTRC register is 0).
To receive response data, set the RFDL[3:0] bits before header transmission (the FTS bit in the RLN3nLTRC register is 0).
When response data of 9 bytes or more is to be transmitted or received, set the RFDL[3:0] bits before data group transmission/reception (RTS bit in the RLN3nLTRC register is 0).
Only the last data group (the LSS bit is 0) includes the checksum, and no other groups (the LSS bit is 1) include the checksum.

20.3.2.17 RLN3nLIDB — LIN ID Buffer Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	IDP1	IDP0	ID[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.28 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7	IDP1	Parity Setting (P1) Sets the parity bits (P1) to be transmitted in the ID field.
6	IDP0	Parity Setting (P0) Sets the parity bits (P0) to be transmitted in the ID field.
5 to 0	ID[5:0]	ID Setting Sets the 6-bit ID value to be transmitted in the ID field.

Set the RLN3nLIDB register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

In LIN self-test mode, this register operates as described below.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see Section 20.9, LIN Self-Test Mode.

IDP[1:0] Bits (Parity Setting)

The IDP bits set the parity bits (P0 and P1) to be transmitted in the ID field of the LIN frame. IDP0 is for P0, and IDP1 is for P1.

Since parity is not automatically calculated, set the calculation result. Note that if the erroneous result is set, it is transmitted as is.

ID[5:0] Bits (ID Setting)

The ID bit sets the 6-bit ID value to be transmitted in the ID field of the LIN frame.

20.3.2.18 RLN3nLCBR — LIN Checksum Buffer Register

Access: This register can only be read in 8-bit units. In LIN self-test mode, this register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 16_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	CKSM[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.29 RLN3nLCBR Register Contents

Bit Position	Bit Name	Function
7 to 0	CKSM[7:0]	Holds the checksum value transmitted or received.

In LIN mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
  The value transmitted can be read from the register. Read the value after transmission is completed.
  Writing to this register is invalid.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
  The value received can be read from the register. Read the value after reception is completed.
  Writing to this register is invalid.

When response data of 9 bytes or more is to be transmitted or received, the checksum is appended only to the last data group; this register is not updated for the other data groups.

In LIN self-test mode, this register operates as follows:

- When the RFT bit in the RLN3nLDFC register is 1 (transmission):
  After completion of the frame transmission (after loopback), the reversed value of the received value can be read.
- When the RFT bit in the RLN3nLDFC register is 0 (reception):
  Write the value to be received before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see Section 20.9, LIN Self-Test Mode.

Set the RLN3nLCBR register when the FTS bit in the RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted).

20.3.2.19 RLN3nLDBRb — LIN Data Buffer b Register (b = 1 to 8)

Access: This register can be read/written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> + 18_H
RLN3nLDBR2: <RLIN3n_base> + 19_H
RLN3nLDBR3: <RLIN3n_base> + $1A_H$
RLN3nLDBR4: <RLIN3n_base> + $1B_H$
RLN3nLDBR5: <RLIN3n_base> + $1C_H$
RLN3nLDBR6: <RLIN3n_base> + $1D_H$
RLN3nLDBR7: <RLIN3n_base> + $1E_H$
RLN3nLDBR8: <RLIN3n_base> + $1F_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.30 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted or holds the received data. Setting range: 00_H to FF_H

- For response transmission:
  The LDBRn registers set the data to be transmitted in the response field.
  Use these registers with the following settings.
  - RFT in RLN3nLDFC register is 1 (transmission)
  - FSM in RLN3nLDFC register is 0 (not frame separate mode)
  - FTS bit in RLN3nLTRC register is 0 (frame transmission or wake-up transmission/reception is halted)

  or
  - RFT in RLN3nLDFC register is 1 (transmission)
  - FSM in RLN3nLDFC register is 1 (frame separate mode)
  - RTS in RLN3nLTRC register is 0 (response transmission/reception is halted)
- For response reception:
  The LDBRn registers hold the data received in the response field.
  The received data is overwritten. If an error is detected, the data up to the byte in which the error was detected are stored in the register.
  Do not read these registers when the FTS bit is 1 (frame transmission or wake-up transmission/reception is started)
- For transmission of response data of 9 bytes or more:
  Use the LDBRn registers with the following settings.
  - RFT in RLN3nLDFC register is 1 (transmission)
  - FSM in RLN3nLDFC register is 1 (frame separate mode)
  - RTS in RLN3nLTRC register is 0 (response transmission/reception is halted)
- For reception of response data of 9 bytes or more:
  Do not read these registers when the RTS bit is 1 (response transmission/reception is started).

In LIN self-test mode, these registers operate as described below.

Write the value to be transmitted before communication. After completion of frame transmission/reception (after loopback), the reversed value of the received value can be read.

For details about the LIN self-test mode, see Section 20.9, LIN Self-Test Mode.

20.3.3 UART Related Registers

20.3.3.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 01_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	NSPB[3:0]				LPRS[2:0]			—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 20.31 RLN3nLWBR Register Contents

Bit Position	Bit Name	Function
7 to 4	NSPB[3:0]	Bit Sampling Count Select b7 b4 0 0 0 0: 16 sampling 0 1 0 1: 6 sampling 0 1 1 0: 7 sampling 0 1 1 1: 8 sampling 1 0 0 0: 9 sampling 1 0 0 1: 10 sampling 1 0 1 0: 11 sampling 1 0 1 1: 12 sampling 1 1 0 0: 13 sampling 1 1 0 1: 14 sampling 1 1 1 0: 15 sampling 1 1 1 1: 16 sampling Settings other than the above are prohibited.
3 to 1	LPRS[2:0]	Prescaler Clock Select b3 b1 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128
0	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.

Set the LN3nLWBR register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

NSPB[3:0] Bits (Bit Sampling Count Select)

The NSPB[3:0] bits select the number of sampling in one Tbit (reciprocal of the bit rate).
In UART mode, it is possible to set the NSPB[3:0] bits from 6 sampling to 16 sampling.

LPRS[2:0] Bits (Prescaler Clock Select)

The LPRS[2:0] bits select the frequency division ratio for the prescaler.

The LIN communication clock source is divided by this prescaler.

20.3.3.2 RLN3nLBRP01 — UART Baud Rate Prescaler 01 Register

Access: RLN3nLBRP01 register can be read/written in 16-bit units.
RLN3nLBRP0 register can be read/written in 8-bit units.
RLN3nLBRP1 register can be read/written in 8-bit units.

Address: RLN3nLBRP01: <RLIN3n_base> + 02_H
RLN3nLBRP0: <RLIN3n_base> + 02_H
RLN3nLBRP1: <RLIN3n_base> + 03_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BRP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.32 RLN3nLBRP01 Register Contents

Bit Position	Bit Name	Function
15 to 0	BRP[15:0]	Assuming that the value set in this register is L (0 to 65535), the baud rate prescaler divides the frequency of the prescaler clock by L + 1. Setting range: 0000_H to $FFFF_H$

Set the RLN3nLBRP01 register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

Assuming that the value set in this register is L, the baud rate prescaler divides the frequency of the clock that is selected by the LPRS[2:0] bits (prescaler clock select bits) in the RLN3nLWBR register by
L + 1.

The RLN3nLBRP01 register can be accessed in 8-bit units by the registers RLN3nLBRP0 and RLN3nLBRP1.

20.3.3.3 RLN3nLMD — UART Mode Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	LRDNFS	—	—	—	LMD[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R	R/W	R/W

Table 20.33 RLN3nLMD Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	LRDNFS	LIN Reception Data Noise Filtering Disable 0: The noise filter is enabled. 1: The noise filter is disabled.
4 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	LMD[1:0]	LIN/UART Mode Select b1 b0 0 1: UART mode

Set the RLN3nLMD register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

LRDNFS Bit (UART Reception Data Noise Filtering Disable)

The LRDNFS bit enables or disables the noise filter when receiving data.
With 0 set, the noise filter is enabled when receiving data.
With 1 set, the noise filter is disabled when receiving data.

LMD[1:0] Bits (LIN/UART Mode Select)

The LMD[1:0] bits select the LIN/UART interface mode.
To use the LIN/UART interface as an UART, set these bits to 01_B.

20.3.3.4 RLN3nLBFC — UART Configuration Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 09_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	UTPS	URPS	UPS[1:0]		USBLS	UBOS	UBLS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.34 RLN3nLBFC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6	UTPS	UART Output Polarity Switch 0: Transmit data normal output 1: Transmit data with inversion output
5	URPS	UART Input Polarity Switch 0: Reception data normal output 1: Reception data with inversion output
4, 3	UPS[1:0]	UART Parity Select 00: Parity prohibited 01: Even Parity 10: 0 Parity 11: Odd parity
2	USBLS	UART Stop Bit length Select 0: Stop bit:1 bit 1: Stop bit: 2 bits
1	UBOS	UART Transfer Format Order Select 0: LSB First 1: MSB First
0	UBLS	UART Character Length Select 0: UART 8 bits communication 1: UART 7 bits communication

Set the RLN3nLBFC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

UTPS Bit (UART Output Polarity Switch)

Sets the output polarity for UART communication.
With 0 set, transmit data is output without inversion.
With 1 set, inverted transmit data is output.
The setting of this bit is valid in all the bits of the UART frame.
In half-duplex communication, this setting should match with the setting of URPS bit.

URPS Bit (UART Input Polarity Switch)

Sets the input polarity for UART communication.
With 0 set, receive data is input without inversion.
With 1 set, receive data is input with inversion.
The setting of this bit is valid in all the bits of the UART frame.
In half-duplex communication, this setting should match with the setting of UTPS bit.

When setting this bit to “1” and expansion bit reception (with expansion bit comparison) or (with data comparison) is performed, set the reversed value of the expected value to the UEBDL bit in the RLN3nLUOR1 register and RLN3nLIDB register for comparison of the reversed values of the received values.

UPS[1:0] Bits (UART Parity Select)

Sets the UART parity.

- When these bits are set to “00_B”, data is communicated without the parity.

  [Transmission]

  A parity bit is not added to transmit data.

  [Reception]

  Data is received without parity processing. Therefore, a parity error does not occur.

- When these bits are set to “01”, data is communicated with the even parity.

  [Transmission]

  If the number of 1s in transmit data is odd, “1” is added to the parity bit. If the number of 1s in transmit data is even, “0” is added to the parity bit.

  [Reception]

  If the number of 1s in receive data including the parity bit is odd, a parity error occurs.

- When these bits are set to “10”, data is communicated with 0 parity.

  [Transmission]

  Regardless of the number 1s in transmit data, “0” is added to the parity bit.

  [Reception]

  The value of the parity bit is not judged. Therefore, no parity error occurs.

- When these bits are set to “11”, data is communicated with the odd parity.

  [Transmission]

  If the number of 1s in transmit data is odd, “0” is added to the parity bit. If the number of 1s in transmit data is even, “1” is added to the parity bit.

  [Reception]

  If the number of 1s in receive data including the parity bit is even, a parity error occurs.

USBLS Bit (UART Stop Bit Length Select)

Sets the stop bit length of data for UART communication.
With 0 set, stop bit length of 1 bit is selected.
With 1 set, stop bit length of 2 bits is selected.

UBOS Bit (UART Transfer Format Select)

Sets the bit order of data for UART communication.
With 0 set, LSB first is selected.
With 1 set, MSB first is selected.

UBLS Bit (UART Character Length Select)

Sets the character length of one frame for UART communication.
With 0 set, the character length is 8 bits.
With 1 set, the character length is 7 bits.
When the character length of one frame is 9 bits (the UEBE bit in the RLN3nLUOR1 register is 1), the setting of this bit is ignored.

20.3.3.5 RLN3nLSC — UART Space Configuration Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + $0A_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	IBS[1:0]		—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R

Table 20.35 RLN3nLSC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5, 4	IBS[1:0]	Inter-Byte Space Select b5 b4 0 0: 0 Tbit 0 1: 1 Tbit 1 0: 2 Tbits 1 1: 3 Tbits
3 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

Set the RLN3nLSC register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

IBS[1:0] Bits (Inter-Byte Space Select)

The IBS[1:0] bits set the width of the space between the UART frame in UART buffer transmit 0 Tbit to 3 Tbits can be set.

Set IBS[1:0] bits to "00_B" when UART buffer is not used.

When data is transferred from the UART transmit data register (RLN3nLUTDR) and the UART wait transmit data register (RLN3nLUWTDR), the setting of these bits is ignored.Set these bits to "00_B".

20.3.3.6 RLN3nLEDE —UART Error Detection Enable Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + $0D_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FERE	OERE	—	BERE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R	R/W

Table 20.36 RLN3nLEDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	FERE	Framing Error Detection Enable 0: Disables framing error detection. 1: Enables framing error detection.
2	OERE	Overrun Error Detection Enable 0: Disables overrun error detection. 1: Enables overrun error detection.
1	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
0	BERE	Bit Error Detection Enable 0: Disables bit error detection. 1: Enables bit error detection.

Set the RLN3nLEDE register when the OMM0 bit in the RLN3nLMST register is 0_B (in LIN reset mode).

FERE Bit (Framing Error Detection Enable)

The FERE bit enables or disables detection of the framing error.
With 0 set, the framing error is not detected.
With 1 set, the framing error is detected.
When this bit is set to 1, the detection result is indicated in the FER flag in the RLN3nLEST register.
For details on the framing error, see Section 20.7.6, Error Status.

OERE Bit (Overrun Error Detection Enable)

This bit enables or disables detection of the overrun error.
With 0 set, the overrun error is not detected.
With 1 set, the overrun error is detected.
When this bit is set to 1, the detection result is reflected in the OER flag in the RLN3nLEST register.
For details on the overrun error, see Section 20.7.6, Error Status.

BERE Bit (Bit Error Detection Enable)

The BERE bit enables or disables detection of the bit error.
With 0 set, the bit error is not detected.
With 1 set, the bit error is detected.
When this bit is set to 1, the detection result is indicated in the BER flag in the RLN3nLEST register.
In full-duplex communication, do not set this bit to “1”.

Do not set this register when the NSPB[3:0] bits in the RLN3nLWBR register is 0101_B (6 sampling) and the LRDNFS bit in the RLN3nLMD register is 0 (noise filter is enabled).

For details on the bit error, see Section 20.7.6, Error Status.

20.3.3.7 RLN3nLCUC — UART Control Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + $0E_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 20.37 RLN3nLCUC Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	OM0	LIN Reset 0: LIN reset mode is caused. 1: LIN reset mode is canceled.

After a value is written to this register, confirm that the value written is actually indicated in the RLN3nLMST register before writing another value.

OM0 Bit (LIN Reset)

The OM0 bit selects whether to transition to or exit LIN reset mode.
With 0 set, reset mode.
With 1 set, reset mode is canceled.

20.3.3.8 RLN3nLTRC — UART Transmission Control Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RTS	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R

Table 20.38 RLN3nLTRC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	RTS	UART Buffer Transmission Start 0: UART Buffer transmission is stopped. 1: UART Buffer transmission is started.
0	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.

RTS Bit (UART Buffer Transmission Start)

When transmitting data from the UART buffer, set this bit to “1”.
Only 1 can be written to this bit; 0 cannot be written.
Write to this bit when the UTOE bit in the RLN3nLUOER register is 1 (transmission enable) and the UTS bit in the RLN3nLST register is 0 (transmission is not in progress).
Once set, regardless of errors, this bit is automatically cleared to 0 upon completion of the number of data transmission specified by the MDL bit in the RLN3nLDFC register. Moreover, this bit is automatically cleared to 0 upon transition to reset mode.
Writing a value to this bit is disabled when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).
When writing 1 to this bit while the UTSW bit in the RLN3nLDFC register is 1 (when UART buffer transmission is requested, the start of transmission is delayed until the stop bit of reception data is completed), write only during the reception of stop bit.

20.3.3.9 RLN3nLMST — UART Mode Status Register

Access: This register can only be read in 8-bit units

Address: <RLIN3n_base> + 11_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OMM0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 20.39 RLN3nLMST Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	OMM0	LIN Reset Status Monitor 0: The module is in LIN reset mode. 1: The module is not in LIN reset mode.

OMM0 Bit (LIN Reset Status Monitor)

The OMM0 bit indicates the current operating mode.

20.3.3.10 RLN3nLST — UART Status Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 12_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	URS	UTS	ERR	—	—	FTC
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 20.40 RLN3nLST Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	URS	Data Reception Status 0:Reception is not operated. 1:Reception is operated.
4	UTS	Transmission Status 0: Transmission is not operated. 1: Transmission is operated.
3	ERR	Error Detection Flag 0: No error has been detected. 1: Error has been detected.
2, 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	FTC	Successful UART Buffer Transmission Flag 0: UART buffer transmission has not been completed. 1: UART buffer transmission has been completed.

The RLN3nLST register is automatically cleared to "00_H" upon transition to LIN reset mode. In LIN reset mode, this register cannot be written to. In LIN reset mode, the register retains "00_H". To clear the specific bits in the register, write 0 to the bits to be cleared and write 1 to the other bits using the store instruction.

URS Bit (Data Reception Status)

At the start of the reception, this flag is set to 1.
The reception is started under the following condition.

- When the start bit is detected

At the end of reception, this flag is cleared to 0. While reception is stopped, this flag retains 0.
The reception is ended under the following conditions.

- Sampling point of the first bit of the stop bits

UTS Bit (Transmission Status Flag)

At the start of the transmission, this flag is set to 1. During the transmission, this flag retains 1.
The transmission is started under the following conditions.

- When transmission data is set to the RLN3nLUTDR or RLN3nLUWTDR register
- When the RTS bit in the RLN3nLTRC register is set to 1

This flag is cleared to 0 at the end of transmission.
The transmission is ended under the following conditions.

- When transmission of data set in the RLN3nLUTDR or RLN3nLUWTDR register is completed and next data is not set
- When transmission from UART buffer is completed (when the RTS bit in the RLN3nLTRC register is cleared to 0)

ERR Flag (Error Detection Flag)

This flag is set to 1 upon detection of an error, detection of an expansion bit, or upon ID matching (when the value of at least one of the flags of the RLN3nLEST register is 1). At this time, an interrupt request for RLIN3n status is generated. However, when this bit is 1, an interrupt is not generated upon detection of an error, detection of an expansion bit, or upon ID matching. To clear the bit to 0, write 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register.

FTC Flag (Successful Frame/Wake-up Transmission Flag)

Only 0 can be written to the FTC flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
Regardless of errors, this bit is set to 1 upon completion of the number of data transmission specified by the MDL bit in the RLN3nLDFC register from the UART buffer. At this time, an interrupt request for RLIN3n transmission is generated.
Write 0 to the bit to be cleared.

20.3.3.11 RLN3nLEST — UART Error Status Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 13_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	UPER	IDMT	EXBT	FER	OER	—	BER
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W

Table 20.41 RLN3nLEST Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6	UPER	Parity Error Flag 0: Parity error has not been detected. 1: Parity error has been detected.
5	IDMT	ID Matching Flag 0: The receive data does not match with the ID value. 1: The receive data matches with the ID value.
4	EXBT	Expanded Bit Detection Flag 0: Expanded bit has not been detected. 1: Expanded bit has been detected.
3	FER	Framing Error Flag 0: Framing error has not been detected. 1: Framing error has been detected.
2	OER	Overrun Error Flag 0: Overrun error has not been detected. 1: Overrun error has been detected.
1	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
0	BER	Bit Error Flag 0: Bit error has not been detected. 1: Bit error has been detected.

The RLN3nLEST register is automatically cleared to 00_H when the module transitions to LIN reset mode. In LIN reset mode, this register cannot be written to, and the value of 00_H is held. To clear certain bits in this register, write 0 to those bits, and write 1 to the bits not to be cleared by using the store instruction.

UPER Flag (Parity Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
This flag is set to 1 upon parity error detection. To clear the bit, write 0 to the bit.

IDMT Flag (ID Matching Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that has been retained before 1 is written.

The IDMT flag becomes 1 when all the following conditions are met:

- The UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enabled)

- The UECD bit in the RLN3nLUOR1 register is 0 (expansion bit comparison enabled)
- The UEBDCE bit in the RLN3nLUOR1 register is 1 (expansion bit/data comparison enabled)
  - The received expansion bit and the value of the UEBDL bit of the RLN3nLUOR1 register are matched.
  - The 8-bit receive data excluding the expansion bit and the value of the RLN3nLIDB register are matched.

To clear the bit, write 0 to the bit.

EXBT Flag (Expanded Bit Detection Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
When the UEBE bit in the RLN3nLUOR1 register is 1 (expansion bit enable), if the received expansion bit matches with the UEBDL bit in the RLN3nLUOR1 register, this flag is set to 1.
To clear the bit, write 0 to the bit.

FER Flag (Framing Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The FER flag is set to 1 upon framing error detection while the FERE bit of the RLN3nLEDE register is 1 (framing error detection enabled). To clear the bit, write 0 to the bit.

OER Flag (Overrun Error Flag)

Only 0 can be written to this flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The 0ER flag is set to 1 upon overrun error detection while the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). To clear the bit, write 0 to the bit.

BER Flag (Bit Error Flag)

Only 0 can be written to the BER flag; when 1 is written, the bit retains the value that has been retained before 1 is written.
The BER flag is set to 1 when the transmitted data and the data monitored by the receive pin do not match while the BERE bit of the RLN3nLEDE register is 1 (bit error detection enabled).
To clear the bit, write 0 to the bit.

20.3.3.12 RLN3nLDFC — UART Data Field Configuration Register

Access: This register can be read/written in 8-bit units.

Address: <RLIN3n_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	UTSW	—	MDL[3:0]			
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 20.42 RLN3nLDFC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	UTSW	Transmission Start Wait 0: When UART transmission is requested, transmission is started immediately. 1: When UART transmission is requested, transmission is not started until reception of the stop bit is completed.
4	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
3 to 0	MDL[3:0]	UART Buffer Data Length Select b3 b0 0 0 0 0: 9 data 0 0 0 1: 1 data 0 0 1 0: 2 data 0 0 1 1: 3 data 0 1 0 0: 4 data 0 1 0 1: 5 data 0 1 1 0: 6 data 0 1 1 1: 7 data 1 0 0 0: 8 data 1 0 0 1: 9 data Settings other than the above are prohibited.

UTSW Bit (Transmission Start Wait)

This bit controls the transmission start timing of UART buffer.
With 0 set, transmission is started as soon as the start of UART buffer transmit is requested.
With 1 set, transmission is started after the completion of the stop bit reception.
Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.
This bit is enabled when the RTS bit in the RLN3nLTRC register is set to 1. In addition, writing a value to this bit is disabled when the RTS bit is 1 (UART buffer transmission started).
Set this bit to 1 only to switch from reception to transmission in half-duplex communication.

MDL[3:0] Bits (UART Buffer Data Length Select)

This bit specifies the data length of the UART buffer.
Writing a value to these bits is disabled when the RTS bit in the RLN3nLTRC register is 1 (UART buffer transmission started).

20.3.3.13 RLN3nLIDB — UART ID Buffer Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 15_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	ID[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.43 RLN3nLIDB Register Contents

Bit Position	Bit Name	Function
7 to 0	ID[7:0]	ID value that is referred for the expansion bit data comparison is set

ID[7:0] Bit (ID Bit)

When the UEBE bit in the RLN3nLUOR1 register is set to 1 (expansion bit enabled), the UECD bit is set to 0 (expansion bit comparison enabled), and the UEBDCE bit is set to 1 (expansion bit/data comparison enabled), set the receive data and the value to be compared. Write to the RLN3nLIDB register when the URS bit in the RLN3nLST register is 0 (receive operation is not in progress).

20.3.3.14 RLN3nLUDB0 — UART Data 0 Buffer Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 17_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	UDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.44 RLN3nLUDB0 Register Contents

Bit Position	Bit Name	Function
7 to 0	UDB[7:0]	Transmission data is set Setting range: 00_H to FF_H.

If the data length selection corresponds to 9 data bytes (RLN3nLDFC.MDL bit is "0_H" or "9_H") for multi-byte UART transmission, then the first data value for UART communication is present in this buffer.

Write to the RLN3nLUDB0 register when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

Table 20.45, Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format shows the bit arrangement according to the set communication format.

For details about the UART buffer, see Section 20.8.1.2, UART Buffer Transmission, (1) UART Buffer Transmission.

Table 20.45 Bit Arrangement of the RLN3nLUDB0 Register According to Each Communication Format

	RLN3nLUDB0							
	b7	b6	b5	b4	b3	b2	b1	b0
7-bit LSB first	–	b6	b5	b4	b3	b2	b1	b0
7-bit MSB first	–	b0	b1	b2	b3	b4	b5	b6
8-bit LSB first	b7	b6	b5	b4	b3	b2	b1	b0
8-bit MSB first	b0	b1	b2	b3	b4	b5	b6	b7

20.3.3.15 RLN3nLDBRb — UART Data Buffer b Register (b = 1 to 8)

Access: This register can be read/written in 8-bit units.

Address: RLN3nLDBR1: <RLIN3n_base> +18_H
RLN3nLDBR2: <RLIN3n_base> + 19_H
RLN3nLDBR3: <RLIN3n_base> + $1A_H$
RLN3nLDBR4: <RLIN3n_base> + $1B_H$
RLN3nLDBR5: <RLIN3n_base> + $1C_H$
RLN3nLDBR6: <RLIN3n_base> + $1D_H$
RLN3nLDBR7: <RLIN3n_base> + $1E_H$
RLN3nLDBR8: <RLIN3n_base> + $1F_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	LDB[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.46 RLN3nLDBRb Register Contents

Bit Position	Bit Name	Function
7 to 0	LDB[7:0]	Sets the data to be transmitted. Setting range: 00_H to FF_H.

This register specifies the data transmitted from the UART buffer.

Write to these registers when the RTS bit of the RLN3nLTRC register is 0 (UART buffer transmission stopped).

Table 20.47, Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format shows the bit arrangement according to the set communication format.

For details about the UART buffer, see Section 20.8.1.2, UART Buffer Transmission, (1) UART Buffer Transmission.

Table 20.47 Bit Arrangement of the RLN3nLDBRb Register According to Each Communication Format

	RLN3nLUDBb							
	b7	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
7-bit LSB first	–	b6	b5	b4	b3	b2	b1	b0
7-bit MSB first	–	b0	b1	b2	b3	b4	b5	b6
8-bit LSB first	b7	b6	b5	b4	b3	b2	b1	b0
8-bit MSB first	b0	b1	b2	b3	b4	b5	b6	b7

20.3.3.16 RLN3nLUOER — UART Operation Enable Register

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	UROE	UTOE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 20.48 RLN3nLUOER Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	UROE	Reception Enable 0: Disables reception. 1: Enables reception.
0	UTOE	Transmission Enable 0: Disables transmission. 1: Enables transmission.

The RLN3nLUOER register is automatically cleared to 00_H upon transition to LIN reset mode.
In LIN reset mode, this register cannot be written to.
In LIN reset mode, the register retains 00_H.

UROE Bit (Reception Enable)

The UROE bit enables or disables reception.
With 0 set, reception is disabled.
With 1 set, reception is enabled.
Do not clear this bit during reception. If the communication is suspended during reception, set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) to shift to the LIN reset mode. However, the transmit operation is also suspended at this time.
Do not set this bit to 1 when data transmission from the UART buffer is in progress.

UTOE Bit (Transmission Enable)

The UTOE bit enables or disables transmission.
With 0 set, transmission is disabled.
With 1 set, transmission is enabled.
Do not clear this bit during transmission. If the communication is suspended during transmission, set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) to shift to the LIN reset mode. However, the receive operation is also suspended at this time.

20.3.3.17 RLN3nLUOR1 — UART Option Register 1

Access: This register can be read/written in 8-bit units

Address: <RLIN3n_base> + 21_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	UECD	UTIGTS	UEBDCE	UEBDL	UEBE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 20.49 RLN3nLUOR1 Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
4	UECD	Expansion Bit Comparison Disable 0: Enables comparison between the received expansion bit and the UEBDL bit value. 1: Disables comparison between the received expansion bit and the UEBDL bit value.
3	UTIGTS	Transmission Interrupt Generation Timing Select 0: Transmission interrupt is generated at the start of transmission. 1: Transmission interrupt is generated at the completion of transmission.
2	UEBDCE	Expansion Bit Data Comparison Enable 0: Disables data comparison after an expansion bit is detected. 1: Enables data comparison after an expansion bit is detected.
1	UEBDL	Expansion Bit Detection Level Select 0: Selects expansion bit value 0 as the expansion bit detection level. 1: Selects expansion bit value 1 as the expansion bit detection level.
0	UEBE	Expansion Bit Enable 0: Disables expansion bit operation. 1: Enables expansion bit operation.

UECD Bit (Expansion Bit Comparison Disable)

The UECD bit enables or disables comparison between the received expansion bit and the UEBDL bit value when the UEBE bit is 1 (expansion bit operation is enabled).
With 0 set, comparison between the received expansion bit and the UEBDL bit value is enabled when the expansion bit is received.
With 1 set, comparison between the received expansion bit and the UEBDL bit value is disabled when the expansion bit is received.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).
Do not set this bit to 1 when the UART buffer is used.
Do not set this bit to 1 when the UEBDCE bit is 1 (expansion bit/data comparison enable).

UTIGTS Bit (Transmission Interrupt Generation Timing Select)

The UTIGTS bit sets the generation timing of the transmission interrupt.
With 0 set, the transmission interrupt is generated at the start of transmission.
With 1 set, the transmission interrupt is generated at the completion of transmission.
When transmission from the UART buffer is performed with 0 set, the transmission interrupt is generated only at the start of the transmission of the last data (the data length is specified by the MDL bits in the RLN3nLDFC register).

When transmission from the UART buffer is performed with 1 set, the transmission interrupt is generated only at the completion of the transmission of the last data of the data length set with the MDL bits in the RLN3nLDFC register.

UEBDCE Bit (Expansion Bit Data Comparison Enable)

After an expansion bit is detected, this bit enables or disables the comparison between the 8-bit receive data excluding the expansion bit and the value of the RLN3nLIDB register.
With 0 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the receive value in the RLN3nLURDR register and the value of the RLN3nLIDB register is disabled.
With 1 set, when the level selected by the UEBDL bit is detected as an expansion bit, the comparison between the receive value in the RLN3nLURDR register and the value of the RLN3nLIDB register is enabled.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).
Do not set this bit to 1 when the UEBE bit is 0 (expansion bit operation disabled).
Do not set this bit to 1 when the UECD bit is 1 (expansion bit comparison disabled).
Do not set this bit to 1 when the UART buffer is used.

UEBDL Bit (Expansion Bit Detection Level Select)

The UEBDL bit sets the level to be detected as the expansion bit when the UEBE bit is 1 (expansion bit operation is enabled) and the UECD bit is 0 (comparison of the expansion bit is enabled).
With 0 set, expansion bit value 0 is the level to be detected as the expansion bit.
With 1 set, expansion bit value 1 is the level to be detected as the expansion bit.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).
Do not set this bit to 1 when the UART buffer is used.

UEBE Bit (Expansion Bit Enable Bit)

The UEBE bit enables or disables expansion bit operation.
With 0 set, expansion bit operation is disabled.
With 1 set, expansion bit operation is enabled.
Set this bit when the OMM0 bit of the RLN3nLMST register is 0_B (in LIN reset mode).
Do not set this bit to 1 when the UART buffer is used.

20.3.3.18 RLN3nLUTDR — UART Transmission Data Register

Access: RLN3nLUTDR register can be read/written in 16-bit units.
RLN3nLUTDRL register can be read/written in 8-bit units.
RLN3nLUTDRH register can be read/written in 8-bit units.

Address: RLN3nLUTDR: <RLIN3n_base> + 24_H
RLN3nLUTDRL: <RLIN3n_base> + 24_H
RLN3nLUTDRH: <RLIN3n_base> + 25_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UTD[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.50 RLN3nLUTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
8 to 0	UTD[8:0]	Sets the data to be transmitted from the transmission buffer. Setting range: 000_H to $1FF_H$

The RLN3nLUTDR register sets the data to be transmitted from the transmit data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

This register can be accessed in 8 bits.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

Also, do not write data to this register when a transmission request is being generated due to write access to the RLN3nLUWTDR register.

When transmitting data continuously, do not set another piece of transmission data in this register before the generation of transmission interrupt.

The table below shows the bit arrangement according to the set communication format.

Table 20.51 Bit Arrangement of the RLN3nLUTDR Register According to Each Communication Format

	RLN3nLUTDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

20.3.3.19 RLN3nLURDR — UART Reception Data Register

Access: RLN3nLURDR register can only be read in 16-bit units.
RLN3nLURDRL register can only be read in 8-bit units.
RLN3nLURDRH register can only be read in 8-bit units.

Address: RLN3nLURDR: <RLIN3n_base> + 26_H
RLN3nLURDRL: <RLIN3n_base> + 26_H
RLN3nLURDRH: <RLIN3n_base> + 27_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	URD [8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20.52 RLN3nLURDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned.
8 to 0	URD [8:0]	Stores the received data Setting range: 000_H to $1FF_H$

The RLN3nLURDR allows the reception data to be read from the receive data register.

When the UROE bit in the RLN3nLUOER register is 1, the reception data is stored in this register and can be read out.

This register is updated at the stop bit of the reception data.

This register is also updated when an error is caused by the parity or stop bit.

However, the value of this register is not updated upon occurrence of an overrun error when the OERE bit of the RLN3nLEDE register is 1 (overrun detection enabled). The value of this register is updated upon occurrence of an overrun error when the OERE bit is 0 (overrun detection disabled).

Read this register upon occurrence of a receive error (overrun error, framing error, parity error) when the OERE bit of the RLN3nLEDE register is 1 (overrun error detection enabled). If the next data is received without reading this register, an overrun error occurs.

This register can be accessed in 8 bits.

However, during expansion bit use (UEBE bit of the RLN3nLUOR1 register is 1 (expansion bit operation enabled), do not attempt 8-bit access.

The table below shows the bit arrangement according to the set communication format.

Table 20.53 Bit Arrangement of the RLN3nLURDR Register According to Each Communication Format

	RLN3nLURDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

20.3.3.20 RLN3nLUWTDR — UART Wait Transmission Data Register

Access: RLN3nLUWTDR register can be read/written in 16-bit units.
RLN3nLUWTDRL register can be read/written in 8-bit units.
RLN3nLUWTDRH register can be read/written in 8-bit units.

Address: RLN3nLUWTDR: <RLIN3n_base> + 28_H
RLN3nLUWTDRL: <RLIN3n_base> + 28_H
RLN3nLUWTDRH: <RLIN3n_base> + 29_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	UWTD[8:0]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20.54 RLN3nLUWTDR Register Contents

Bit Position	Bit Name	Function
15 to 9	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
8 to 0	UWTD[8:0]	Sets the data to be transmitted from the wait transmit data register after waiting for the stop bit reception to be completed. Setting range: 000_H to $1FF_H$

The RLN3nLUWTDR register sets the data to be transmitted from the UART wait transmit data register.

Writing data to this register with the UTOE bit in the RLN3nLUOER register set to 1 starts transmission.

Use this register only to switch from reception to transmission in half-duplex communication.

The user should write to this register only while the stop bit is being received.

Note that the wait time is only 1 bit even if the stop bit length is set to 2 bits with the USBLS bit in the RLN3nLBFC register.

When this register is read, the RLN3nLUTDR register value is actually read.

In 9-bit communication mode, do not attempt 8-bit access.

Do not write data to this register when data transmission from the UART buffer is in progress.

The table below shows the bit arrangement according to the set communication format.

Table 20.55 Bit Arrangement of the RLN3nLUWTDR Register According to Each Communication Format

	RLN3nLUWTDR								
	b8	b7	b6	b5	b4	b3	b2	b1	b0
7-bit; LSB first	—	—	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
7-bit; MSB first	—	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6
8-bit; LSB first	—	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
8-bit; MSB first	—	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7
9-bit; LSB first	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
9-bit; MSB first	Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8

20.4 Interrupt Sources

The LIN/UART interface generates four types of interrupt requests.

- RLIN3n transmission interrupt
- RLIN3n successful reception interrupt
- RLIN3n status interrupt
- RLIN3n interrupt

Setting the LIOS bit in the RLN3nLMD register to 0 allows to perform logical OR operation on all of the interrupt sources, outputting the interrupt request from the RLIN3n interrupt.

Setting the LIOS bit in the RLN3nLMD register to 1 allows to output the RLIN3n transmission interrupt, RLIN3n successful reception interrupt, or RLIN3n status interrupt depending on the interrupt request.

Table 20.56 lists the sources for each interrupt.

Table 20.56 Interrupt Sources

	LIOS bit in RLN3nLMD register is 0	LIOS bit in RLN3nLMD register is 1*1		
	RLIN3n Interrupt	RLIN3n Transmission Interrupt	RLIN3n Successful Reception Interrupt	RLIN3n Status Interrupt
LIN master mode	• Successful frame transmission • Successful frame reception • Successful wake-up transmission • Successful wake-up reception • Successful header transmission • Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error	• Successful frame transmission • Successful wake-up transmission • Successful header transmission	• Successful wake-up reception • Successful frame reception	• Bit error • Physical bus error • Frame/response timeout error • Framing error • Checksum error • Response preparation error
UART mode	—	• Transmission start/successful transmission	• Successful reception • Expansion bit mismatch	• Bit error • Overrun error • Framing error • Expansion bit match • ID match • Parity error

Note 1. The LIOS bit setting is valid in LIN Mode. In UART mode, setting the LIOS bit is not required.

In LIN mode, each interrupt request is output when the corresponding bit in the RLN3nLIE register is 1 (interrupt is enabled) and the corresponding flag in the RLN3nLST register is 1.

20.5 Modes

The LIN/UART interface provides the following four modes, depending upon the specific function to be performed:

- LIN reset mode
- LIN master mode
- UART mode
- LIN self-test mode

Figure 20.2 shows mode transitions. **Table 20.60** describes mode transition conditions. **Table 20.60** lists operations available in each mode.

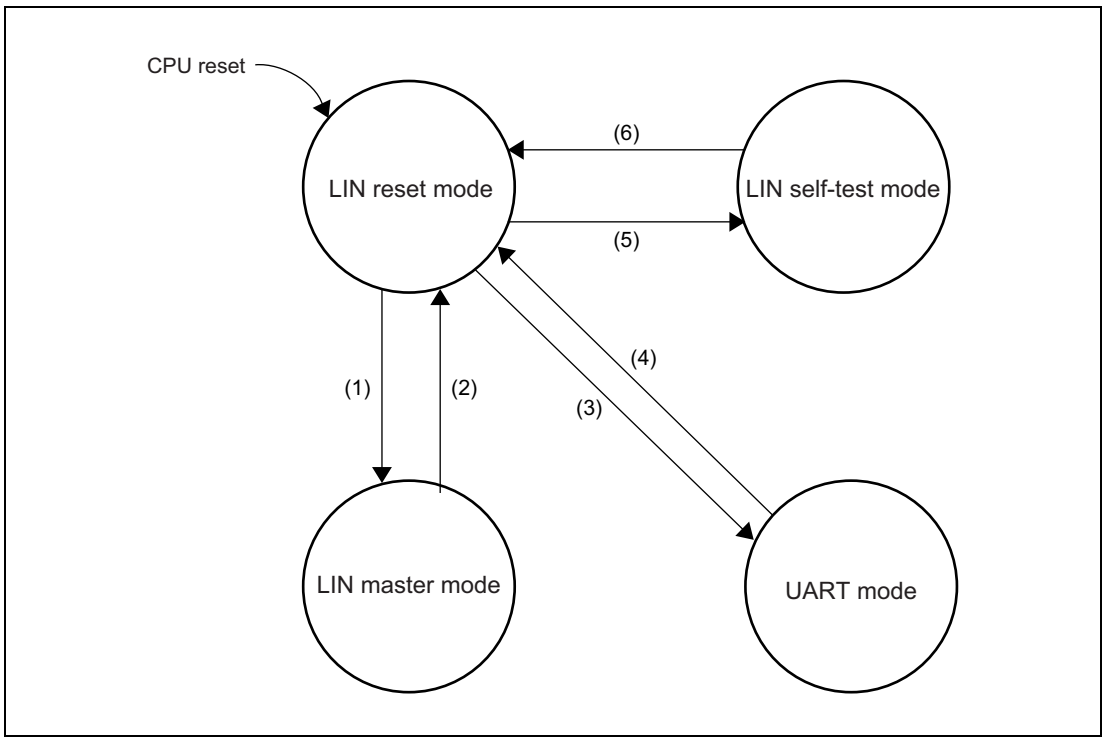


Figure 20.2 Mode Transitions

Table 20.57 Operations Available in Each Mode

LIN Master Mode	UART Mode	LIN Self-Test Mode
Header transmission Response transmission Response reception Wake-up transmission Wake-up reception Error detection	UART transmission UART reception Error detection	Self test

Whether a transition has been caused to LIN reset mode, the LIN mode, or the UART mode can be verified by reading the LMD[1:0] bits in the RLN3nLMD register or the OMM0 bit in the RLN3nLMST register.

For a description of the LIN self-test mode, see Section 20.9, LIN Self-Test Mode.

20.6 LIN Reset Mode

Setting the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode) causes a transition to LIN reset mode. The change to LIN reset mode can be verified by determining that the OMM0 bit in the RLN3nLMST register has been set to 0 (LIN reset mode). In this mode, the LIN communication and the UART communication functions are halted.

From LIN reset mode, transitions to LIN mode, UART mode, and LIN self-test mode can be made.

When the mode changes to LIN reset mode, the following registers are initialized to their reset values, and as long as LIN reset mode is in effect, they retain their initial values.

- RLN3nLTRC register
- RLN3nLST register
- RLN3nLEST register
- RLN3nLUOER register

The following registers retain their previous values even when a transition to LIN reset mode is made:

- RLN3nLWBR register
- RLN3nLBRP0 register
- RLN3nLBRP1 register
- RLN3nLMD register
- RLN3nLBFC register
- RLN3nLSC register
- RLN3nLWUP register
- RLN3nLIE register
- RLN3nLEDE register
- RLN3nLDFC register
- RLN3nLIDB register
- RLN3nLCBR register
- RLN3nLUDB0 register
- RLN3nLDBRb register (b = 1 to 8)
- RLN3nLUOR1 register
- RLN3nLUTDR register
- RLN3nLURDR register
- RLN3nLUWTDR register

20.7 LIN Mode

LIN mode can operate in LIN master mode.

In LIN master mode, the following operations can be performed: header transmission, response transmission, response reception, wake-up transmission, wake-up reception, and error detection. In LIN reset mode, setting the LMD[1:0] bits in the RLN3nLMD register to 00_B (LIN master mode) and the OM1 and OM0 bits in the RLN3nLCUC register to either 01_B or 11_B sets LIN master mode, turning the OMM1 and OMM0 bits in the RLN3nLMST register to either 01_B to 11_B.

When changing a submode to another submode within LIN mode, a transition to LIN reset mode should first be made and change the LMD[1:0] bits in the RLN3nLMD register.

The LIN mode provides the following two operation modes:

- LIN operation mode
- LIN wake-up mode

Figure 20.3 shows the transition of operation modes. **Table 20.58** describes the transition conditions of operation modes.

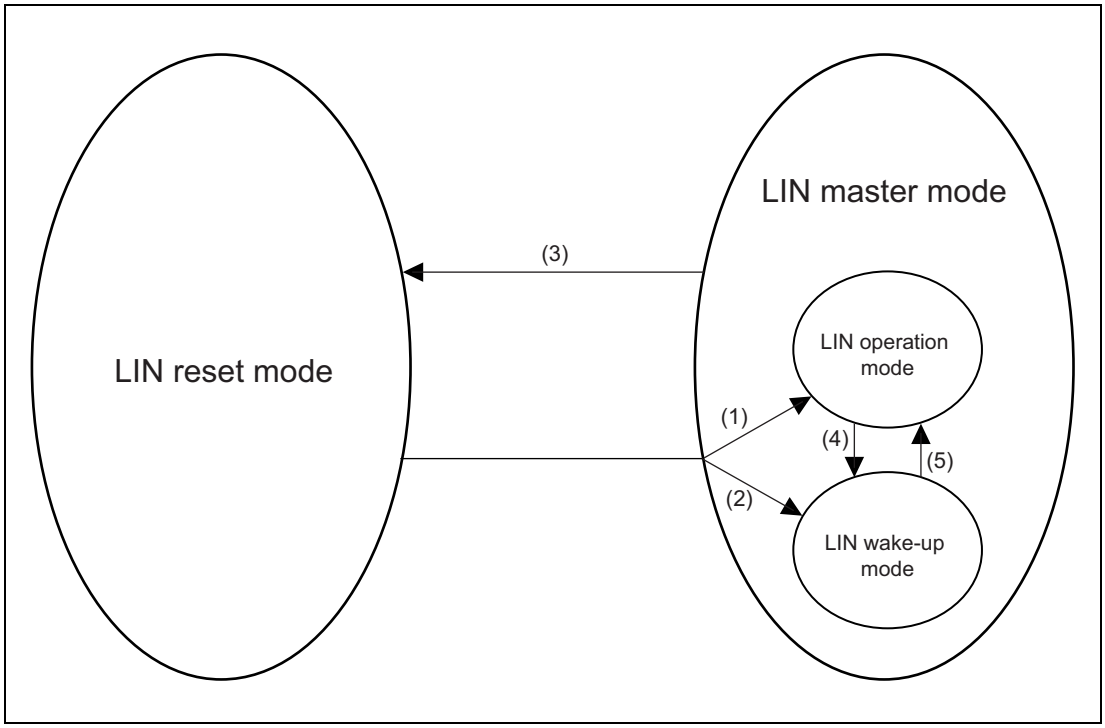


Figure 20.3 Transition of Operation Modes

Table 20.58 Transition Condition for Operation Mode

	Operation Mode Transition			Transition Condition
(1)	LIN reset mode	→	LIN mode • LIN operation mode	LMD[1:0] bit in RLN3nLMD register = 00_B and OM1 and OM0 bits in RLN3nLCUC register = 11_B
(2)	LIN reset mode	→	LIN mode • LIN wake-up mode	LMD[1:0] bit in RLN3nLMD register = 00_B and OM1 and OM0 bits in RLN3nLCUC register = 01_B
(3)	LIN mode • LIN operation mode • LIN wake-up mode	→	LIN reset mode	OM0 bit in RLN3nLCUC register = 0_B
(4) *1	LIN mode • LIN operation mode	→	LIN mode • LIN wake-up mode	OM1 and OM0 bits in RLN3nLCUC register = 01_B
(5) *1	LIN mode • LIN wake-up mode	→	LIN mode • LIN operation mode	OM1 and OM0 bits in RLN3nLCUC register = 11_B

Note 1. Transition between LIN operation mode and LIN wake-up mode cannot be made when communication is going on (when the FTS bit in the RLN3nLTRC register is 1).

(1) LIN Operation Mode

In LIN operation mode, frame processing (header transmission, header reception, response transmission, response reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 11_B changes the mode to LIN operation mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 11_B. Communication settings should be performed after the OMM1 and OMM0 bits have become 11_B.

(2) LIN Wake-up Mode

In LIN wake-up mode, wake-up signal processing (wake-up transmission, wake-up reception, and error detection) can be performed.

During a transition from LIN reset mode to LIN mode, setting the OM1 and OM0 bits in the RLN3nLCUC register to 01_B changes the mode to LIN wake-up mode, changing the OMM1 and OMM0 bits in the RLN3nLMST register to 01_B. Communication settings should be performed after the OMM1 and OMM0 bits have become 01_B.

20.7.1 LIN Master Mode

20.7.1.1 Header Transmission

Figure 20.4 shows the operation of the LIN/UART interface (LIN master mode) in header transmission. **Table 20.59** shows processing in header transmission.

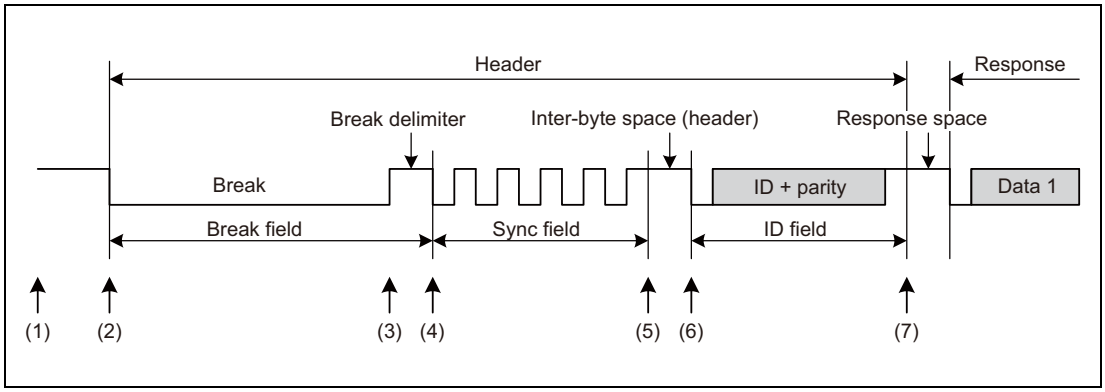


Figure 20.4 Operation in Header Transmission

Table 20.59 Processing in Header Transmission

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Enables interrupt • Enables error detection • Sets frame configuration parameters • Changes the LIN/UART interface to the LIN master mode: LIN operation mode • Sets information on the frame to be transmitted (ID, parity, data length, response direction, Checksum method, and transmission data)	Waits for the setting of the FTS bit in the RLN3nLTRC register by software (idle)
(2)	Sets the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started)	Transmits a break.
(3)	Waits for an interrupt request	Transmits a break delimiter.
(4)		Transmits a sync field (55_H).
(5)		Transmits an inter-byte space (header).
(6)		Transmits an ID field.
(7)		Sets a successful header transmission flag.

NOTE

For information about error detection conditions, see Section 20.7.6, Error Status.

20.7.1.2 Response Transmission

Figure 20.5shows the operation of the LIN/UART interface (LIN master mode) in response transmission. **Table 20.60** provides processing in response transmission.

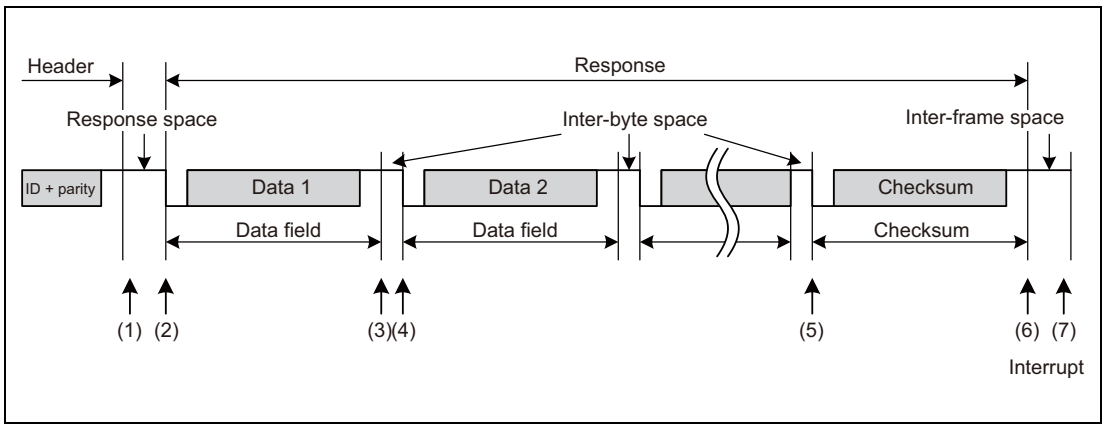


Figure 20.5 Operation in Response Transmission

Table 20.60 Processing in Response Transmission

	Software Processing	LIN/UART Interface Processing
(1)	[When in frame separate mode] • Sets the RTS bit in the RLN3nLTRC register to 1 (response transmission/reception started) [When not in frame separate mode] • Waits for an interrupt request	[When in frame separate mode] • Waits for the setting of the RTS bit in the RLN3nLTRC register to 1 by software. • When the bit is set to 1, sends a response space. [When not in frame separate mode] • Sends a response space.
(2)	Waits for an interrupt request	Transmits data 1.
(3)		Transmits an inter-byte space.
(4)		• Transmits data 2. • Transmits an inter-byte space • Transmits data 3. • Transmits an inter-byte space (Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : :
(5)		Transmits the checksum.
(6)		• Sets a successful frame/wake-up transmission flag. • Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped) [When in frame separate mode] • Set the RTS bit in the RLN3nLTRC register to 0 (response transmission/reception stopped).
(7)	• Processing after communication Checks the RLN3nLST register, and clears flags.	Idle

NOTE

For information about error detection conditions, see Section 20.7.6, Error Status.

20.7.1.3 Response Reception

Figure 20.6 shows the operation of the LIN/UART interface (LIN master mode) on response reception. **Table 20.61** provides processing in response reception.

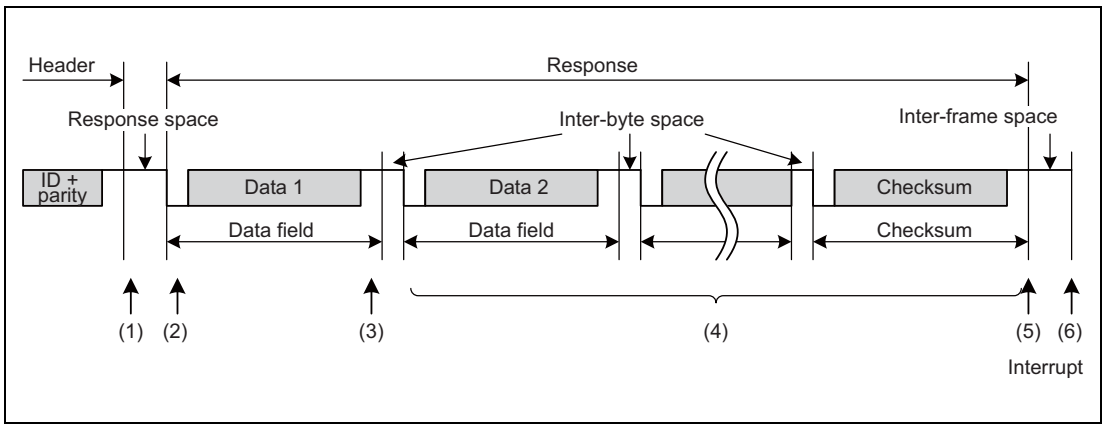


Figure 20.6 Operation in Response Reception

Table 20.61 Processing in Response Reception

	Software Processing	LIN/UART Interface Processing
(1)	Waits for an interrupt request (no processing)	Waits for detection of a start bit.
(2)		Receives data 1 when the start bit is detected.
(3)		Sets the successful data 1 reception flag.
(4)		• Receives data 2 when the start bit is detected. • Receives data 3 when the start bit is detected. Repeats the transmission of inter-byte spaces as many times as the data length specified in bits RFDL[3:0] in the RLN3nLDFC register). : : • Receives the checksum when the start bit is detected.
(5)		• Determines the checksum. • Sets the successful frame/wake-up reception flag. • Sets the FTS bit in the RLN3nLTRC register to 0 (frame transmission or wake-up transmission/reception stopped).
(6)	• Processing after communication Reads the received data. Checks the RLN3nLST register, and clears flags.	Idle

NOTE

For information about error detection conditions, see Section 20.7.6, Error Status.

20.7.2 Data Transmission/Reception

20.7.2.1 Data Transmission

One bit of data is transmitted per 1 Tbit.

The data that is transmitted returns to the reception data input pin via the LIN transceiver. The received data and the transmitted data is compared bit by bit, and the results are stored in the BER flag in the RLN3nLEST register (see Section 20.7.6, Error Status).

Figure 20.7 shows an example of data transmission timing.

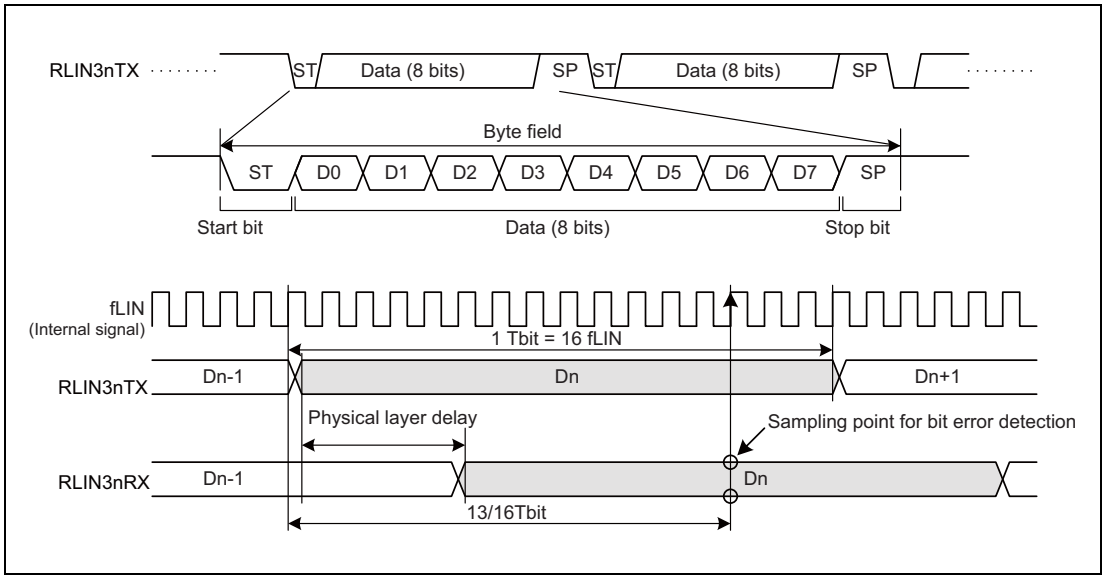


Figure 20.7 Example of Data Transmission Timing (LIN Master Mode)

20.7.2.2 Data Reception

Data reception is performed by using the synchronized RLIN3nRX signal (an internal signal) that is the input from the RLIN3nRX pin synchronized with prescaler clock.

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, sampling is performed again 0.5 Tbit later, and the falling edge is recognized as a start bit if the synchronized RLIN3nRX signal is low level. The falling edge is not recognized as a start bit if the RLIN3nRX signal after the clearing of the resetting is low-level-fixed or if a high level is detected on re-sampling.

After the start bit is detected, the system samples 1 bit per Tbit.

The LIN/UART interface has a noise filter function with respect to reception data. If the LRDNFS bit in the RLN3nLMD register is 0, the LIN/UART interface uses a noise filter, and for a sampling value the value determined by a 3-sampling majority rule on prescaler clocks is used. If the LRDNFS bit in the RLN3nLMD register is 1, the LIN/UART interface does not use a noise filter, and for a sampling value the value of the synchronized RLIN3nRX value at the sampling position is used as is.

Figure 20.8 shows an example of data reception timing.

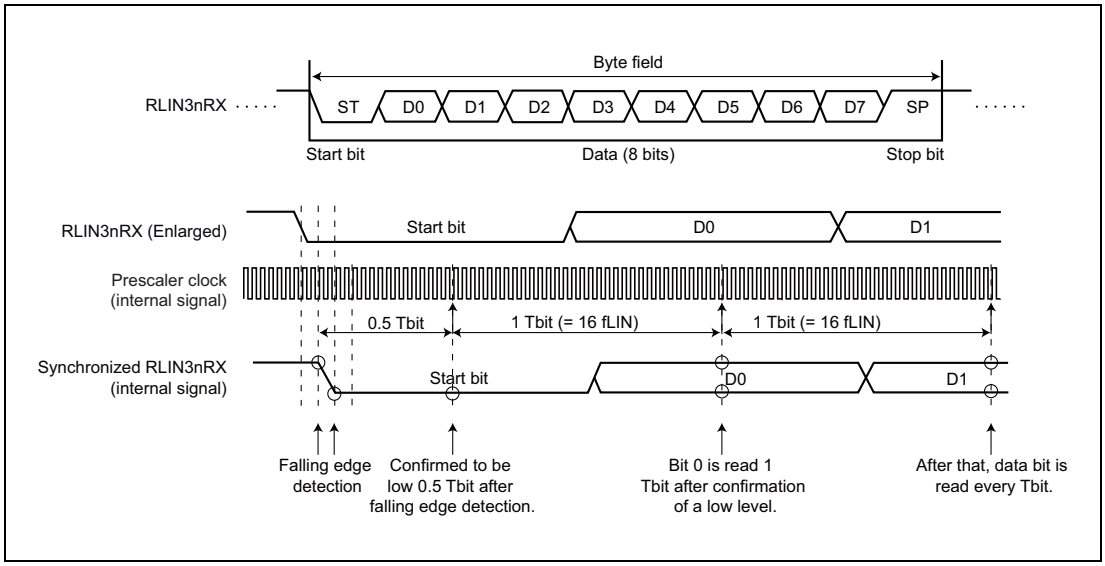


Figure 20.8 Example of Data Reception Timing (LIN Master Mode)

20.7.3 Transmission/Reception Data Buffering

This section explains the buffer processing that takes place when the LIN/UART interface sends or receives data continuously.

20.7.3.1 Transmission of LIN Frames

For an 8-byte transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR8 are sequentially transmitted to data areas 1 to 8 of the LIN frame. In the case of a 4-bytes transmission, the contents stored in registers RLN3nLDBR1 to RLN3nLDBR4 are transmitted to data areas 1 to 4 of the LIN frame, but the contents of registers RLN3nLDBR5 to RLN3nLDBR8 are not transmitted. The transmitted checksum data is stored in the RLN3nLCBR register.

Figure 20.9 depicts the LIN transmission processing and the required buffer.

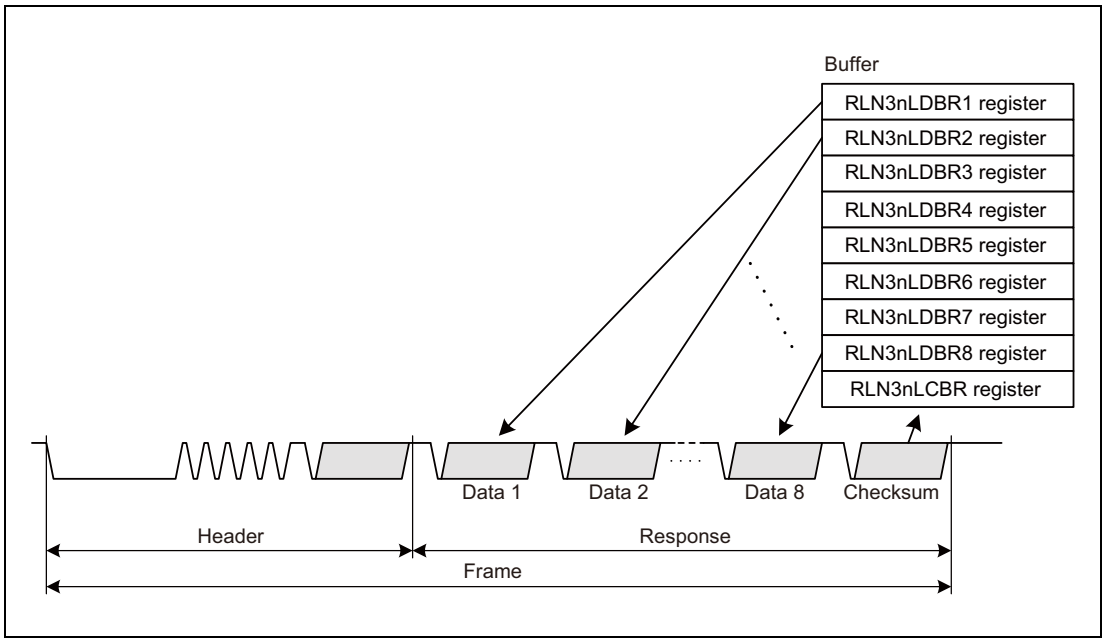


Figure 20.9 LIN Transmission Processing and Required Buffer

[Frame Separate Mode]

Setting the FSM bit in the RLN3nLDFC register to 1 turns on the frame separate mode.

In frame separate mode, the header and response are separately transmitted when prompted by respective transmission start requests.

When the transmission of a header is finished, the HTRC flag in the RLN3nLST register turns 1 (successful header transmission).

Use frame separate mode when sending or receiving response data of 9 bytes or greater in LIN master mode.

20.7.3.2 Reception of LIN Frames

For an 8-byte reception, the contents of data areas 1 to 8 of the LIN frame is stored in registers RLN3nLDBR1 to RLN3nLDBR8, respectively, upon receipt of a stop bit. In the case of a 4-byte reception, the contents of data areas 1 to 4 of the LIN frame are stored in registers RLN3nLDBR1 to RLN3nLDBR4, respectively; however, no data is stored in registers RLN3nLDBR5 to RLN3nLDBR8. Also, the received checksum data is stored in the RLN3nLCBR register.

Figure 20.10 depicts the LIN reception processing and the required buffer.

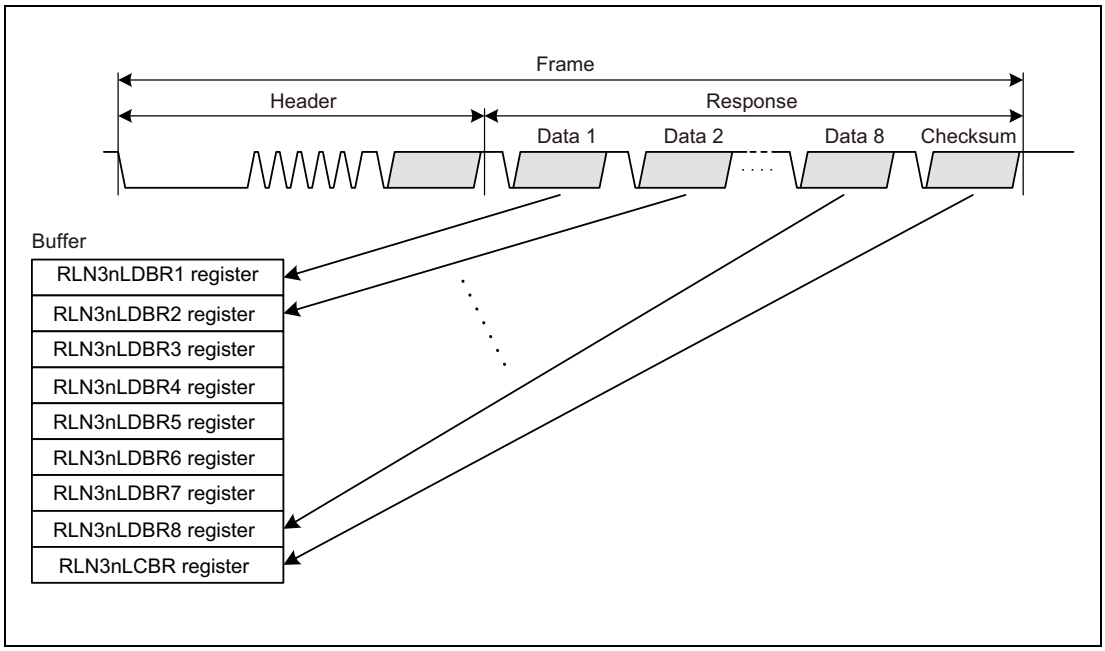


Figure 20.10 LIN Reception Processing and Required Buffer

[Reception of Data 1]

When the reception of the first byte of data is finished, the D1RC flag in the RLN3nLST register turns 1 (successful data 1 reception).

20.7.3.3 Multi-Byte Response Transmission/Reception Function

Normally in LIN communications, a response is 9 bytes or less including a checksum field; however, responses in 10 bytes or greater can also be sent and received.

In such a case, the bit error, framing error, response preparation error detection, and auto checksum functions are enabled.

If the data length is greater than 8 bytes, the LSS bit in RLN3nLDFC register should be set to 1 (indicating that the next data group to be transmitted or received is not the final data group) in the first data group (variable from 0 to 8 bytes) before transmitting or receiving the data group. After the transmission or reception, the user should determine whether the next data group is the final data group. If it is the final data group, the LSS bit in the RLN3nLDFC register should be set to 0 (indicating that the next data group to be transmitted or received is the final data group) before transmitting or receiving the data group, and a checksum should be appended to the final data group.

By changing the RFDL[3:0] bit in RLN3nLDFC register settings when the RTS bit in RLN3nLTRC register is 0, the user can change the data length for each data group.

When performing multi-byte response transmission/reception in LIN master mode, set the FSM bit in RLN3nLDFC register in the RLN3nLDFC register to 1 (frame separate mode).

20.7.4 Wake-up Transmission/Reception

The wake-up transmission/reception can be used in LIN wake-up mode.

20.7.4.1 Wake-up Transmission

In LIN wake-up mode, setting the RFT bit in the RLN3nLDFC register to 1 (LIN master mode: response transmission), and the FTS bit in the RLN3nLTRC register to 1 (frame transmission, header reception or wake-up transmission/reception started) causes a wake-up signal to be output from the output pin. The low width of the wake-up signal should be set using the WUTL[3:0] bits in the RLN3nLWUP register.

However, if the LWBR0 bit of the RLN3nLWBR register in LIN master mode is 1 (LIN2.x use), the LIN system clock (fLIN) becomes low level width at fa regardless of the setting of the LCKS[1:0] bit of the RLN3nLMD register. By setting the WUTL[3:0] bit of the RLN3nLWUP register to 0100_B (5 Tbits), 260 µs low width can be output in LIN wake-up mode regardless of the setting of the LCKS[1:0] bit of the RLN3nLMD register.

If a wake-up low is output without any bit error, the FTC flag in the RLN3nLST register turns 1 (successful frame response or wake-up transmission); when the FTCIE bit in the RLN3nLIE register is 1 (successful frame response/wake-up transmission interrupt enabled), an interrupt request for RLIN3n transmission is generated.

If RLN3nLEDE.BERE is set and a bit error is detected, wake-up transmission is canceled and the BER flag in the RLN3nLEST register is set to 1 (bit error detection).
When RLN3nLEDE.PBERE is set in LIN master mode, set RLN3nLEST.PBER flag to 1 (physical bus error detection) at the same time of a bit error.

Figure 20.11 shows the wake-up transmission timing.

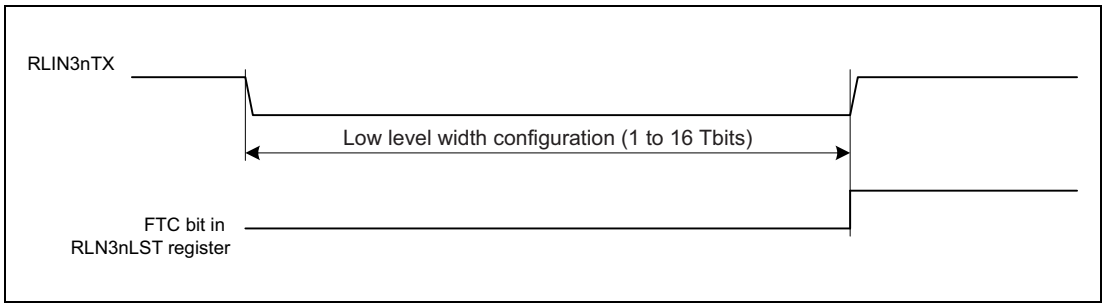


Figure 20.11 Wake-up Transmission Timing

20.7.4.2 Wake-up Reception

The detection of a wake-up involves the use of an input signal low level width count function. The input signal low level width count function measures the low level width of the input signal to the RLIN3nRX pin, using the same sampling point as data reception. This allows the 2.5-Tbit or longer low-level width of the input signal of fLIN to be measured.

In LIN master mode, by setting the LWBR0 bit in the RLN3nLWBR register, operation is executed without changing the baud rate generator setting at a transition between LIN operation mode and LIN wake-up mode.

When LIN Specification Package Revision 1.3 is used, set the LWBR0 bit in the RLN3nLWBR register to 0. When LIN Specification Package Revision 2.x is used, set the LWBR0 bit to 1. Setting the LWBR0 bit to 1 selects the LIN system clock (fLIN) to fa regardless of the setting of the LCKS[1:0] bit in the RLN3nLMD register. (The LCKS[1:0] bit is not changed). By setting the baud rate to 19200bps while fa is selected, the 130 µs or longer low-level width of the input signal to be measured regardless of the setting of the LCKS[1:0] bit in the RLN3nLMD register.

When using this function, in LIN wake-up mode set the RFT bit in the RLN3nLDFC register to 0 (LIN master mode: response reception), or the FTS bit in the RLN3nLTRC register to 1 (frame transmission (header reception) or wake-up transmission/reception started).

When the low level width to be measured is reached, the FRC flag in the RLN3nLST register turns 1 (successful frame response/wake-up reception). If the FRCIE bit in the RLN3nLIE register is 1 (successful frame response or wake-up reception interrupt enabled), an interrupt request for successful RLIN3n reception is generated.

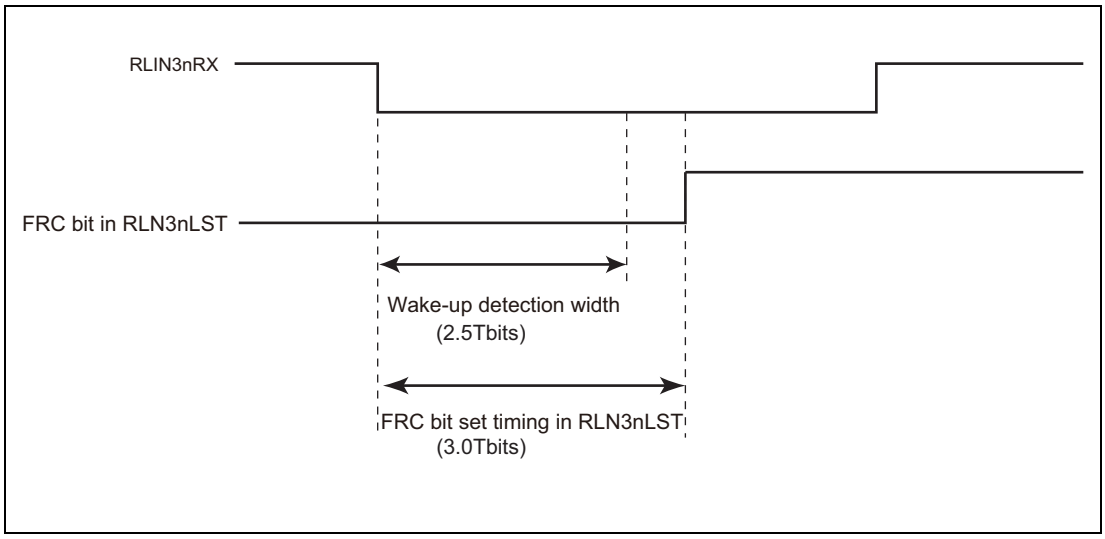


Figure 20.12 Input Signal Low level Count Function

20.7.4.3 Wake-up Collision

If the master node and the slave node transmit wake-up signals simultaneously, a collision will occur on the LIN bus, though a collision of wake-up signals is not detected in the LIN/UART interface.

20.7.5 Status

During LIN mode operation, the LIN/UART interface can detect seven types of statuses.

The four statuses, successful frame/wake-up transmission, successful frame/wake-up reception, error detection, successful header transmission/reception, can generate interrupt requests.

Table 20.62 shows the types of statuses available in LIN master mode.

Table 20.62 Types of Statuses in LIN Master Mode

Status	Status Set Condition	Status Clear Condition	Operation Mode Capable of Status Detection	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN–reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode.	All modes	OMM0 bit in RLN3nLMST register	—
Operation mode	After the OM1 bit in the RLN3nLCUC register is set to LIN operation mode, if actually the LIN/UART interface enters LIN operation mode.	After the OM1 bit in the RLN3nLCUC register is set to LIN wake-up mode, if actually the LIN/UART interface enters LIN wake-up mode.	• LIN operation mode • LIN wake-up mode	OMM1 bit in RLN3nLMST register	—
Frame/wake-up transmission end	When a frame (header transmission + response transmission), a wake-up signal, or a data group is transmitted successfully.	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FTC flag in RLN3nLST register	√
Frame/wake-up reception end	When a frame (header transmission + response reception), a wake-up signal, or a data group is received successfully.	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	FRC flag in RLN3nLST register	√
Error detection	If any of the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, and BER flags in the RLN3nLEST register turns 1 (error detected).	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software**Note 1.** • After transition to LIN reset mode	• LIN operation mode • LIN wake-up mode	ERR flag in RLN3nLST register	√
Data 1 reception end	The RFT bit in the RLN3nLDFC register is 0 (reception) and the first byte of the response field or the first byte of each data group is received.*2	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • • When cleared by software • After transition to LIN reset mode	LIN operation mode	D1RC flag in RLN3nLST register	—
Header transmission end	When a header field is transmitted successfully.	• When another communication is started (When the FTS bit in the RLN3nLTRC register is set) • When cleared by software • After transition to LIN reset mode	LIN operation mode	HTRC flag in RLN3nLST register	√

Note 1. In LIN operation mode, the ERR flag in the RLN3nLST register is cleared to 0 by writing 0 to the RPER flag, CSER flag, FER flag, FTER flag, PBER flag, or BER flags in the RLN3nLEST register.

Note 2. Not detected when the RFDL [3:0] bits in the RLN3nLDFC register are 0000_B (0-byte + checksum).

20.7.6 Error Status

(1) Types of Error Statuses

The LIN/UART interface can detect six types of error statuses in LIN master mode. The condition of these error statuses can be checked by means of the corresponding bits in the RLN3nLEST register.

All error statuses represent interrupt events.

Table 20.63 shows the types of error statuses.

Table 20.63 Types of Error Statuses in LIN Master Mode

Status	Error Detection Condition	Operation Mode Capable of Error Detection	Communication	Enable/Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data on the LIN bus monitored by the receive pin do not match *1*2	• LIN operation mode • LIN wake-up mode	Cancel	Enabled	BER flag in RLN3nLEST register
Physical bus error	• LIN bus is detected to be high level when sending a break • LIN bus is detected to be low level when sending a break delimiter • LIN bus is detected to be high level when sending a wake-up	• LIN operation mode • LIN wake-up mode	Cancel	Enabled	PBER flag in RLN3nLEST register
Timeout error	A frame or response transmission/reception does not terminate within a given time*3	LIN operation mode	Cancel	Enabled	FTER flag in RLN3nLEST register
Framing error	In response field reception, a stop bit of each data byte is low level	LIN operation mode	Cancel	Enabled	FER flag in RLN3nLEST register
Checksum error	In response field reception, the result of checksum test gives an error	LIN operation mode	—	Disabled	CSER flag in RLN3nLEST register
Response preparation error	One of the following conditions occurs in frame separate mode during a multi-byte response reception: • The first reception data byte is received after completion of header transmission but before a response transmission/reception request is set • The first reception data byte is received after the completion of previous data group reception before a transmission/reception request for another data group is set.	LIN operation mode	Cancel	Disabled	RPER flag in RLN3nLEST register

Note 1. If a bit error is detected, the process is canceled after a stop bit is sent. If a bit error is detected in a non-data area, such as an inter-byte space, the transmission is canceled immediately after that area. If a bit error is detected during the transmission of a wake-up, the transmission of the wake-up is canceled after the error-causing bit is sent.

Note 2. In a multi-byte response transmission, bit errors are detected also between data groups.

Note 3. The timeout time depends on the response field data length (the RFDL [3:0] bits in the RLN3nLDFC register) and the checksum selection (the CSM bit in the RLN3nLDFC register), and this can be calculated according to the following formula:
When the FSM bit in the RLN3nLDFC register is set to 1 (frame separate mode), the timeout time is that of the 8 data bytes until the RTS bit of the RLN3nLTRC register is set. Once the RTS bit is set, the timeout time is changed to the time based on the response field data length (the RFDL[3:0] bits in the RLN3nLDFC register).

[Frame timeout]
On classic selection (when the CSM bit in RLN3nLDFC is 0): Timeout time = 49 + (number of data bytes + 1) × 14 [Tbit]
On enhanced selection (when the CSM bit in RLN3nLDFC is 1): Timeout time = 48 + (number of data bytes + 1) × 14 [Tbit]
The aforementioned timeout time is a time greater than the TFRAME_MAX of LIN Specification Package Revision 1.3 on classic selection, or the TFRAME_MAX of LIN Specification Package Revision 2.x on enhanced selection.
[Response timeout]
Timeout time = (number of data bytes + 1) × 14 [Tbit]

When an error is detected, time-out error detection function stops.

The error status is cleared when the next communication is started (when the FTS bit in the RLN3nLTRC register is set), by software, or at a transition to LIN reset mode.

(2) Target Time Area for LIN Error Detection

Figure 20.13 shows the time domain in which the LIN/UART interface in LIN master mode performs monitoring for error detection.

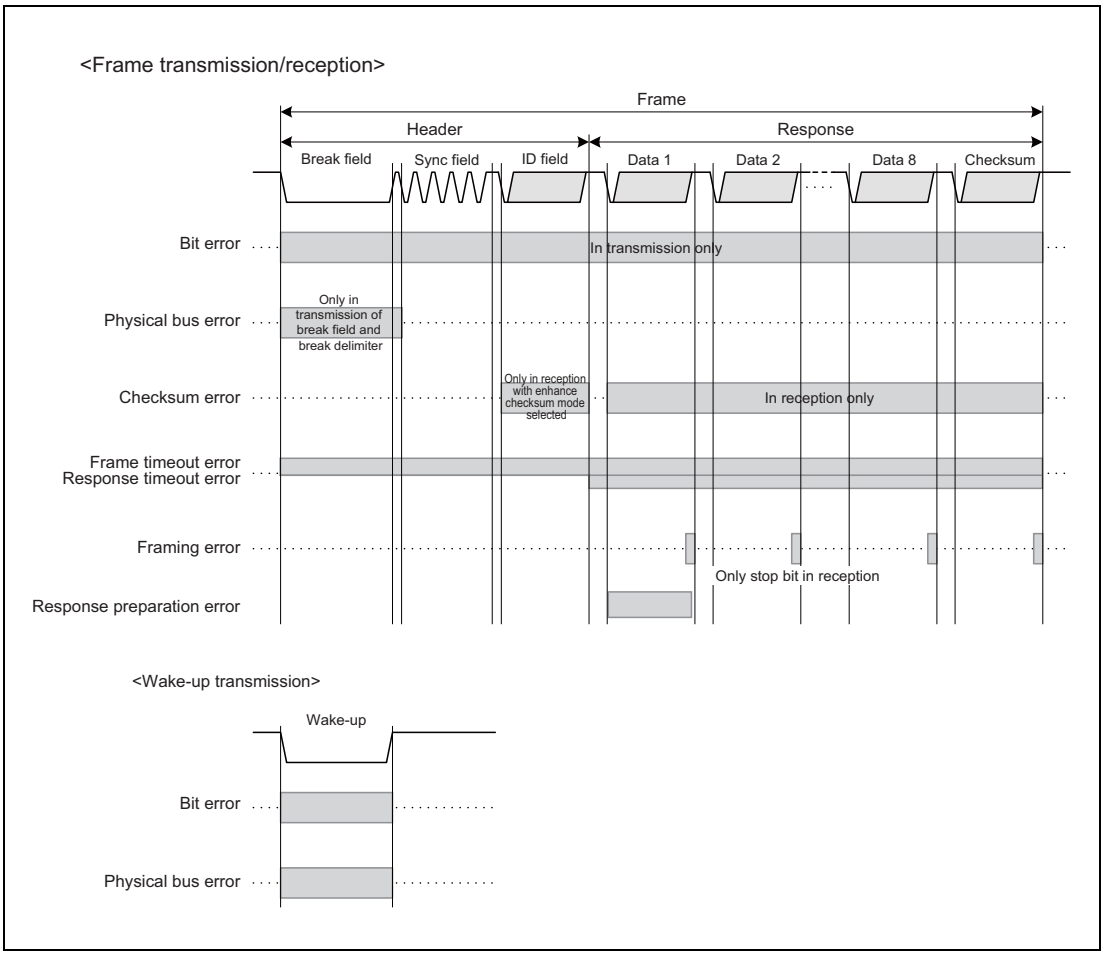


Figure 20.13 Target Time Area for LIN Error Detection (LIN Master Mode)

20.8 UART Mode

In LIN reset mode, setting the LMD[1:0] bits in the RLN3nLMD register to 01_B (UART mode) and the OM0 bit in the RLN3nLCUC register to 1 changes the mode to UART mode, turning the OMM0 bit in the RLN3nLMST register to 1.

20.8.1 Transmission

Figure 20.14 shows LIN/UART interface (in UART mode) transmission operations; **Table 20.64** shows LIN/UART interface (in UART mode) transmission processing.

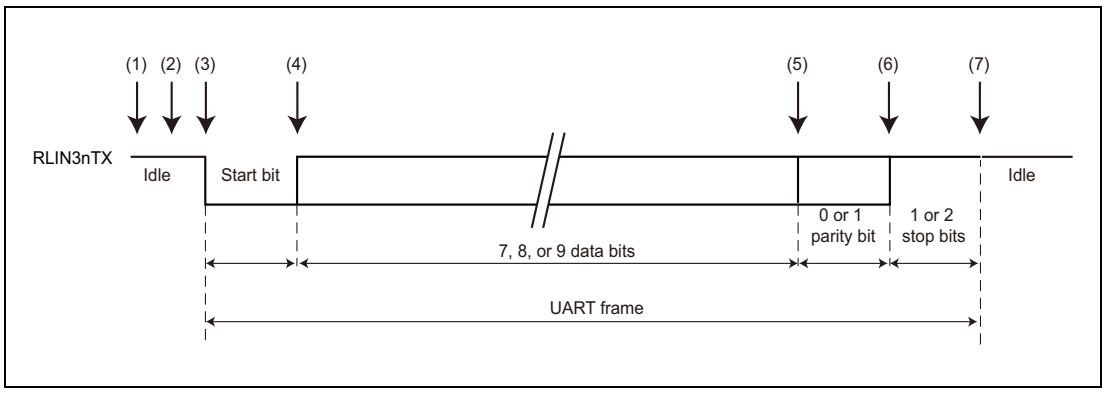


Figure 20.14 LIN/UART Interface (in UART Mode) Transmission Operation

Table 20.64 LIN/UART Interface (UART Mode) Transmission Processing (1/2)

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate. • Sets noise filter ON/OFF. • Sets error detection enable. • Sets data format. • Sets an interrupt generation timing. • Clears the LIN/UART interface from LIN reset mode. • Sets the transmit enable bit (UTOE bit) to 1.	• Waits for a transmission trigger (RLN3nLUTDR register) by software.
(2)	• Sets the transmit data to the UART transmit data register (RLN3nLUTDR) or UART wait transmit data register (RLN3nLUWTDR).	• Sets the transmit status flag.
(3)	• Waits an interrupt request.	• Transmits a start bit (for switching between transmission and reception in half duplex communication, transmits a start bit after receiving 1 stop bit. For details about this function, see Section 20.8.1.4, Transmission Start Wait Function.
	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • When transmitting data continuously, sets another piece of transmission data in the UART transmit data register (RLN3nLUTDR register), waits for the generation of an interrupt request.	[When the UTIGTS bit is 0 (a transmission interrupt is generated)] • Generates a RLIN3n transmission interrupt request.
(4)		Transmits the data set in the UART (for wait) transmit data register.
(5)		Transmits a parity bit when parity is used.
(6)		Transmits 1 or 2 stop bits.

Table 20.64 LIN/UART Interface (UART Mode) Transmission Processing (2/2)

	Software Processing	LIN/UART Interface Processing
(7)	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • If another piece of transmission data is set, goes to step (3).	[When the UTIGTS bit is 0 (a transmission interrupt request is generated upon start of transmission)] • If another piece of transmission data is set, goes to step (3). • If another piece of transmission data is not set, clears the transmit status flag.
	[When the UTIGTS bit is 1 (a transmission interrupt is output upon end of transmission)] • When transmitting data continuously, goes to step (2).	[When the UTIGTS bit is 1 (a transmission interrupt is output upon end of transmission)] • Generates a RLIN3n transmission interrupt request. • Clears the transmission status flag.

NOTE

For information about error detection conditions, see Section 20.8.5, Error Status.

20.8.1.1 Continuous Transmission

The LIN/UART interface (in UART mode) can transmit multiple sets of data continuously by using the RLN3nLUTDR register. **Figure 20.15** shows an operation example where the transmission interrupt generation timing is the start of transmission and an operation example where the transmission interrupt generation timing is the end of transmission.

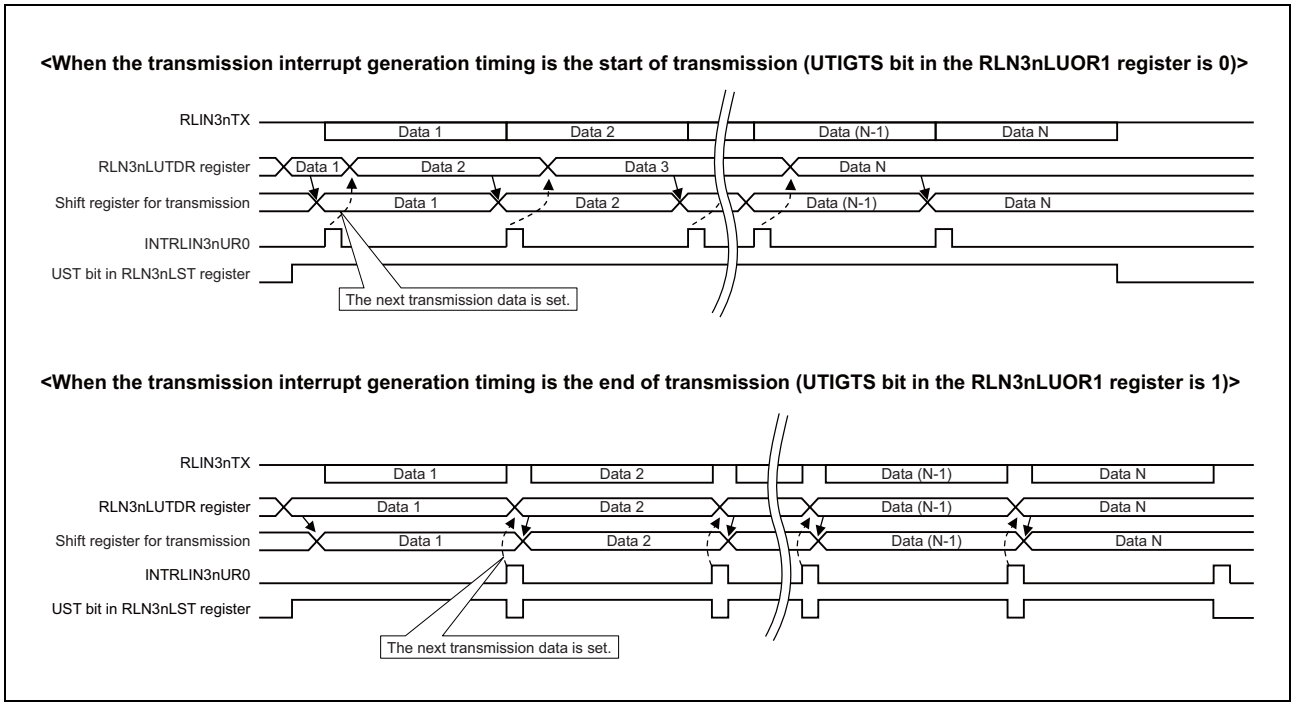


Figure 20.15 Operation Example of LIN/UART Interface (UART Mode) Continuous Transmission

An interrupt can be generated at the end of a transmission by changing the UTIGTS bit in the RLN3nLUOR1 register from 0 to 1 after the start of transmission of final data, provided only that the transmission interrupt generation timing is the start of transmission and the end of transmission of final data needs to be known.

20.8.1.2 UART Buffer Transmission

The LIN/UART interface (in UART mode) has a maximum of nine bytes of UART buffers, and thus it is capable of performing continuous transmissions through the use of UART buffers.

Figure 20.16 shows the UART buffer transmission operation in the LIN/UART interface (in UART mode). **Table 20.65** shows the UART buffer transmission processing.

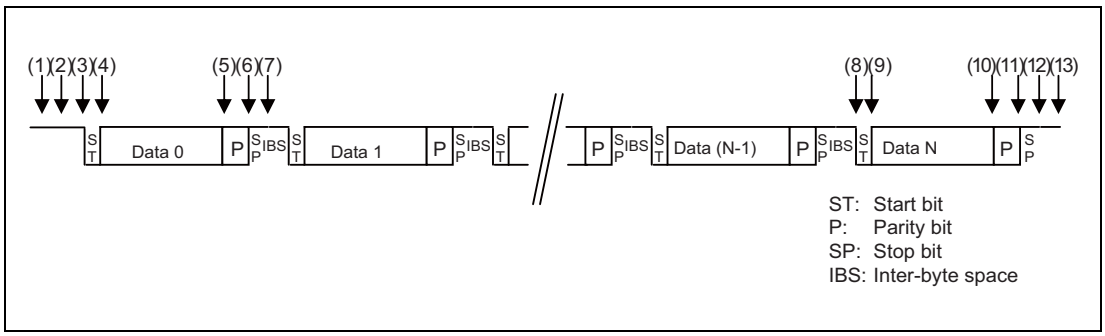


Figure 20.16 UART Buffer Transmission in LIN/UART Interface (in UART Mode)

Table 20.65 UART Buffer Transmission Processing in LIN/UART Interface (in UART Mode) (1/2)

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate • Sets noise filter ON/OFF • Sets error detection enable • Sets data format • Sets an interrupt generation timing to the end of transmission. • Clears the LIN/UART interface from LIN reset mode. • Sets the transmit enable bit (UTOE bit) to 1	• Waits for a transmission trigger (RTS bit) by software
(2)	• Sets the UART buffer data length and whether the system must wait for the start of transmission. • Specifies the transmission data to be transmitted in the UART data 0 buffer register (RLN3nLUDB0) and the LIN data buffer b register (RLN3nLDBRb). (b =1 to 8) • Sets the UART buffer transmission start bit (RTS).	• Sets the transmit status flag.

Table 20.65 UART Buffer Transmission Processing in LIN/UART Interface (in UART Mode) (2/2)

	Software Processing	LIN/UART Interface Processing
(3)	Waits for an interrupt request.	Transmits a start bit. (When switching from reception to transmission during half-duplex communication, transmits the start bit upon completion of the stop bit for reception. For details about this function, see **Section 20.8.1.4, Transmission Start Wait Function**.)
(4)		Transmits the data specified in the UART data buffer 0 register (RLN3nLUDB0) or the LIN/UART data buffer b register (RLN3nLDBRb).
(5)		Transmits a parity bit when parity is used.
(6)		Transmits 1 or 2 stop bits (When the number of data set in UART buffer data length setting bit, proceed to (12).)
(7)		Transmits an inter-byte space (idle).
		Repeats steps (3) to (7) until the number of data -1 that was set in the UART buffer data length select bits is reached.
(8)		Transmits a start bit.
(9)		Transmits the data set in the LIN/UART data buffer b register (RLN3nLDBRb).
(10)		Transmits a parity bit when parity is used.
(11)		Transmits 1 or 2 stop bits.
(12)		• Sets the buffer transmission end flag. • Clears the UART buffer transmit start bit (RTS). • A transmission interrupt request signal. • Clears the transmission status flag.
(13)	• Checks the RLN3nLST register, and clears flags • In the case of continuous data transmission, goes to step (2).	

NOTE

For information about error detection conditions, see Section 20.8.5, Error Status.

(1) UART Buffer Transmission

For a 9-byte transmission, the contents stored in the RLN3nLUDB0 and RLN3nLDBR1 to RLN3nLDBR8 registers are transmitted to data areas 0 to 8. The RLN3nLUDB0 register is used only if 9-byte transmission is set. In other cases, the RLN3nLDBR1 to RLN3nLDBR8 registers are selected depending upon the length of data involved. For a 4-byte transmission, the contents stored in the RLN3nLDBR1 to RLN3nLDBR4 registers are transmitted to data areas 1 to 4, but the contents of the RLN3nLDBR5 to RLN3nLDBR8 registers are not transmitted. An RLIN3n transmission interrupt is generated after the transmission of the data that is set in the MDL[3:0] bits of the RLN3nLDFC register. The spaces between transmission data items can be set in the IBS[1:0] bit in the RLN3nLSC register.

Figure 20.17 shows a 9-byte UART buffer and the transmission processing.

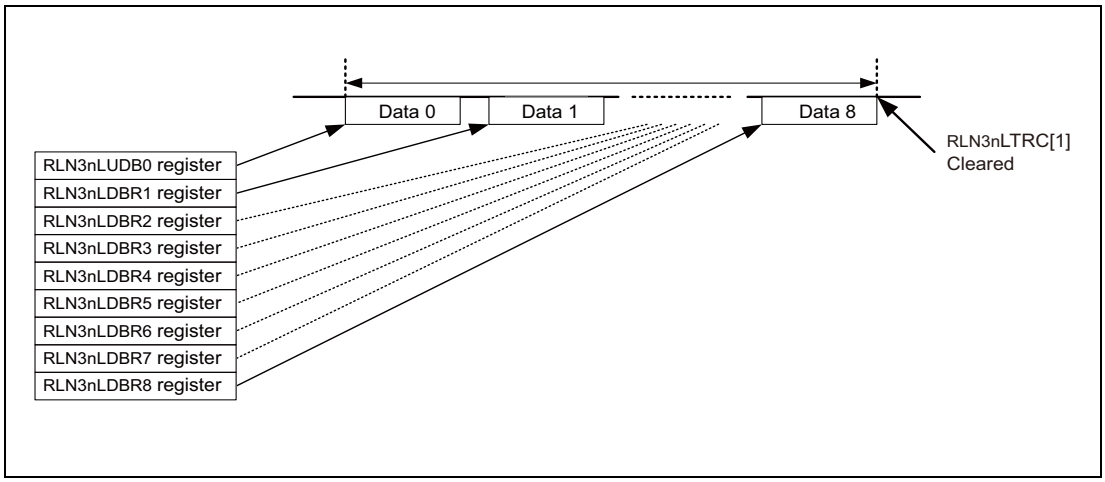


Figure 20.17 UART Buffer and Transmission Processing (for 9-Byte Transmission)

20.8.1.3 Data Transmission

One bit of data is transmitted per Tbit.

In half-duplex communication, if the BERE bit in the RLN3nLEDE register is 1 (bit error detection enabled), the transmission data and the input pin level are compared bit by bit during data transmission, and the results are stored in the BER flag in the RLN3nLEST register (see Section 20.7.6, Error Status). The timing at which the input pin is sampled during data transmission can vary depending upon the settings of the LPRS[2:0] and NSPB[3:0] bits in the RLN3nLWBR register.

The bit error detection timing in UART mode is shown in **Table 20.66**.

Table 20.66 Error Detection Timing in UART Mode

Sampling Count Per Bit	Bit Error Detection Timing
6 samples	3rd clock cycle + 1 prescaler clock
7 samples	4th clock cycle + 1 prescaler clock
8 samples	4th clock cycle + 1 prescaler clock
9 samples	5th clock cycle + 1 prescaler clock
10 samples	5th clock cycle + 1 prescaler clock
11 samples	6th clock cycle + 1 prescaler clock
12 samples	6th clock cycle + 1 prescaler clock
13 samples	7th clock cycle + 1 prescaler clock
14 samples	7th clock cycle + 1 prescaler clock
15 samples	8th clock cycle + 1 prescaler clock
16 samples	8th clock cycle + 1 prescaler clock

Example of Data Transmission Timing (when 1 Tbit = 16 Sampling) is shown in **Figure 20.18**

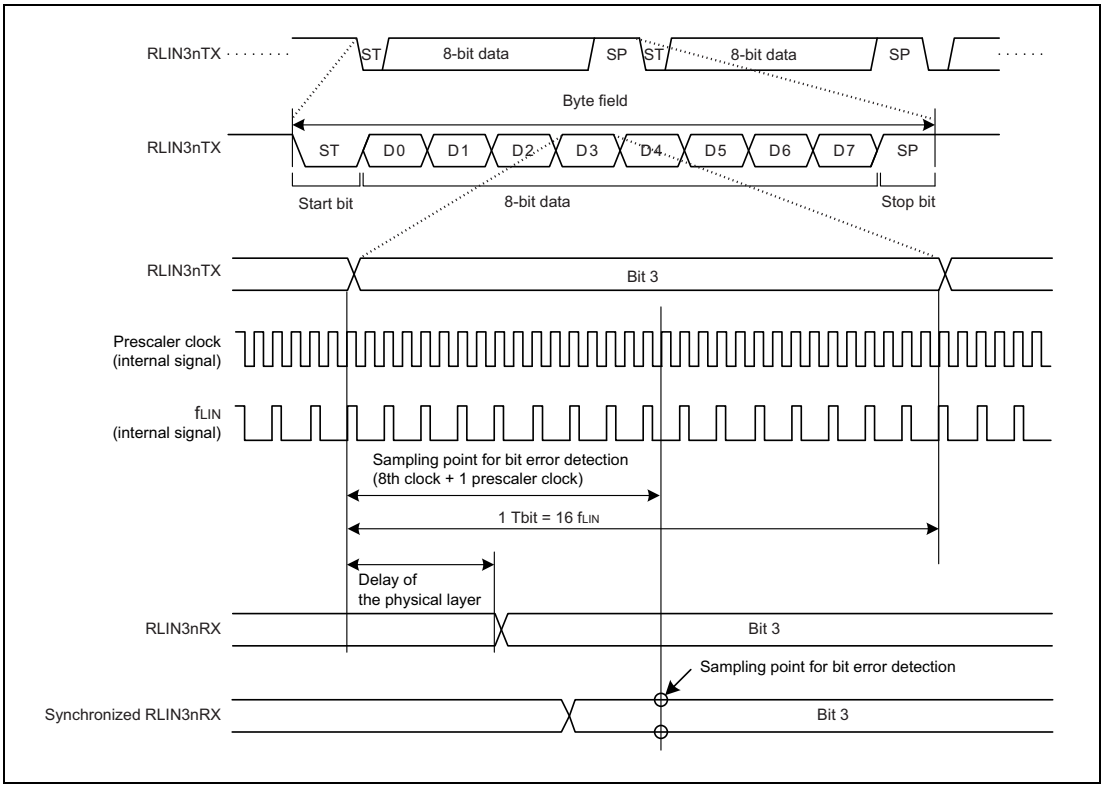


Figure 20.18 Example of Data Transmission Timing (When 1 Tbit = 16 Sampling)

20.8.1.4 Transmission Start Wait Function

For performing half-duplex communication, the LIN/UART interface (in UART mode) has the function of securing the reception stop bit length when switching from reception to transmission.

If it is desired to delay the start of transmission until the stop bits for the reception are completed, set data in the RLN3nLUWTDR register, which is used only for the wait function, instead of setting transmission data in the RLN3nLUTDR register as a start-of-transmission request. When transmitting from the UART buffer, set 1 (UART buffer transmission started) in the RTS bit in the RLN3nLTRC register with 1 set in the UTSW bit in the RLN3nLDFC register.

In such a case, the LIN/UART interface delays the start of transmission until the stop bits of reception data are completed.

It should be noted that even if the UART stop bit length select bit (USBLS) in RLN3nLBFC register is 1 (stop bits = 2 bits), delay is made only for 1 bits.

Figure 20.19 shows the operation of transmission wait function.

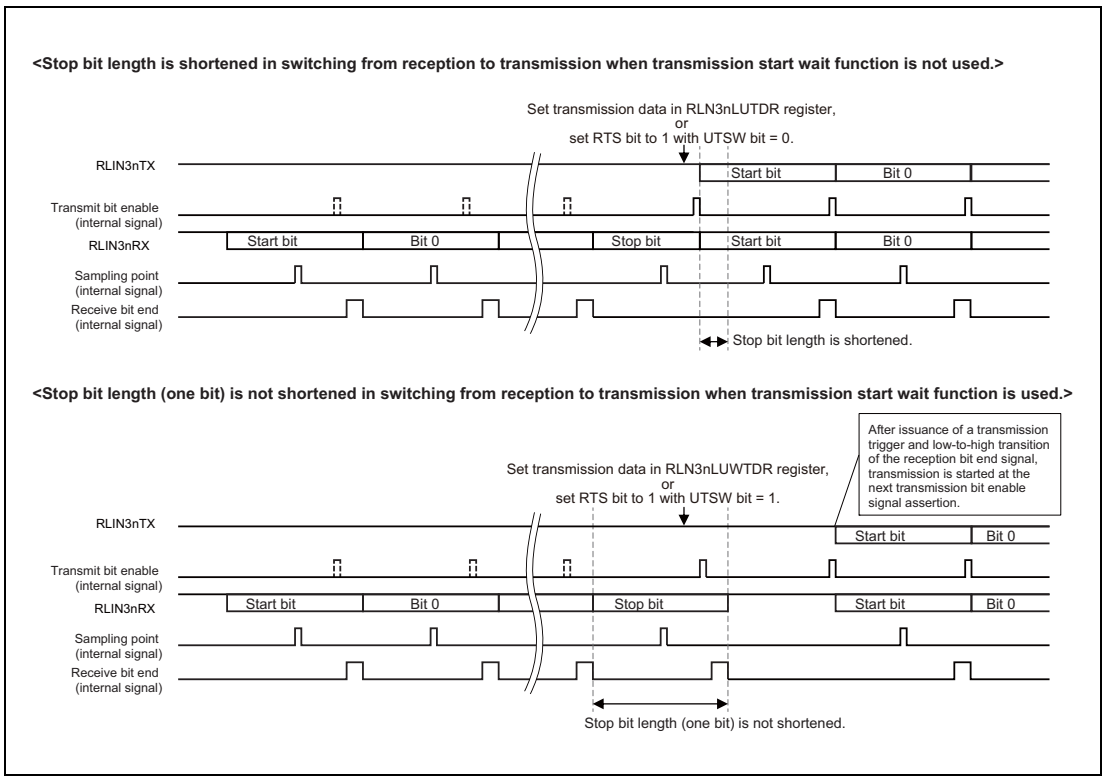


Figure 20.19 Transmission Wait Function (If Transmission Data is Set during the Stop Bits in the Received Data)

20.8.2 Reception

Figure 20.16 shows the LIN/UART interface (in UART mode) reception operation.**Table 20.67** shows the LIN/UART interface (in UART mode) reception processing.

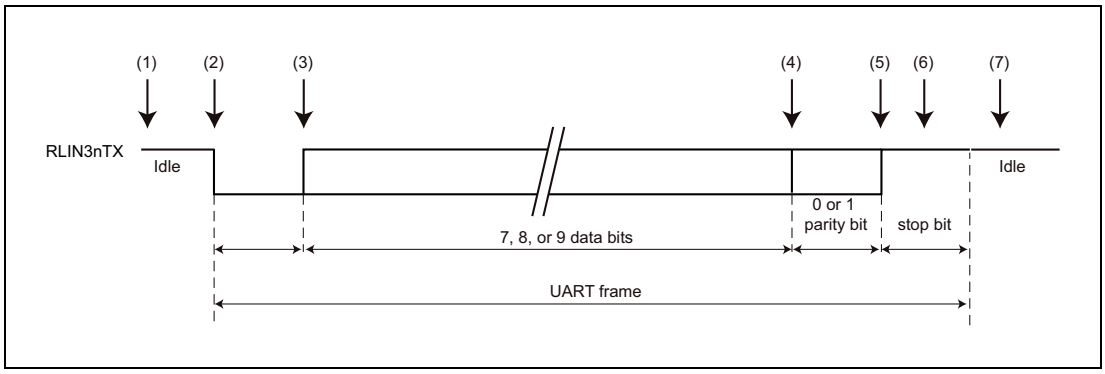


Figure 20.20 LIN/UART Interface (in UART Mode) Reception Operation

Table 20.67 LIN/UART Interface (in UART Mode) Reception Processing

	Software Processing	LIN/UART Interface Processing
(1)	• Sets a baud rate. • Sets noise filter ON/OFF. • Sets error detection enable. • Sets data format. • Clears the LIN/UART interface from LIN reset mode. • Sets the receive enable bit (UROE bit) to 1.	• Waits for reception enable state switching by software. • Waits for detection of a start bit.
(2)	Waits for an interrupt request.	• Waits for a falling edge from the reception pin, and detects a start bit. • Sets the reception status flag.
(3)		Receives data.
(4)		Receives a parity bit when parity is used.
(5)		Receives only 1 stop bit.
(6)		• Generates a successful RLIN3n reception interrupt request. • Clears the reception status flag.
(7)	Checks the RLN3nLST register, and clears flags	Waits for a falling edge from the reception pin.

NOTE

For information about error detection conditions, see Section 20.8.5, Error Status.

20.8.2.1 Data Reception

Data reception is performed by using the synchronized RLIN3nRX (an internal signal) that is the input from the RLIN3nRX pin synchronized with the prescaler clock.

The byte field is synchronized at the falling edge of the start bit for the synchronized RLIN3nRX signal. After the falling edge is detected, resampling is performed 0.5 Tbits later when the number of the sampling per 1 Tbit is even and {(the number of the sampling + 1) / 2} / (the number of sampling) Tbits later when the number is odd. If the synchronized RLIN3nRX signal is low level, the bit is recognized as a start bit. The bit is not recognized as a start bit if the RLIN3nRX signal is fixed at low level after the reset is cleared or if a high level is detected during the resampling.

After the start bit is detected, 1 bit is sampled per Tbit.

However, when the BERE bit in the RLN3nLEDE register is 1, the sampling point is the same as the bit error detection timing.

The LIN/UART interface has a noise filter function with respect to received data. If the LRDNFS bit in the RLN3nLMD register is 0, the noise filter is used. For a sampling value, the value determined by a 3-sampling majority rule by the prescaler clock is used. If the LRDNFS bit in the RLN3nLMD register is 1, the noise filter is not used. In this case, for a sampling value, the synchronized RLIN3nRX value at the sampling position is used as is.

Figure 20.21 shows an example of data reception timing.

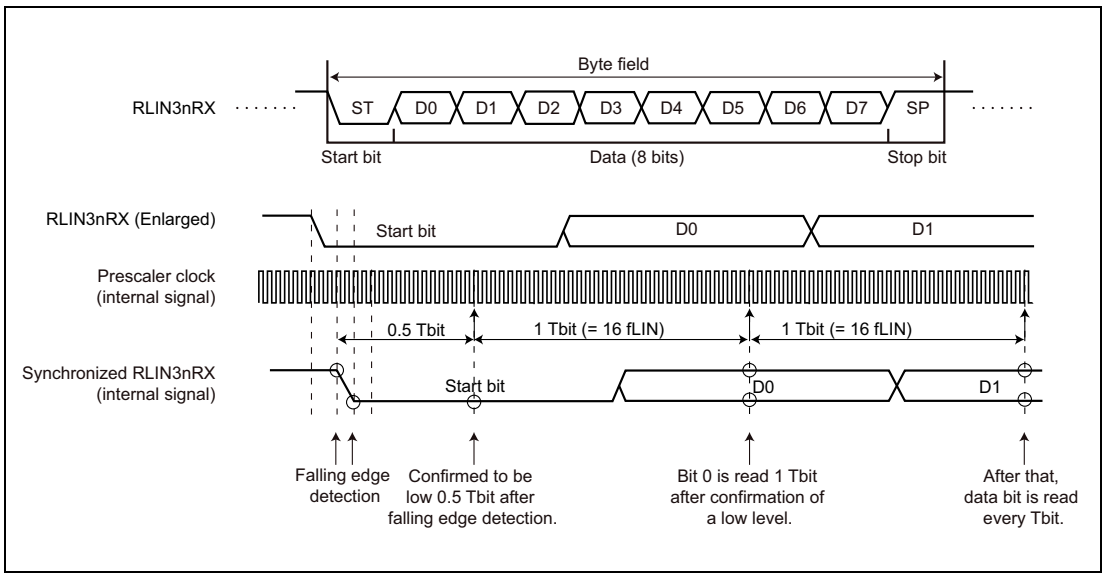


Figure 20.21 Example of Data Reception Timing (When Sampling Count is 16 in 1 Tbit)

20.8.3 Expansion Bits

The LIN/UART interface (in UART mode) can transmit and receive 9-bit long data by setting the UEBE bit in the RLN3nLUOR1 register to 1.

20.8.3.1 Expansion Bit Transmission

The LIN/UART interface (in UART mode) can transmit 9-bit long data when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1 and by writing the 9-bit data to either the UART transmission data register (RLN3nLUTDR) or the UART wait transmission data register (RLN3nLUWTDR).

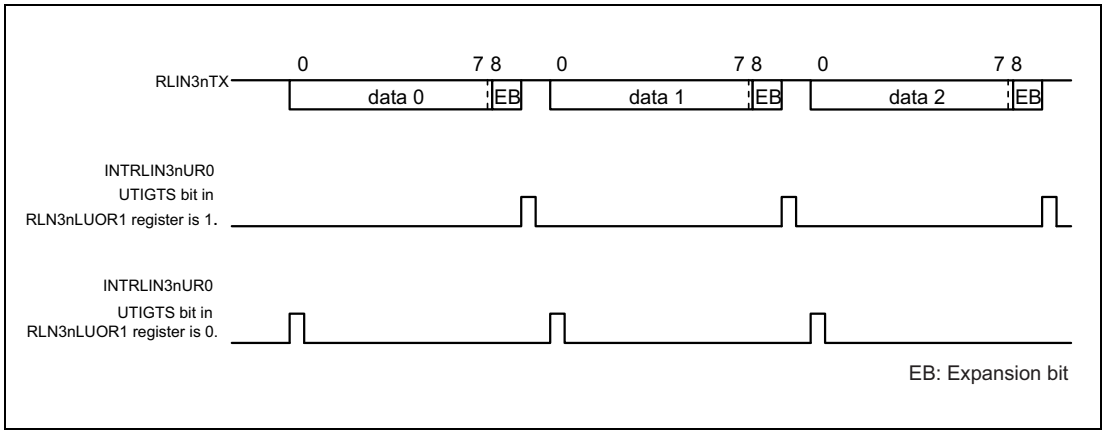


Figure 20.22 Transmission Example When Expansion Bit is Enabled (LSB First)

20.8.3.2 Expansion Bit Reception

With the LIN/UART interface (in UART mode), 9-bit data can always be received without requiring a comparison of expansion bits, provided that the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1, the expansion bit comparison disable bit (UECD) is 1, and the expansion bit data comparison enable bit (UEBDCE) is 0. Irrespective of the particular setting of the expansion bit detection level select bit (UEBDL) in the UART option register 1 (RLN3nLUOR1), a successful RLIN3n reception interrupt is generated when 9-bit data is received.

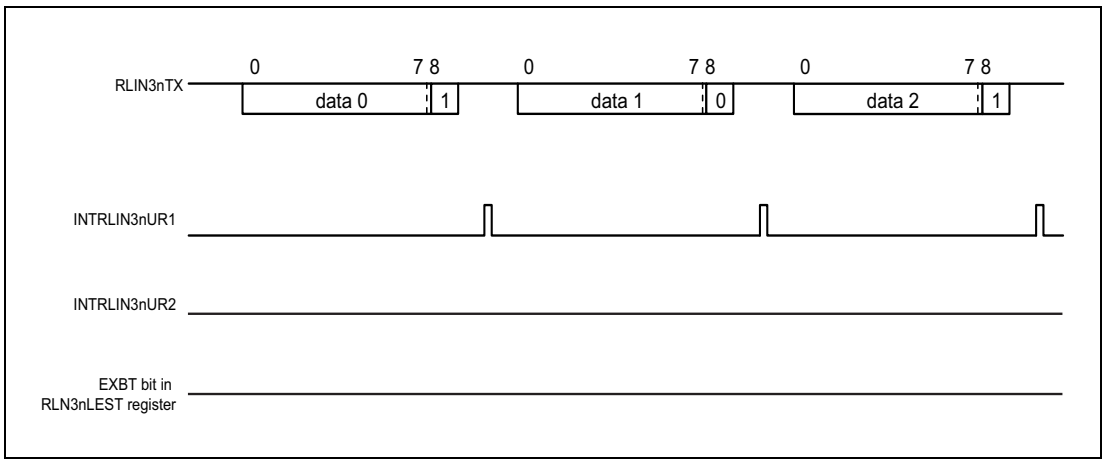


Figure 20.23 Expansion Bit Reception Example (LSB First)

20.8.3.3 Expansion Bit Reception (with Expansion Bit Comparison)

The LIN/UART interface (in UART mode) can compare received expansion bits and the UEBDL bits when the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1 and the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 0.

If the level that was set in the expansion bit detection level select bit (UEBDL) is detected, an RLIN3n status interrupt request is generated upon completion of data reception, and the expansion bit detection flag (EXBT) in the LIN error status register (RLN3nLEST) is set. If the reversed value of an expansion bit detection level is detected, a successful RLIN3n reception interrupt request is generated. In either case, the received data is stored in the UART reception data register (RLN3nLURDR), unless there was an overrun error.

Figure 20.24 shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.

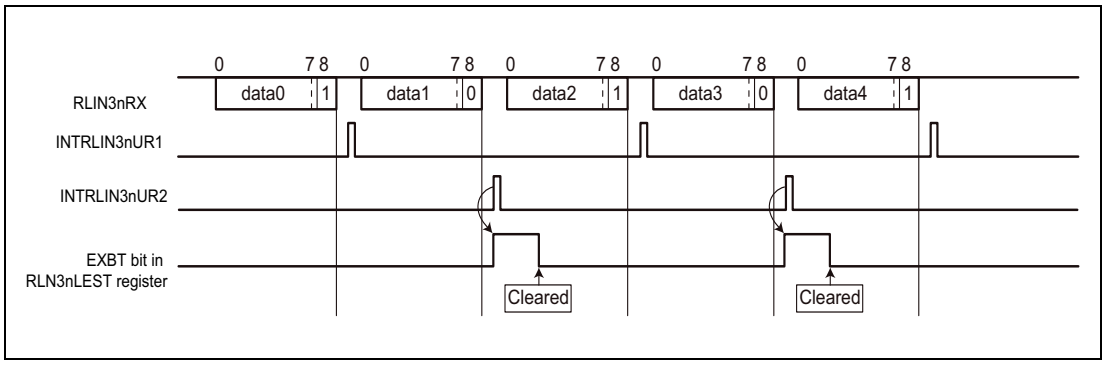


Figure 20.24 Expansion Bit Reception Example (with Expansion Bit Comparison) (LSB First, UEBDL = 0)

NOTE

- If a reception error (parity error, framing error, or overrun error) occurs in received data 0, 2, or 4 (if a reversed value of an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. In this case, a successful RLIN3n reception interrupt is not generated.
- If a reception error (parity error, framing error, or overrun error) occurs in received data 1 or 3 (if an expansion bit detection level is detected), an RLIN3n status interrupt is generated, and the error flag is updated. If the overrun error occurs, the expansion bit detection flag (EXBT) is also set.

20.8.3.4 Expansion Bit Reception (with Data Comparison)

If the expansion bit enable bit (UEBE) in the UART option register 1 (RLN3nLUOR1) is 1 and the expansion bit comparison disable bit (UECD) is 0 and the expansion bit/data comparison enable bit (UEBDCE) is 1, when the level that was set by the expansion bit detection level select bit (UEBDL) is detected, the LIN/UART interface (in UART mode) compares the 8 bits of the received data excluding the expansion bit, exclusive of the expansion bit in the received data, with the a preset RLN3nLIDB register value.

If the result of the comparison is a match, the LIN/UART interface performs the following operations:

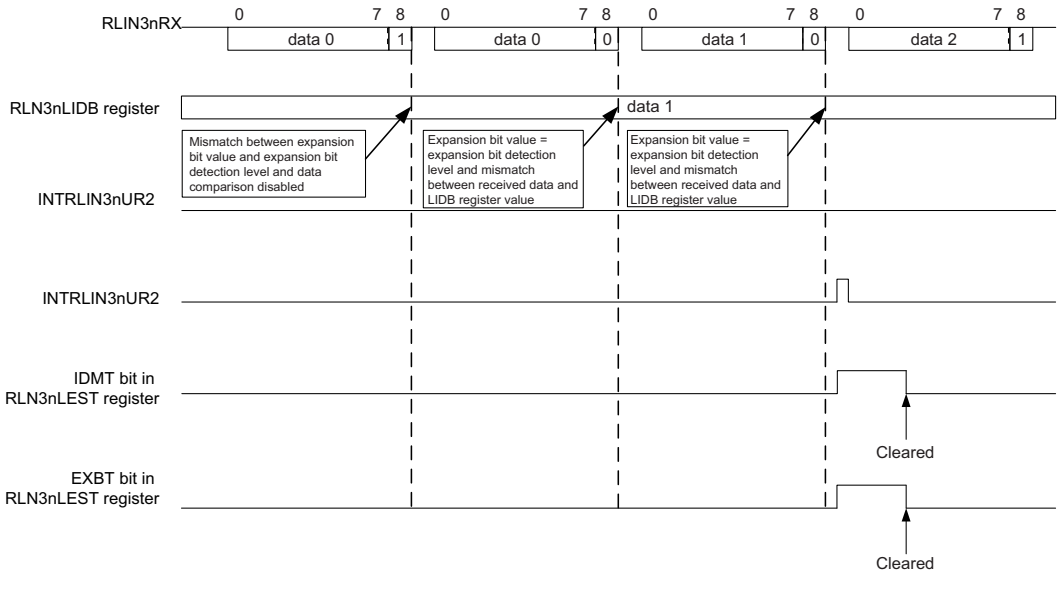
- Generates an RLIN3n status interrupt
- Sets an expansion bit detection flag (EXBT)
- Sets an ID match flag (IDMT)
- Stores the received data in the UART reception data register (RLN3nLURDR)

Even when the result of the comparison is a match, a successful RLIN3n reception interrupt is not generated.

If the result of the comparison is not a match, no RLIN3n successful reception interrupt or RLIN3n status interrupt is generated, and the EXBT and IDMT flags are not set to 1. The received data is not stored in the UART reception data register (RLN3nLURDR).

When changing the UEBDCE bit to 0, make the change before the reception of another set of data is finished.

Figure 20.25 shows an example when the expansion bit detection level select bit (UEBDL) is set to 0.



Note: If a reception error (parity error or overrun error) occurs, an RLIN3n status interrupt is generated and the error flag is updated. If the overrun error occurs and the results of the comparison is a match, the EXBT and IDMT are also set to 1.

Figure 20.25 Expansion Bit Reception Example (with Data Comparison) (LSB First, UEBDL = 0)

20.8.4 Status

In UART mode, the LIN/UART interface can detect five types of statuses.

Two statuses, successful UART buffer transmission and error detection, can generate interrupt requests.

Table 20.68 shows the types of statuses available in UART mode.

Table 20.68 Types of Statuses in UART Mode

Status	Status Set Condition	Status Clear Condition	Corresponding Bit	Interrupt
Reset	After the OM0 bit in the RLN3nLCUC register is set to not-LIN–reset-mode, if actually the LIN/UART interface is cleared from LIN reset mode.	After the OM0 bit in the RLN3nLCUC register is set to LIN reset mode, if actually the LIN/UART interface enters LIN reset mode.	OMM0 bit in RLN3nLMST register	—
Successful UART buffer transmission	• When the UTIGTS bit in the RLN3nLUOR1 register is 0 (transmission interrupt request is generated upon start of transmission), the transmission of the last data of the data length set by the MDL bit in the RLN3nLDFC register is started. • When the UTIGTS bit in the RLN3nLUOR1 register is 1 (transmission interrupt request is generated upon end of transmission), the transmission of the data length set by the MDL bit in the RLN3nLDFC register is ended.	• When cleared by software • After transition to LIN reset mode	FTC flag in RLN3nLST register	√
Error detection	If any of the UPER flag, IDMT flag, EXBT flag, FER flag, OER flag, and BER flags in the RLN3nLEST register turns 1 (error detected).	• When cleared by software*1 • After transition to LIN reset mode	ERR flag in RLN3nLST register	√
Transmission status	• When data is written to the RLN3nLUTDR or RLN3nLUWTDR register. • When a 1 is written to the RTS bit in the RLN3nLTRC register.	• The transmission of the data set in the RLN3nLUTDR or RLN3nLUWTDR register is complete, but another transmission data item is not set • The transmission of the data in the UART buffer is complete, and the RTS bit in the RLN3nLTRC register is cleared • After transition to LIN reset mode	UTS flag in RLN3nLST register	—
Reception status	• When a start bit is detected.	• When a sampling point for stop bits is detected • After transition to LIN reset mode	URS flag in RLN3nLST register	—

Note 1. Writing a 0 to the UPER, IDMT, EXBT, FER, OER, and BER flags in the RLN3nLEST register when the LIN reset mode is being canceled turns the ERR flag in the RLN3nLST register to 0.

20.8.5 Error Status

Types of Error Statuses

In UART mode, the LIN/UART interface can detect four types of errors and two types of statuses. The condition of these statuses can be checked by means of the corresponding bits in the RLN3nLEST register.

Table 20.69 lists applicable status types.

Table 20.69 Types of Statuses in UART Mode

Status	Error Detection Condition	Communication	Enable/Disable Detection	Corresponding Bit
Bit error	The transmitted data and the data monitored on the receive pin do not match*1	Continues until the transmission of the set transmission data is finished.	Enabled	BER flag in RLN3nLEST register
Overrun error	After received data is stored in the RLN3nLURDR register, another data item is received before the data is read. (In this case, no data is stored in the RLN3nLURDR register).	— (Reception is finished by the time this error is detected)	Enabled	OER flag in RLN3nLEST register
Framing error	When the first stop bit is low level in the reception processing.	— (Reception is finished by the time this error is detected)	Enabled	FER flag in RLN3nLEST register
Parity error	The received parity value fails to match the parity value calculated from the received data	Continues until the data reception is finished.	Disabled *2	UPER flag in RLN3nLEST register
Expansion bit detection	The value of the received expansion bit matches the value of the UEBDL bit in the RLN3nLUOR1 register.	—	Enabled	EXBT flag in RLN3nLEST register
ID match detection	The value of the received expansion bit matches the value of the UEBDL bit in the RLN3nLUOR1 register and the 8-bit receive data excluding the expansion bit matches the value of the RLN3nLIDB register.	—	Enabled	IDMT flag in RLN3nLEST register

Note 1. In the case of transmission from the UART buffer, bit errors are detected even in the space between UART frames (inter-byte space).

Note 2. Setting the UPS[1:0] bits in the RLN3nLBFC register to 10_B (0 parity) disables the checking of parity bit values. In this case, no parity error is generated.

The error status is cleared by software or at a transition to LIN reset mode.

20.9 LIN Self-Test Mode

The LIN/UART interface provides a LIN self-test mode. When the LIN/UART interface enters the LIN self-test mode, RLIN3nTX and RLIN3nRX are disconnected from external pins and they are internally connected within the LIN/UART interface. Therefore, the frame transmitted from RLIN3nTX is looped back to RLIN3nRX. The LIN self-test mode can perform tests exclusively in LIN mode.

The self-test can be performed in the following two types.

- LIN master self-test mode (transmission): Header transmission and response transmission
- LIN master self-test mode (reception): Header transmission and response reception

In LIN self-test mode, the operate is at the fastest baud rate, regardless of the setting of the baud rate generator.

Regardless of the setting of the baud rate related registers, the baud rate operates at the LIN communication clock source/16 [bps]. (The NSPB[3:0] bits in the RLN3nLWBR register should be set to 0000_B or 1111_B.)

In addition, in LIN self-test mode, the following functions are not supported.

- LIN wake-up mode
- Frame separate mode
- Multi-byte response transmission/reception
- Frame/response timeout error

Do not use these functions.

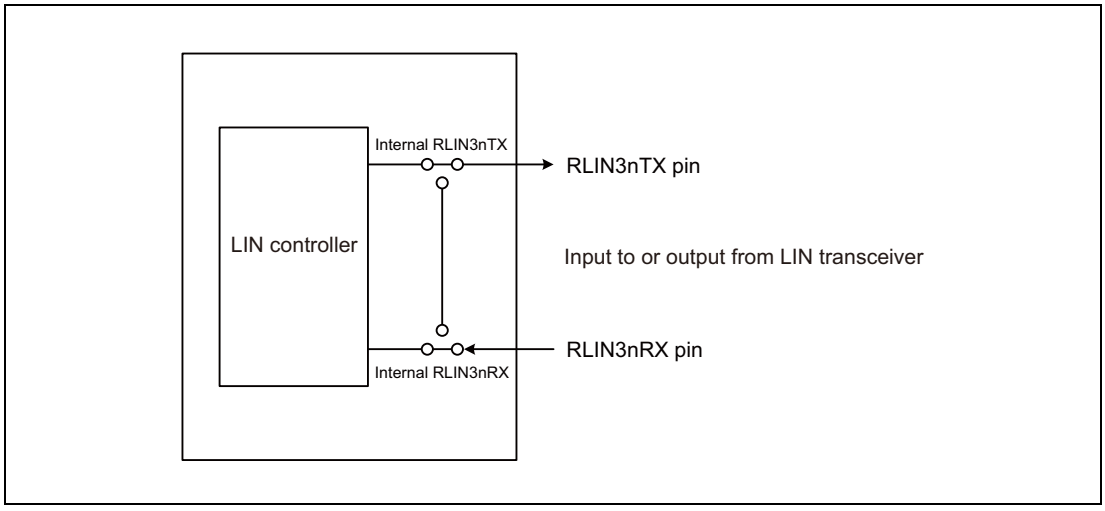


Figure 20.26 Connection in LIN Reset Mode, LIN Mode, and UART Mode

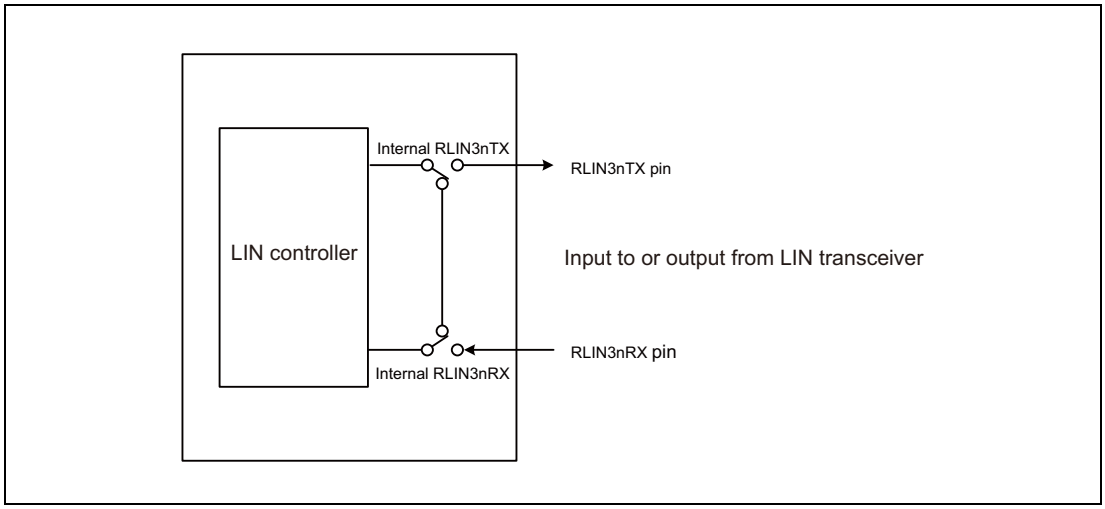


Figure 20.27 Connection in LIN Self-Test Mode

20.9.1 Change to LIN Self-Test Mode

Writing to the RLN3nLSTC register makes a transition to the LIN self-test mode.
When the LSTM bit in the RLN3nLSTC register is set to 1, the shift to the LIN self-test mode is checked.

When changing to LIN self-test mode, be sure to execute a specific sequence. In that sequence, information must be written three times consecutively to the LIN self-test control register, as follows:

- Change to LIN reset mode
  Set the OM0 bit in the RLN3nLCUC register to 0 (LIN reset mode).
  Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).
- Select a LIN mode
  LMD[1:0] bits in RLN3nLMD = 00_B (LIN master mode)
- 1st write: RLN3nLSTC register = $1010\ 0111_B$ ($A7_H$)
- 2nd write: RLN3nLSTC register = $0101\ 1000_B$ (58_H)
- 3rd write: RLN3nLSTC register = $0000\ 0001_B$ (01_H)
- Verify the transition to LIN self-test mode
  Read the LSTM bit in the RLN3nLSTC register; verify that it is 1 (LIN self-test mode).

If the key of the first write ($A7_H$) is written twice by mistake, the transition to LIN self-test mode is canceled. The above sequence should be retried from the step of first write. In addition, if a write to another LIN-related register is performed during transition to LIN self-test mode (three consecutive write operations to the RLN3nLSTC register), the transition is also canceled.

20.9.2 Transmission in LIN Master Self-Test Mode

To execute a self-test on LIN master transmission, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = 0000 $xxxx_B$*1
  RLN3nLBRP0 register = xxxx $xxxx_B$*1
  RLN3nLBRP1 register = xxxx $xxxx_B$*1
  RLN3nLMD register = 00xx $xx00_B$*1
- Set the interrupt enable and error enable related registers.
  RLN3nLIE register = 0000 $xxxx_B$*2
  RLN3nLEDE register = x000 x0xx
- Set the break field and space related registers.
  RLN3nLBFC register = 00xx $xxxx_B$
  RLN3nLSC register = 00xx $0xxx_B$
- Cancel the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Set the transmit frame related registers.
  RLN3nLDFC register = 00x1 $xxxx_B$
  RLN3nLIDB register = xxxx $xxxx_B$
  RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx $xxxx_B$
- Header transmission → response transmission started
  Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started).
  The LIN master self-test mode (transmission) is executed. In this mode, interrupt are generated, and status and error status are also updated. The checksum is automatically calculated by the LIN/UART interface.
  To suspend the LIN master self-test mode (transmission) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the transmission is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is reversed before being stored because the transmitted value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the transmission fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Don't care

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The LPRS[2:0] bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1 register and the LCKS[1:0] bit in the RLN3nLMD register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in Section 7, Interrupt.

Note 3. When the successful header transmission interrupt and the successful frame transmission interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame transmission interrupt, the SHIE bit in the RLN3nLIE register should not be

set to 1 (successful header transmission interrupt enabled). The time required from the set of the successful header transmission flag to the set of the successful frame/wake-up transmission flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = frequency of LIN communication clock source × 16

20.9.3 Reception in LIN Master Self-Test Mode

To execute a self-test on LIN master reception, perform the procedure below:

- Set the baud rate, noise filter, and interrupt output related registers.
  RLN3nLWBR register = 0000 $xxxx_B$*1
  RLN3nLBRP0 register = xxxx $xxxx_B$*1
  RLN3nLBRP1 register = xxxx $xxxx_B$*1
  RLN3nLMD register = 00xx $xx00_B$*1
- Set the interrupt enable and error enable related registers.
  RLN3nLIE register = 0000 $xxxx_B$*2
  RLN3nLEDE register = x000 $x0xx_B$
- Set the break field and space related registers.
  RLN3nLBFC register = 00xx $xxxx_B$
  RLN3nLSC register = 00xx $0xxx_B$*1
- Cancel the LIN reset mode.
  Write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register, and check that the OMM1 and OMM0 bits in the RLN3nLMST register are 11_B.
- Configure the reception frame related registers.
  RLN3nLDFC register = 00x0 $xxxx_B$
  RLN3nLIDB register = xxxx $xxxx_B$
  RLN3nLDRB1 to RLN3nLDRB8 registers = xxxx $xxxx_B$
  RLN3nCBR register = xxxx $xxxx_B$
  Since the checksum value to be transmitted is not automatically calculated, perform the calculation value in the RLN3nLCBR register.
- Header transmission → response reception started
  Set the FTS bit in the RLN3nLTRC register to 1 (frame transmission or wake-up transmission/reception started).
  The LIN master self-test mode (reception) is executed. In this mode, interrupt are generated, and status and error status are also updated.
  To suspend the LIN master self-test mode (reception) being executed, write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register for transition to LIN reset mode.
- When the reception is completed, the reversed value of the looped-back frame data is stored in the RLN3nLIDB, RLN3nLDBRb (b = 1 to 8), and RLN3nLCBR registers (the data is reversed before being stored because the set value should be compared with the looped-back value). Then, the FTS bit in the RLN3nLTRC register is cleared.
- If the reception fails to complete due to an error, the applicable error flag is set and the FTS bit in the RLN3nLTRC register is cleared.

Note: x: Don't care

Note 1. The following register settings are not reflected to the operation of the LIN self-test mode. The LPRS[2:0] bit in the RLN3nLWBR register, the RLN3nLBRP0 register, the RLN3nLBRP1

register, the LCKS[1:0] bit in the RLN3nLMD register, and the IBS[1:0] bit in the RLN3nLSC register. Therefore, those settings are not necessary.

Note 2. If necessary, set the related registers described in Section 7, Interrupt.

Note 3. When the successful header transmission interrupt and the successful frame reception interrupt are used in the same interrupt processing, if the software processing of the successful header transmission interrupt is not completed before the generation of the successful frame reception interrupt, the SHIE bit in the RLN3nLIE register should not be set to 1 (successful header transmission interrupt enabled).
The time required from the set of the successful header transmission flag to the set of the successful frame/wake-up reception flag is calculated by the following formula.

10 × (number of data bytes + 1) [Tbit]

1 Tbit = 1/LIN communication clock source × 16

20.9.4 Terminating LIN Self-Test Mode

To terminate LIN self-test mode, perform the procedure below:

- Write 0 (LIN reset mode) to the OM0 bit in the RLN3nLCUC register.
  If the OMM1 and OMM0 bits in the RLN3nLMST register are not 11_B, write 11_B to the OM1 and OM0 bits in the RLN3nLCUC register. After confirming that the OMM1 and OMM0 bits in the RLN3nLMST register have turned 11_B, change to LIN reset mode.
- Verify the cancelation of LIN self-test mode.
  Read the LSTM bit in the RLN3nLSTC register; confirm that it is not 0 (not in LIN self-test mode)
- Verify the transition to LIN reset mode.
  Read the OMM0 bit in the RLN3nLMST register; verify that it is 0 (LIN reset mode).

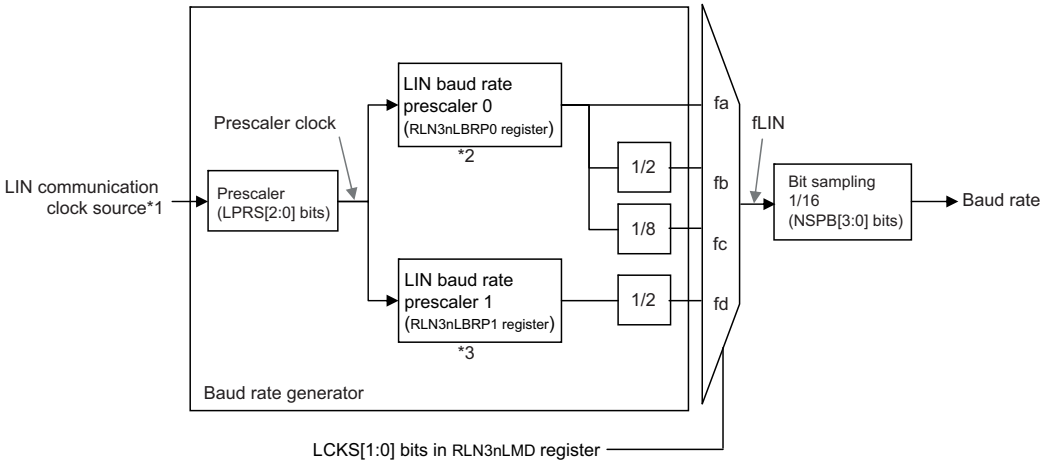
20.10 Baud Rate Generator

The prescaler clock is obtained by frequency-dividing the LIN communication clock source by the prescaler, and the LIN system clock (fLIN) is obtained by frequency-dividing the prescaler clock by the baud rate generator. The clock obtained by frequency-dividing the LIN system clock (fLIN) by the number of samples is the baud rate. The inverse of this baud rate is called the bit time (Tbit).

The LIN/UART interface has two kinds of baud rate generators. The baud rate generators switch over according to the mode used.

20.10.1 LIN Master Mode

Figure 20.28 shows a block diagram of baud rate generation in LIN master mode.



Note 1. For the LIN communication clock source, see Section 12, Clock Controller.
Note 2. When the value in RLN3nLBRP0 register is N (N = 0 to 255), the clock frequency is divided by N + 1.
Note 3. When the value in RLN3nLBRP1 register is M (M = 0 to 255), the clock frequency is divided by M + 1.

Figure 20.28 Block Diagram of Baud Rate Generation in LIN Master Mode

By setting the RLN3nLBRP0 register so that fa is 307200 Hz (= 19200 × 16), the resulting bit rates are fa = 19200 × 16, fb = 9600 × 16, and fc = 2400 × 16. These bit rates are frequency-divided by 16 in the bit timing generator, enabling baud rates of 19200 bps, 9600 bps and 2400 bps, to be generated. Also, by setting the RLN3nLBRP1 register so that fd is 166672 Hz (= 10417 × 16), the resulting bit rate is fd = 10417 × 16. This bit rate is frequency-divided by 16 in the bit timing generator, enabling 10417 bps to be generated.

Baud rate of LIN master

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)
÷ (RLN3nLBRP0+ 1) ÷ 16 [bps] (When fa is selected for fLIN)

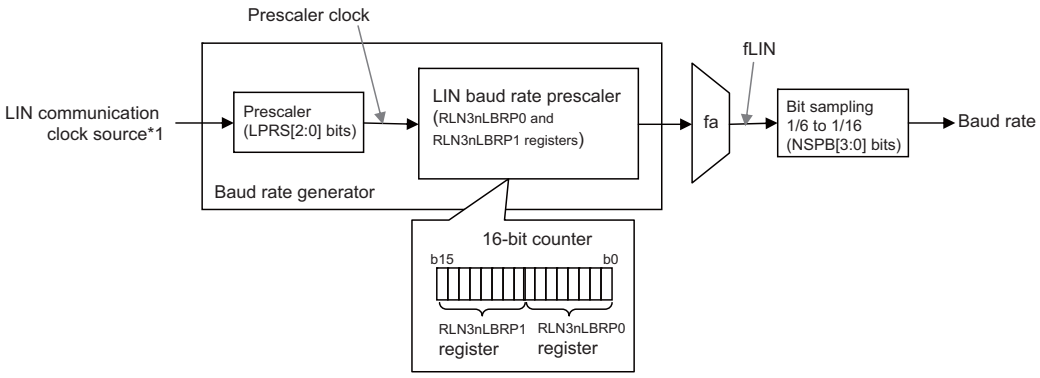
= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)
÷ (RLN3nLBRP0 + 1) ÷ 2 ÷ 16 [bps] (When fb is selected for fLIN)

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)
÷ (RLN3nLBRP0 + 1) ÷ 8 ÷ 16 [bps] (When fc is selected for fLLIN)

= {Frequency of LIN communication clock source} × (RLN3nLWBR.LPRS[2:0] selection clock)
÷ (RLN3nLBRP1 + 1) ÷ 2 ÷ 16 [bps] (When fd is selected for fLIN)

20.10.2 UART Mode

Figure 20.29 shows a block diagram of baud rate generation in UART mode.



Note 1. For the LIN communication clock source, see Section 12, Clock Controller.

Figure 20.29 Block Diagram of Baud Rate Generation in UART Mode

UART baud rate is calculated with the following formula:

UART baud rate
= {LIN communication clock source frequency} × (RLN3nLWBR.LPRS[2:0] select clock) ÷ (RLN3nLBRP01 + 1) ÷ {RLN3nLWBR.NSPB[3:0] select count} [bps]

20.11 Noise Filter

The LIN/UART interface has a noise filter for reducing erroneous receiving of data due to noise. By setting the LRDNFS bit in the RLN3nLMD register to 0 (to use the noise filter), the noise filter is activated. The noise filter samples the level of the synchronized RLIN3nRX with the prescaler clock, and outputs the sampling value determined by a 3-sampling majority rule. The value of each bit of the receive data is determined based on the noise filter output.

Figure 20.30 shows the configuration of the noise filter, **Figure 20.31** shows an example of a noise filter circuit, and **Figure 20.32** shows the determination of the received data when the noise filter is used.

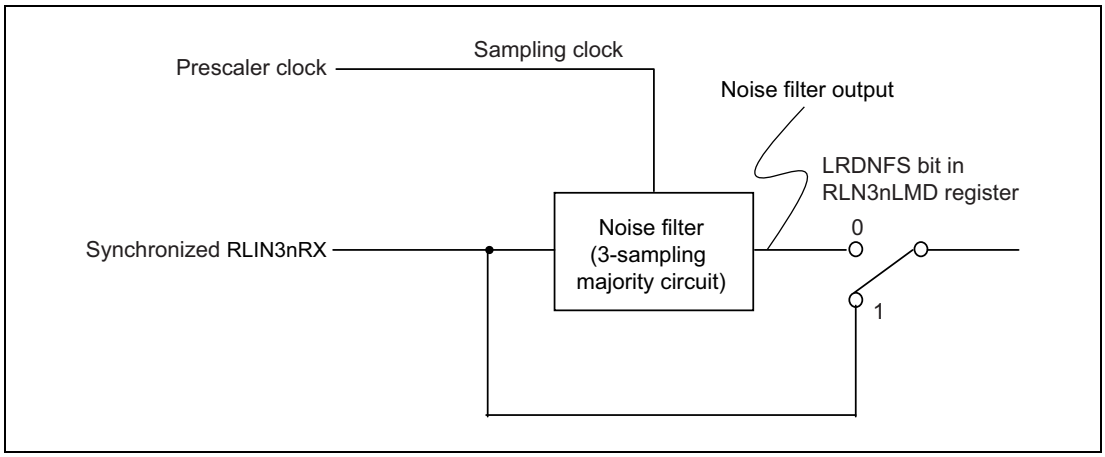


Figure 20.30 Configuration of Noise Filter

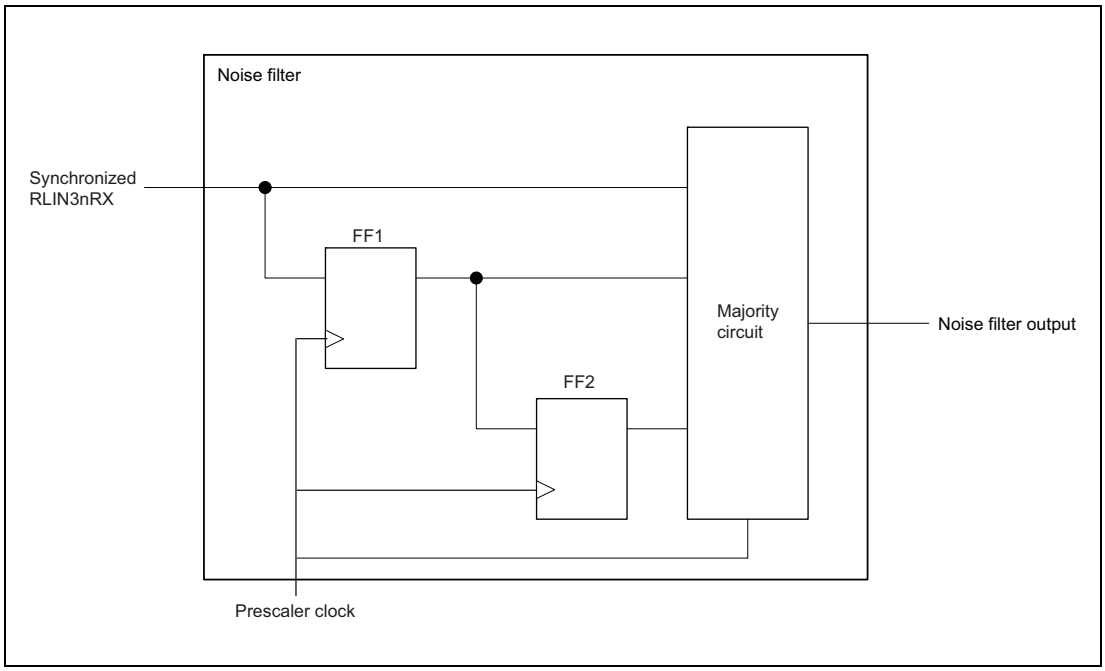


Figure 20.31 Example of Noise Filter Circuit

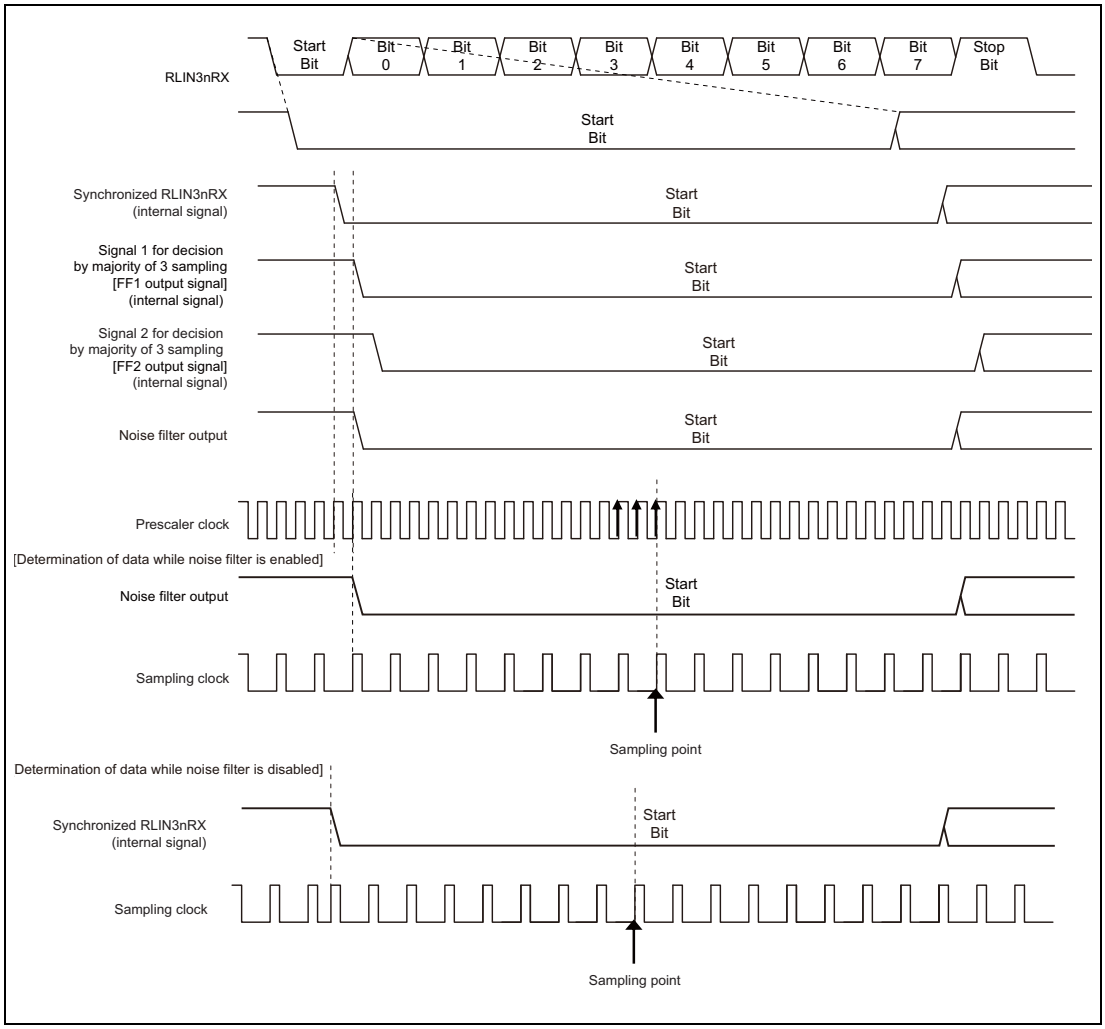


Figure 20.32 Determination of Received Data when Noise Filter is Used

Section 21 I2C Bus Interface (RIIC)

This section contains a generic description of the I2C Bus Interface (RIIC).

The first part of this section describes all RH850/D1L/D1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RIIC.

21.1 Features of RH850/D1L/D1M RIIC

21.1.1 Number of Units and Channels

This microcontroller has the following number of RIIC units.

Each RIIC unit has one channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 21.1 Number of Units

Product Name	All products
Number of Units	2
Name	RIICn (n = 0, 1)

Table 21.2 Index

Index	Meaning
n	Throughout this section, the individual RIIC units are identified by the index “n” (n = 0, 1): for example, RIICnCR1 is the I2C bus control register1.

21.1.2 Register Base Address

RIIC base addresses are listed in the following table.

RIIC register addresses are given as offsets from the base addresses in general.

Table 21.3 Register Base Address

Base Address Name	Base Address
<RIIC0_base>	FFDB 0000_H
<RIIC1_base>	FFDB 1000_H

21.1.3 Clock Supply

The RIIC clock supply is shown in the following table.

Table 21.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RIICn	PCLK	Clock Controller CLKJIT

Note 1. Set the period of PCLK no greater than 1/2 of the width at high level of the SCL clock.

21.1.4 Interrupt Requests

RIIC interrupt requests are listed in the following table.

Table 21.5 Interrupt Requests

RIICn signals	Function	Connected to
RIIC0		
INTIICnEE	Transfer Error/Event Generation	Interrupt Controller INTRIIC0LEE
INTIICnRI	Receive complete	Interrupt Controller INTRIIC0RI DMA Controller trigger ID 110
INTIICnTI	Transmit data empty	Interrupt Controller INTRIIC0TI DMA Controller trigger ID 109
INTIICnTEI	Transmit end	Interrupt Controller INTRIIC0TEI
RIIC1		
INTIICnEE	Transfer Error/Event Generation	Interrupt Controller INTRIIC1LEE
INTIICnRI	Receive complete	Interrupt Controller INTRIIC1RI DMA Controller trigger ID 112
INTIICnTI	Transmit data empty	Interrupt Controller INTRIIC1TI DMA Controller trigger ID 111
INTIICnTEI	Transmit end	Interrupt Controller INTRIIC1TEI

21.1.5 Reset Sources

RIIC reset sources are listed in the following table. RIIC is initialized by these reset sources.

Table 21.6 Reset Sources

Unit Name	Reset Source
RIICn	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

21.1.6 External Input/Output Signals

External input/output signals of RIIC are listed below.

Table 21.7 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
RIIC0		
RIICnSCL	Serial clock I/O pin	RIIC0SCL
RIICnSDA	Serial data I/O pin	RIIC0SDA
RIIC1		
RIICnSCL	Serial clock I/O pin	RIIC1SCL
RIICnSDA	Serial data I/O pin	RIIC1SDA

When using these ports, the PBDCn register for the corresponding port and the corresponding bit in the PODCn register must be set to 1.

21.2 Overview

21.2.1 Functional Overview

Communications format

- I2C bus format
- Master mode or slave mode selectable
- Automatic securing of the various set-up times, hold times, and bus-free times for the transfer rate

Transfer rate

Up to 400 kbps

SCL clock

- For master operation, the duty cycle of the SCL clock is selectable in the following range:
  – 0%< Duty < 100%

Issuing and detecting conditions

Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.

Slave address

- Up to three slave-address settings can be made.
- Seven- and ten-bit address formats are supported (along with the use of both at once).
- General call addresses and device ID addresses are detectable.

Acknowledgement

- For transmission, the acknowledge bit is automatically loaded
  - Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge bit.
- For reception, the acknowledge bit is automatically transmitted
  - If a wait between the eighth and ninth clock cycles has been selected, software control of the value in the acknowledge field in response to the received value is possible.

Wait function

- In reception, the following periods of waiting can be obtained by holding the clock signal (SCL) at the low level:
  - Waiting between the eighth and ninth clock cycles
  - Waiting between the ninth clock cycle and the first clock cycle of the next transfer (WAIT function)

SDA output delay function

Timing of the output of transmitted data, including the acknowledge bit, can be delayed.

Arbitration

- For multi-master operation
  - Operation to synchronize the SCL (clock) signal in cases of conflict with the SCL signal from another master is possible.
  - When issuing the start condition would create conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line.
  - In master operation, loss of arbitration is detected by testing for non-matching of internal and line levels for transmit data.
- Loss of arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the issuing of double start conditions).
- Loss of arbitration in transfer of a not-acknowledge bit due to the internal signal for the SDA line and the level on the SDA line not matching is detectable.
- Loss of arbitration due to non-matching of internal and line levels for data is detectable in slave transmission.

Timeout function

The internal time-out function is capable of detecting long-interval stop of the SCL (clock signal).

Noise removal

The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software.

Interrupt sources

- Four sources:
  - Error in transfer or occurrence of events (detection of arbitration lost, NACK, time-out, a start condition including a restart condition, or a stop condition)
  - Receive complete (including matching with a slave address)
  - Transmit data empty (including matching with a slave address)
  - Transmission end

21.2.2 Block Diagram

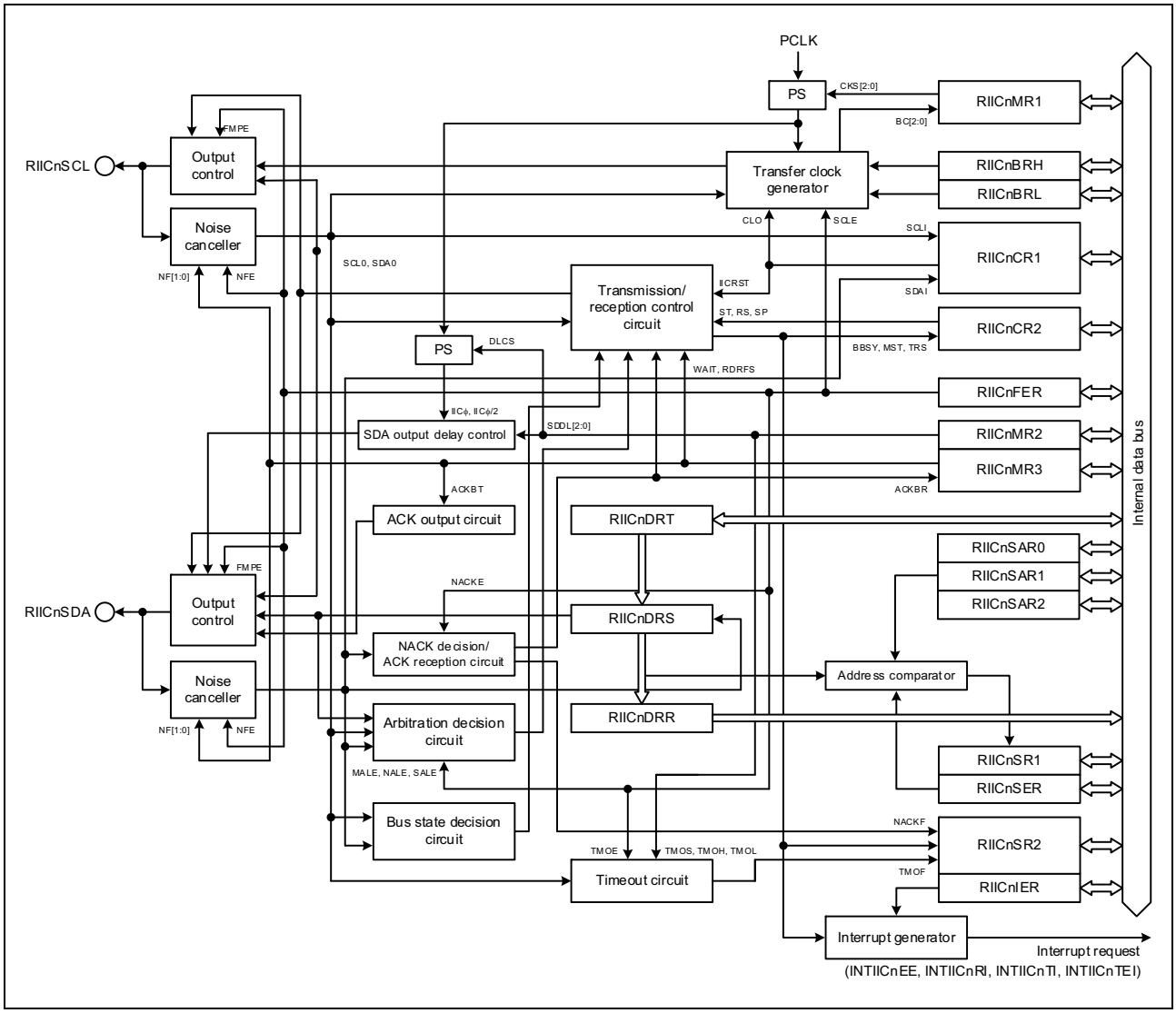


Figure 21.1 Block Diagram of RIIC

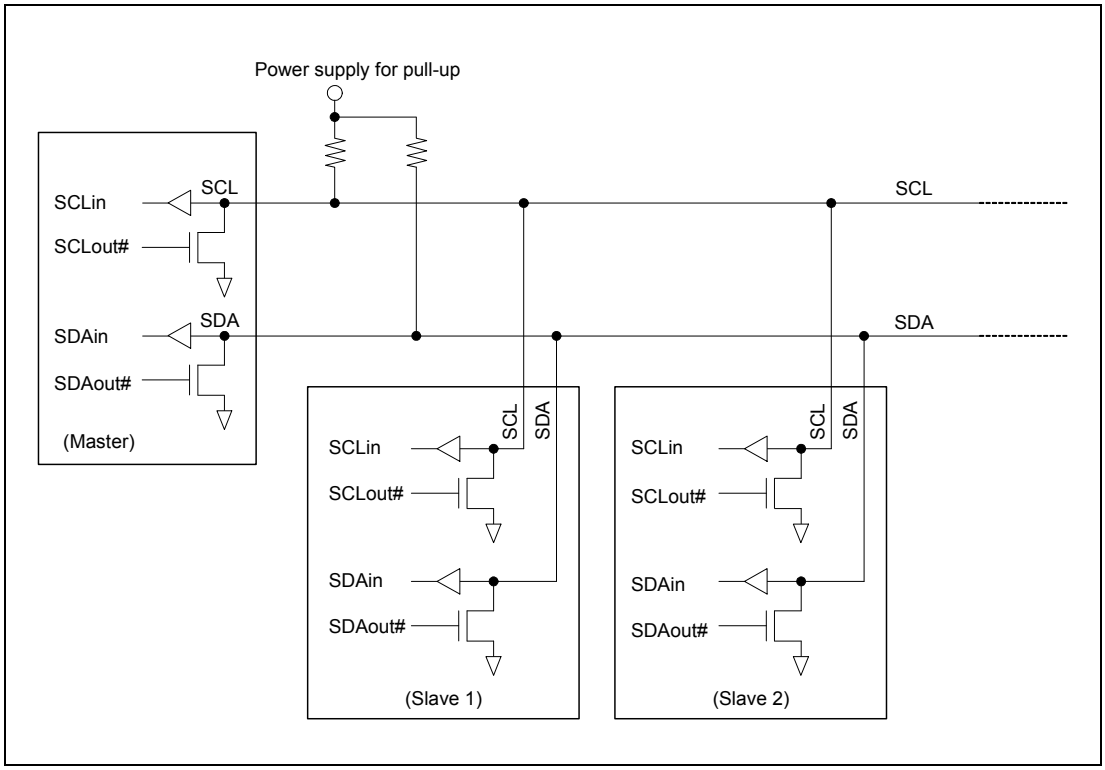


Figure 21.2 Connections to the External Circuit by the I/O Pins (I2C Bus Configuration Example)

21.3 Registers

21.3.1 List of Registers

RIIC registers are listed in the table below.

For details about <RIICn_base>, see Section 21.1.2, Register Base Address.

Table 21.8 List of Registers

Module Name	Register Name	Symbol	Address
RIICn	I2C Bus Control Register 1	RIICnCR1	<RIICn_base> + 0000_H
RIICn	I2C Bus Control Register 2	RIICnCR2	<RIICn_base> + 0004_H
RIICn	I2C Bus Mode Register 1	RIICnMR1	<RIICn_base> + 0008_H
RIICn	I2C Bus Mode Register 2	RIICnMR2	<RIICn_base> + $000C_H$
RIICn	I2C Bus Mode Register 3	RIICnMR3	<RIICn_base> + 0010_H
RIICn	I2C Bus Function Enable Register	RIICnFER	<RIICn_base> + 0014_H
RIICn	I2C Bus Status Enable Register	RIICnSER	<RIICn_base> + 0018_H
RIICn	I2C Bus Interrupt Enable Register	RIICnIER	<RIICn_base> + $001C_H$
RIICn	I2C Bus Status Register 1	RIICnSR1	<RIICn_base> + 0020_H
RIICn	I2C Bus Status Register 2	RIICnSR2	<RIICn_base> + 0024_H
RIICn	I2C Slave Address Register 0	RIICnSAR0	<RIICn_base> + 0028_H
RIICn	I2C Slave Address Register 1	RIICnSAR1	<RIICn_base> + $002C_H$
RIICn	I2C Slave Address Register 2	RIICnSAR2	<RIICn_base> + 0030_H
RIICn	I2C Bus Bit Rate Low-Level Register	RIICnBRL	<RIICn_base> + 0034_H
RIICn	I2C Bus Bit Rate High-Level Register	RIICnBRH	<RIICn_base> + 0038_H
RIICn	I2C Bus Transmit Data Register	RIICnDRT	<RIICn_base> + $003C_H$
RIICn	I2C Bus Receive Data Register	RIICnDRR	<RIICn_base> + 0040_H
RIICn	I2C Bus Shift Register	RIICnDRS	—

21.3.2 RIICnCR1 — I2C Bus Control Register 1

Access: RIICnCR1 is a 32-bit readable/writable register.
RIICnCR1L and RIICnCR1H are 16-bit readable/writable registers.
RIICnCR1LL, RIICnCR1LH, RIICnCR1HL, and RIICnCR1HH are 8-bit readable/writable registers.

Address: RIICnCR1: <RIICn_base> + 0000_H

RIICnCR1L: <RIICn_base> + 0000_H, RIICnCR1H: <RIICn_base> + 0002_H

RIICnCR1LL: <RIICn_base> + 0000_H, RIICnCR1LH: <RIICn_base> + 0001_H,
RIICnCR1HL: <RIICn_base> + 0002_H, RIICnCR1HH: <RIICn_base> + 0003_H

Value after reset: 0000 $001F_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 21.9 RIICnCR1 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
7	ICE	I2C Bus Interface Enable 0: Disabled (the RIICnSCL and RIICnSDA pins are not driven). 1: Enabled (the RIICnSCL and RIICnSDA pins driven). (This bit selects an RIIC reset or internal reset in combination with the IICRST bit.)
6	IICRST	I2C Bus Interface Internal Reset 0: Clears the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL/SDA output latch)
5	CLO	Extra SCL Clock Cycle Output 0: Does not output an extra SCL clock cycle (default). 1: Outputs an extra SCL clock cycle. (The CLO bit is cleared automatically after one clock cycle is output.)
4	SOWP	SCLO/SDAO Write Protect 0: Bits SCLO and SDAO can be written. 1: Bits SCLO and SDAO are protected. (This bit is read as 1.)
3	SCLO	SCL Output Control/Monitor • Read: 0: The RIIC has driven the RIICnSCL pin low. 1: The RIIC has released the RIICnSCL pin. • Write: 0: The RIIC drives the RIICnSCL pin low. 1: The RIIC releases the RIICnSCL pin.

Table 21.9 RIICnCR1 Register Contents (2/2)

Bit Position	Bit Name	Function
2	SDAO	SDA Output Control/Monitor • Read: 0: The RIIC has driven the RIICnSDA pin low. 1: The RIIC has released the RIICnSDA pin. • Write: 0: The RIIC drives the RIICnSDA pin low. 1: The RIIC releases the RIICnSDA pin.
1	SCLI	SCL Line Monitor 0: RIICnSCL line is low. 1: RIICnSCL line is high.
0	SDAI	SDA Line Monitor 0: RIICnSDA line is low. 1: RIICnSDA line is high.

SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)

These bits are used to directly control the RIICnSDA and RIICnSCL signals output from the RIIC.

When writing to these bits, also write 0 to the SOWP bit.

The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a START condition may be detected and the bus may be released depending on the bit settings.

Do not rewrite these bits during a START condition, STOP condition, repeated START condition, or during transmission or reception. Operation after rewriting under the above conditions is not guaranteed.

When reading these bits, the state of signals output from the RIIC can be read.

CLO Bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.

Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.

For details on this function, see Section 21.13.2, Extra SCL Clock Cycle Output Function.

IICRST Bit (I2C Bus Interface Internal Reset)

This bit is used to reset the internal states of the RIIC.

Setting this bit to 1 initiates an RIIC reset or internal reset.

Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. **Table 21.10** lists the types of RIIC reset.

The RIIC reset resets all registers (except ICE and IICRST) including the RIICnCR2.BBSY flag and internal states of the RIIC, and the internal reset resets the bit counter (RIICnMR1.BC[2:0] bits), the I2C bus shift register (RIICnDRS), and the I2C bus status registers (RIICnSR1 and RIICnSR2) as well as the internal states of the RIIC. For the reset conditions for each register, see Section 21.14, Reset Function of RIIC.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error.

If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the RIICnSCL pin and RIICnSDA pin at a high impedance.

CAUTION

If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If an internal reset is necessary because the RIIC hangs up with the SCL line in a low level output state in slave mode, initiate an internal reset and then issue a restart condition from the master device or resume communication from the start condition issuance after issuing a stop condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 21.10 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers (except ICE and IICRST) and internal states of the RIIC.
	1	Internal reset	Reset the RIICnMR1.BC[2:0] bits, and the RIICnSR1, RIICnSR2, RIICnDRS registers and the internal states of the RIIC.

ICE Bit (I2C Bus Interface Enable)

The ICE bit selects driving or non-driving of the RIICnSCL and RIICnSDA pins. Moreover, this bit can perform two types of reset in combination with the IICRST bit. For the types of reset, see Table 21.10, RIIC Resets.

Set the ICE bit to 1 when using RIIC. Setting the ICE bit to 1 selects driving of the RIICnSCL and RIICnSDA pins.

Set the ICE bit to 0 when RIIC is not to be used. Clearing the ICE bit to 0 stops driving of the RIICnSCL and RIICnSDA pins.

CAUTION

Though the output from RIICnSDA or RIICnSCL is disabled while the ICE bit is 0, the input to RIICnSDA or RIICnSCL is enabled. The RIICnSCL and RIICnSDA pin functions should not be assigned to the RIIC. If assigned, it causes the slave addresses to be compared.

21.3.3 RIICnCR2 — I2C Bus Control Register 2

Access: RIICnCR2 is a 32-bit readable/writable register.
RIICnCR2L and RIICnCR2H are 16-bit readable/writable registers.
RIICnCR2LL, RIICnCR2LH, RIICnCR2HL, and RIICnCR2HH are 8-bit readable/writable registers.

Address: RIICnCR2: <RIICn_base> + 0004_H

RIICnCR2L: <RIICn_base> + 0004_H, RIICnCR2H: <RIICn_base> + 0006_H

RIICnCR2LL: <RIICn_base> + 0004_H, RIICnCR2LH: <RIICn_base> + 0005_H,
RIICnCR2HL: <RIICn_base> + 0006_H, RIICnCR2HH: <RIICn_base> + 0007_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R

Table 21.11 RIICnCR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
7	BBSY	Bus Busy Detection Flag 0: The I2C bus is released (the bus is free). 1: The I2C bus is occupied (the bus is busy).
6	MST*1	Master/Slave Mode 0: Slave mode 1: Master mode
5	TRS*1	Transmit/Receive Mode 0: Receive mode 1: Transmit mode
4	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
3	SP	Stop Condition Issuance Request 0: Does not request to issue a stop condition. 1: Requests to issue a stop condition.
2	RS	Restart Condition Issuance Request 0: Does not request to issue a restart condition. 1: Requests to issue a restart condition.
1	ST	Start Condition Issuance Request 0: Does not request to issue a start condition. 1: Requests to issue a start condition.
0	Reserved	This bit is read as the value after reset. The write value should be the value after reset.

Note 1. When the RIICnMR1.MTWP bit is set to 1, the MST and TRS bits can be written to.

ST Bit (Start Condition Issuance Request)

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free).

For details on the start condition issuance, see Section 21.12, Start Condition/Restart Condition/Stop Condition Issuing Function.

[Setting condition]

When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been issued
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free).
Note that arbitration may be lost as the start condition issuance error if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy).

RS Bit (Restart Condition Issuance Request)

This bit is used to request that a restart condition be issued in master mode.

When this bit is set to 1 to request to issue a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the restart condition issuance, see Section 21.12, Start Condition/Restart Condition/Stop Condition Issuing Function.

[Setting condition]

When 1 is written to the RS bit with the RIICnCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued or a start condition is detected
- When a stop condition is detected
- When the RIICnCR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTIONS

1. Do not set the RS bit to 1 while issuing a stop condition.
2. It is commended to issue a restart condition in master transmit mode. If the RS bit is set to 1 (restart condition issuance request) in mode other than master mode, the restart condition is not issued in this mode but the RS bit remains set. If the operating mode changes to master mode with the bit not being cleared, the restart condition may be issued.

SP Bit (Stop Condition Issuance Request)

This bit is used to request that a stop condition be issued in master mode.

When this bit is set to 1 to request to issue a stop condition, a stop condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the stop condition issuance, see Section 21.12, Start Condition/Restart Condition/Stop Condition Issuing Function.

[Setting condition]

When 1 is written to the SP bit with both the RIICnCR2.BBSY flag and the RIICnCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been issued or a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTIONS

1. Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free).
2. Do not set the SP bit to 1 while a restart condition is being issued.

TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.

The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.

The value of the TRS bit is automatically changed to the value for transmission mode or reception mode (1 or 0) by detection or issuing of a start condition, setting or clearing of the R/W# bit, etc. Although writing to the TRS bit is possible when the RIICnMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When the address received in slave mode matches the address enabled in RIICnSER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the RIICnMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The RIICnSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave transmit mode, a restart condition is detected (a restart condition is detected with RIICnCR2.BBSY = 1 and RIICnCR2.MST = 0)

- When 0 is written to the TRS bit with the RIICnMR1.MTWP bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.

The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.

The value of the MST bit is automatically changed to the value for master mode or slave mode (1 or 0) by detection or issuing of a start condition, etc. Although writing to the MST bit is possible when the RIICnMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the RIICnMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 0 is written to the MST bit with the RIICnMR1.MTWP bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

BBSY Flag (Bus Busy Detection)

The BBSY flag indicates whether the I2C bus is occupied (bus busy) or released (bus free).

This bit is set to 1 when the SDA line changes from high to low under the condition of SCL = high, assuming that a start condition has been issued.

When the SDA line changes from low to high under the condition of SCL = high, this bit is cleared to 0 after the bus free time (specified in RIICnBRL) start condition is not detected, assuming that a stop condition has been issued.

[Setting condition]

When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in RIICnBRL) start condition is not detected after detecting a stop condition
- When 1 is written to the RIICnCR1.IICRST bit with the RIICnCR1.ICE bit set to 0 (RIIC reset)

CAUTIONS

1. **When an internal reset is applied while the bus is free after detection of a stop condition, the setting of the BBSY flag is 0 while the bus is free following de-assertion of the internal reset signal.**
2. **When an internal reset is applied while the bus is not free, the BBSY flag is not cleared.**

21.3.4 RIICnMR1 — I2C Bus Mode Register 1

Access: RIICnMR1 is a 32-bit readable/writable register.
RIICnMR1L and RIICnMR1H are 16-bit readable/writable registers.
RIICnMR1LL, RIICnMR1LH, RIICnMR1HL, and RIICnMR1HH are 8-bit readable/writable registers.

Address: RIICnMR1: <RIICn_base> + 0008_H

RIICnMR1L: <RIICn_base> + 0008_H, RIICnMR1H: <RIICn_base> + $000A_H$

RIICnMR1LL: <RIICn_base> + 0008_H, RIICnMR1LH: <RIICn_base> + 0009_H,
RIICnMR1HL: <RIICn_base> + $000A_H$, RIICnMR1HH: <RIICn_base> + $000B_H$

Value after reset: 0000 0008_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	W	R/W	R/W	R/W

Table 21.12 RIICnMR1 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
7	MTWP	MST/TRS Write Protect 0: Disables writing to the RIICnCR2.MST and TRS bits. 1: Enables writing to the RIICnCR2.MST and TRS bits.
6 to 4	CKS[2:0]	Internal Reference Clock Selection (IICϕ) b6 b4 0 0 0: PCLK/1 clock 0 0 1: PCLK/2 clock 0 1 0: PCLK/4 clock 0 1 1: PCLK/8 clock 1 0 0: PCLK/16 clock 1 0 1: PCLK/32 clock 1 1 0: PCLK/64 clock 1 1 1: PCLK/128 clock
3	BCWP*1	BC[2:0] Write Protect 0: Enables a value to be written in the BC[2:0] bits. 1: Protects the BC[2:0] bits (This bit is read as 1.)
2 to 0	BC[2:0]	Bit Counter b2 b0 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits

Note 1. When rewriting the BC[2:0] bits, write 0 to the BCWP bit simultaneously.

BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred frames. When setting any value other than 000_B, set the value while the SCL line is at a low level.

[Clearing conditions]

- When 1 is written to the RIICnCR1.IICRST bit and a RIIC reset or internal reset is initiated.
- Data transfer including the acknowledge bit being completed.
- A start condition including a restart condition being detected.

21.3.5 RIICnMR2 — I2C Bus Mode Register 2

Access: RIICnMR2 is a 32-bit readable/writable register.
RIICnMR2L and RIICnMR2H are 16-bit readable/writable registers.
RIICnMR2LL, RIICnMR2LH, RIICnMR2HL, and RIICnMR2HH are 8-bit readable/writable registers.

Address: RIICnMR2: <RIICn_base> + $000C_H$

RIICnMR2L: <RIICn_base> + $000C_H$, RIICnMR2H: <RIICn_base> + $000E_H$

RIICnMR2LL: <RIICn_base> + $000C_H$, RIICnMR2LH: <RIICn_base> + $000D_H$,
RIICnMR2HL: <RIICn_base> + $000E_H$, RIICnMR2HH: <RIICn_base> + $000F_H$

Value after reset: $0000\ 0006_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 21.13 RIICnMR2 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
7	DLCS	SDA Output Delay Clock Source Selection 0: The internal reference clock (IICϕ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICϕ/2) is selected as the clock source of the SDA output delay counter.*1
6 to 4	SDDL[2:0]	SDA Output Delay Counter • When RIICnMR2.DLCS = 0 (IICϕ) b6 b4 0 0 0: No output delay 0 0 1: 1 IICϕ cycle 0 1 0: 2 IICϕ cycles 0 1 1: 3 IICϕ cycles 1 0 0: 4 IICϕ cycles 1 0 1: 5 IICϕ cycles 1 1 0: 6 IICϕ cycles 1 1 1: 7 IICϕ cycles • When RIICnMR2.DLCS = 1 (IICϕ/2) b6 b4 0 0 0: No output delay 0 0 1: 1 or 2 IICϕ cycles 0 1 0: 3 or 4 IICϕ cycles 0 1 1: 5 or 6 IICϕ cycles 1 0 0: 7 or 8 IICϕ cycles 1 0 1: 9 or 10 IICϕ cycles 1 1 0: 11 or 12 IICϕ cycles 1 1 1: 13 or 14 IICϕ cycles
3	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
2	TMOH	Timeout H Count Control 0: Disables counting while the SCL line is at a high level. 1: Enables counting while the SCL line is at a high level.

Table 21.13 RIICnMR2 Register Contents (2/2)

Bit Position	Bit Name	Function
1	TMOL	Timeout L Count Control 0: Disables counting while the SCL line is at a low level. 1: Enables counting while the SCL line is at a low level.
0	TMOS	Timeout Detection Time Selection 0: Long mode is selected. 1: Short mode is selected.

Note 1. The setting DLCS = 1 (IICϕ/2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting DLCS = 1 becomes invalid and the clock source becomes the internal reference clock (IICϕ).

TMOS Bit (Timeout Detection Time Selection)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (RIICnFER.TMOE bit = 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit-counter. While the SCL line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICϕ) as a count source.

For details on the timeout function, see Section 21.13.1, Timeout Function.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held low when the timeout function is enabled (RIICnFER.TMOE bit = 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held high when the timeout function is enabled (RIICnFER.TMOE bit = 1).

SDDL[2:0] Bits (SDA Output Delay Setup Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledge bit.

For details on this function, see Section 21.7, Facility for Delaying SDA Output.

CAUTION

Set the SDA output delay time to meet the I2C bus standard (within the data enable time/acknowledge enable time*1). Note that, if a value outside the standard is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

Note 1. Data enable time/acknowledge enable time
3,450 ns (up to 100 kbps: standard mode [Sm])
900 ns (up to 400 kbps: fast mode [Fm])

21.3.6 RIICnMR3 — I2C Bus Mode Register 3

Access: RIICnMR3 is a 32-bit readable/writable register.
RIICnMR3L and RIICnMR3H are 16-bit readable/writable registers.
RIICnMR3LL, RIICnMR3LH, RIICnMR3HL, and RIICnMR3HH are 8-bit readable/writable registers.

Address: RIICnMR3: <RIICn_base> + 0010_H

RIICnMR3L: <RIICn_base> + 0010_H, RIICnMR3H: <RIICn_base> + 0012_H

RIICnMR3LL: <RIICn_base> + 0010_H, RIICnMR3LH: <RIICn_base> + 0011_H,
RIICnMR3HL: <RIICn_base> + 0012_H, RIICnMR3HH: <RIICn_base> + 0013_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W

Table 21.14 RIICnMR3 Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
6	WAIT*2	WAIT 0: No WAIT (The period between ninth clock cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading RIICnDRR.
5	RDRFS*2	RDRF Flag Set Timing Selection 0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle. (The SCL line is not held low at the falling edge of the eighth clock cycle.) 1: The RDRF flag is set at the rising edge of the eighth SCL clock cycle. (The SCL line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKBT bit.
4	ACKWP*1	ACKBT Write Protect 0: Modification of the ACKBT bit is disabled. 1: Modification of the ACKBT bit is enabled.
3	ACKBT*1	Transmit Acknowledge 0: A 0 is sent as the acknowledge bit (ACK transmission). 1: A 1 is sent as the acknowledge bit (NACK transmission).
2	ACKBR	Receive Acknowledge 0: A 0 is received as the acknowledge bit (ACK reception). 1: A 1 is received as the acknowledge bit (NACK reception).
1, 0	NF[1:0]	Digital noise Filter Stage Selection b1 b0 0 0: Noise of up to one IICϕ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IICϕ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IICϕ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IICϕ cycles is filtered out (4-stage filter).

Note 1. If it is attempted to write 1 to both ACKWP and ACKBT bits, the ACKBT bit cannot be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

NF[1:0] Bits (Digital noise Filter Stage Selection)

These bits are used to select the number of stages of the digital noise filter.

CAUTION

Set the noise range to be filtered out by the noise filter within a range less than the SCL line high-level period or low-level period. If the noise range is set to a value of (SCL clock width: high-level period or low-level period, whichever is shorter) - [1.5 internal reference clock synchronized (IICϕ) cycles] or more, the SCL clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

ACKBR Bit (Receive Acknowledge)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

When 1 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1

[Clearing conditions]

- When 0 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledge timing in receive mode.

[Setting condition]

When 1 is written to this bit with the ACKWP bit set to 1

[Clearing conditions]

- When 0 is written to this bit after ACKWP reading while the ACKWP bit is set to 1
- When stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

CAUTION

The ACKBT bit must be written to while the ACKWP bit is 1. If the ACKBT bit is written to with the ACKWP bit cleared to 0, writing to the ACKBT bit is disabled.

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Selection)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL line low at the falling edge of the eighth SCL clock cycle.

When the RDRFS bit is 0, the SCL line is not held low at the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCL line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCL line is released by writing a value to the ACKBT bit.

After data is received with this setting, the SCL line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1) according to receive data.

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (RIICnDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCL line is held low from the falling edge of the ninth clock cycle until the RIICnDRR value is read each time single-byte data is received. This enables receive operation in byte units.

CAUTION

When the value of the WAIT bit is cleared to 0, be sure to read the RIICnDRR beforehand.

21.3.7 RIICnFER — I2C Bus Function Enable Register

Access: RIICnFER is a 32-bit readable/writable register.
RIICnFERL and RIICnFERH are 16-bit readable/writable registers.
RIICnFERLL, RIICnFERLH, RIICnFERHL, and RIICnFERHH are 8-bit readable/writable registers.

Address: RIICnFER: <RIICn_base> + 0014_H

RIICnFERL: <RIICn_base> + 0014_H, RIICnFERH: <RIICn_base> + 0016_H

RIICnFERLL: <RIICn_base> + 0014_H, RIICnFERLH: <RIICn_base> + 0015_H,
RIICnFERHL: <RIICn_base> + 0016_H, RIICnFERHH: <RIICn_base> + 0017_H

Value after reset: 0000 0072_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.15 RIICnFER Register Contents

Bit Position	Bit Name	Function
31 to 7	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
6	SCLE	SCL Synchronous Circuit Enable 0: No SCL synchronous circuit is used. 1: An SCL synchronous circuit is used.
5	NFE	Digital Noise Filter Circuit Enable 0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.
4	NACKE	NACK Reception Transfer Suspension Enable 0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).
3	SALE	Slave Arbitration-Lost Detection Enable 0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.
1	MALE	Master Arbitration-Lost Detection Enable 0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the RIICnCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the RIICnCR2.MST and TRS bits automatically when arbitration is lost.) If 1 is written to the ST bit while the BBSY flag is 1 in slave transmit mode, the TRS bit is not cleared.
0	TMOE	Timeout Function Enable 0: The timeout function is disabled. 1: The timeout function is enabled.

TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, see Section 21.13.1, Timeout Function.

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to select enabling or disabling of the arbitration-lost detection function. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether the detection of ACK during transmission of NACK in reception (such as when slaves with the same address are present on the bus and each is transmitting different data, or when two or more masters select the same slave device simultaneously with different numbers of bytes for reception) is judged to represent a loss in arbitration.

SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended.

When the NACKE bit is 0, the next transfer operation is continued regardless of the received acknowledge content.

SCLE Bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with a rising or falling edge on the SCL line. Normally, set this bit to 1.

When the SCLE bit is cleared to 0 (SCL synchronous circuit not used), the RIIC does not synchronize the SCL clock with the SCL input clock. In this setting, the RIIC outputs the SCL clock with the transfer rate set in RIICnBRH and RIICnBRL regardless of the SCL line state. For this reason, if the bus load of the I2C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.

This bit must not be cleared to 0 except for checking the output of the transfer rate.

21.3.8 RIICnSER — I2C Bus Status Enable Register

Access: RIICnSER is a 32-bit readable/writable register.
RIICnSERL and RIICnSERH are 16-bit readable/writable registers.
RIICnSERLL, RIICnSERLH, RIICnSERHL, and RIICnSERHH are 8-bit readable/writable registers.

Address: RIICnSER: <RIICn_base> + 0018_H

RIICnSERL: <RIICn_base> + 0018_H, RIICnSERH: <RIICn_base> + $001A_H$

RIICnSERLL: <RIICn_base> + 0018_H, RIICnSERLH: <RIICn_base> + 0019_H,
RIICnSERHL: <RIICn_base> + $001A_H$, RIICnSERHH: <RIICn_base> + $001B_H$

Value after reset: 0000 0009_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	DIDE	—	GCE	SAR2	SAR1	SAR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W

Table 21.16 RIICnSER Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
5	DIDE	Device-ID Address Detection Enable 0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.
4	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
3	GCE	General Call Address Enable 0: General call address detection is disabled. 1: General call address detection is enabled.
2	SAR2	Slave Address Register 2 Enable 0: Slave address in RIICnSAR2 is disabled. 1: Slave address in RIICnSAR2 is enabled.
1	SAR1	Slave Address Register 1 Enable 0: Slave address in RIICnSAR1 is disabled. 1: Slave address in RIICnSAR1 is enabled.
0	SAR0	Slave Address Register 0 Enable 0: Slave address in RIICnSAR0 is disabled. 1: Slave address in RIICnSAR0 is enabled.

SARy Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in RIICnSARy.

When this bit is set to 1, the slave address set in RIICnSARy is enabled and is compared with the received slave address.

When this bit is cleared to 0, the slave address set in RIICnSARy is disabled and is ignored even if it matches the received slave address.

GCAE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address ($0000\ 000_B$ + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in RIICnSARy (y = 0 to 2) and performs data receive operation.

When this bit is cleared to 0, the received slave address is ignored even if it matches the general call address.

DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the Device-ID address when a device ID ($1111\ 100_B$) is received in the first frame after a start condition or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the RIIC recognizes that the Device-ID address has been received. When the following R/W# bit is 0 [W], the RIIC recognizes the second and the following frames as slave addresses and continues the receive operation.

When this bit is cleared to 0, the RIIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device-ID address detection, see **Section 21.9.3, Device-ID Address Detection**.

21.3.9 RIICnIER — I2C Bus Interrupt Enable Register

Access: RIICnIER is a 32-bit readable/writable register.
RIICnIERL and RIICnIERH are 16-bit readable/writable registers.
RIICnIERLL, RIICnIERLH, RIICnIERHL, and RIICnIERHH are 8-bit readable/writable registers.

Address: RIICnIER: <RIICn_base> + $001C_H$

RIICnIERL: <RIICn_base> + $001C_H$, RIICnIERH: <RIICn_base> + $001E_H$

RIICnIERLL: <RIICn_base> + $001C_H$, RIICnIERLH: <RIICn_base> + $001D_H$,
RIICnIERHL: <RIICn_base> + $001E_H$, RIICnIERHH: <RIICn_base> + $001F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.17 RIICnIER Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
7	TIE	Transmit Data Empty Interrupt Enable 0: Transmit data empty interrupt request (INTIICnTI) is disabled. 1: Transmit data empty interrupt request (INTIICnTI) is enabled.
6	TEIE	Transmit End Interrupt Enable 0: Transmit end interrupt request (INTIICnTEI) is disabled. 1: Transmit end interrupt request (INTIICnTEI) is enabled.
5	RIE	Receive Complete Interrupt Enable 0: Receive complete interrupt request (INTIICnRI) is disabled. 1: Receive complete interrupt request (INTIICnRI) is enabled.
4	NAKIE	NACK Reception Interrupt Enable 0: NACK reception interrupt request (NAKI) is disabled. 1: NACK reception interrupt request (NAKI) is enabled.
3	SPIE	Stop Condition Detection Interrupt Enable 0: Stop condition detection interrupt request (SPI) is disabled. 1: Stop condition detection interrupt request (SPI) is enabled.
2	STIE	Start Condition Detection Interrupt Enable 0: Start condition detection interrupt request (STI) is disabled. 1: Start condition detection interrupt request (STI) is enabled.
1	ALIE	Arbitration-Lost Interrupt Enable 0: Arbitration-lost interrupt request (ALI) is disabled. 1: Arbitration-lost interrupt request (ALI) is enabled.
0	TMOIE	Timeout Interrupt Enable 0: Timeout interrupt request (TMOI) is disabled. 1: Timeout interrupt request (TMOI) is enabled.

TMOIE Bit (Timeout Interrupt Enable)

This bit is used to enable or disable timeout interrupt requests (TMOI) when the RIICnSR2.TMOF flag is set to 1. A TMOI interrupt request is canceled by clearing the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Enable)

This bit is used to enable or disable arbitration-lost interrupt requests (ALI) when the RIICnSR2.AL flag is set to 1. An ALI interrupt request is canceled by clearing the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Enable)

This bit is used to enable or disable start condition detection interrupt requests (STI) when the RIICnSR2.START flag is set to 1. An STI interrupt request is canceled by clearing the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Enable)

This bit is used to enable or disable stop condition detection interrupt requests (SPI) when the RIICnSR2.STOP flag is set to 1. An SPI interrupt request is canceled by clearing the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Enable)

This bit is used to enable or disable NACK reception interrupt requests (NAKI) when the RIICnSR2.NACKF flag is set to 1. An NAKI interrupt request is canceled by clearing the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Complete Interrupt Enable)

This bit is used to enable or disable receive complete interrupt requests (INTIICnRI) when the RIICnSR2.RDRF flag is set to 1.

TEIE Bit (Transmit End Interrupt Enable)

This bit is used to enable or disable transmit end interrupts (INTIICnTEI) when the RIICnSR2.TEND flag is set to 1. An INTIICnTEI interrupt request is canceled by clearing the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Enable)

This bit is used to enable or disable transmit data empty interrupts (INTIICnTI) when the RIICnSR2.TDRE flag is set to 1.

21.3.10 RIICnSR1 — I2C Bus Status Register 1

Access: RIICnSR1 is a 32-bit readable/writable register.
RIICnSR1L and RIICnSR1H are 16-bit readable/writable registers.
RIICnSR1LL, RIICnSR1LH, RIICnSR1HL, and RIICnSR1HH are 8/1-bit readable/writable registers.

Address: RIICnSR1: <RIICn_base> + 0020_H

RIICnSR1L: <RIICn_base> + 0020_H, RIICnSR1H: <RIICn_base> + 0022_H

RIICnSR1LL: <RIICn_base> + 0020_H, RIICnSR1LH: <RIICn_base> + 0021_H,
RIICnSR1HL: <RIICn_base> + 0022_H, RIICnSR1HH: <RIICn_base> + 0023_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R(/W)*1	R	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1

Note 1. Only 0 can be written to this bit.

Table 21.18 RIICnSR1 Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
5	DID	Device-ID Address Detection Flag 0: Device-ID address is not detected. 1: Device-ID address is detected.
4	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
3	GCA	General Call Address Detection Flag 0: General call address is not detected. 1: General call address is detected.
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 is not detected. 1: Slave address 2 is detected.
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 is not detected. 1: Slave address 1 is detected.
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 is not detected. 1: Slave address 0 is detected.

AASy Flag (Slave Address y Detection) (y = 0 to 2)

[Setting conditions]

<For 7-bit address format: RIICnSARy.FSy = 0>

When the received slave address matches the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

When the received slave address matches a value of (1111 0_B + RIICnSARy.SVA[9:8]) and the following address matches the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy bit after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

<For 7-bit address format: RIICnSARy.FSy = 0>

- When the received slave address does not match the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

- When the received slave address does not match a value of (1111 0_B + RIICnSARy.SVA[9:8]) with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (1111 0_B + RIICnSARy.SVA[9:8]) and the following address does not match the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

GCA Flag (General Call Address Detection)

[Setting condition]

- When the received slave address matches the general call address ($0000\ 000_B$ + 0 [W]) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address ($0000\ 000_B$ + 0 [W]) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection)

[Setting condition]

- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID ($1111\ 100_B$) + 0 [W]) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled). This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.
- When a restart condition is detected after a match with the device ID address and the device ID address ($1111\ 100_B$) plus 1[R] has matched while the setting of the RIICnSER.DIDE bit is 1 (device ID address detection enabled). This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start condition or restart condition is detected does not match a value of (device ID ($1111\ 100_B$)) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID ($1111\ 100_B$) + 0 [W]) and the second frame does not match any of slave addresses 0 to 2 with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

21.3.11 RIICnSR2 — I2C Bus Status Register 2

Access: RIICnSR2 is a 32-bit readable/writable register.
RIICnSR2L and RIICnSR2H are 16-bit readable/writable registers.
RIICnSR2LL, RIICnSR2LH, RIICnSR2HL, and RIICnSR2HH are 8/1-bit readable/writable registers.

Address: RIICnSR2: <RIICn_base> + 0024_H

RIICnSR2L: <RIICn_base> + 0024_H, RIICnSR2H: <RIICn_base> + 0026_H

RIICnSR2LL: <RIICn_base> + 0024_H, RIICnSR2LH: <RIICn_base> + 0025_H,
RIICnSR2HL: <RIICn_base> + 0026_H, RIICnSR2HH: <RIICn_base> + 0027_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1	R(/W)*1

Note 1. Only 0 can be written to this bit.

Table 21.19 RIICnSR2 Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
7	TDRE	Transmit Data Empty Flag 0: RIICnDRT contains transmit data. 1: RIICnDRT contains no transmit data.
6	TEND	Transmit End Flag 0: Data is being transmitted. 1: Data has been transmitted.
5	RDRF	Receive Complete Flag 0: RIICnDRR contains no receive data. 1: RIICnDRR contains receive data.
4	NACKF	NACK Detection Flag 0: NACK is not detected. 1: NACK is detected.
3	STOP	Stop Condition Detection Flag 0: Stop condition is not detected. 1: Stop condition is detected.
2	START	Start Condition Detection Flag 0: Start condition is not detected. 1: Start condition is detected.
1	AL	Arbitration-Lost Flag 0: Arbitration is not lost. 1: Arbitration is lost.
0	TMOF	Timeout Detection Flag 0: Timeout is not detected. 1: Timeout is detected.

TMOF Flag (Timeout Detection)

This flag is set to 1 when the RIIC recognizes timeout after the SCL line state remains unchanged for a certain period.

[Setting condition]

The timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state that the SCL line is stuck low or high during the following conditions:

- The bus is busy (the RIICnCR2.BBSY flag is 1) in master mode (the RIICnCR2.MST bit is 1).
- The RIIC's own slave address matches (the RIICnSR1 register is not 00_H) and the bus is busy (the RIICnCR2.BBSY bit is 1) in slave mode (the RIICnCR2.MST bit is 0).
- The bus is free (the RIICnCR2.BBSY flag is 0) while generation of a start condition is requested (the RIICnCR2.ST bit is 1).

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

AL Flag (Arbitration-Lost)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is issued or an address and data are transmitted. The RIIC monitors the level on the SDA line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL bit to 1 to indicate that the bus is occupied by another device.

The RIIC can also set the flag to indicate the detection of loss of arbitration during NACK transmission in receive mode or during data transmission in slave mode.

[Setting conditions]

<When master arbitration-lost detection is enabled: RIICnFER.MALE = 1>

- When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA line is driven low while the internal SDA output is at a high level (the SDA pin is in the high impedance state))
- When a start condition is detected while the RIICnCR2.ST bit is 1 (start condition issuance request) or the internal SDA output state does not match the SDA line level
- When the RIICnCR2.ST bit is set to 1 (start condition issuance request) with the RIICnCR2.BBSY flag set to 1.

<When NACK arbitration-lost detection is enabled: RIICnFER.NALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode

<When slave arbitration-lost detection is enabled: RIICnFER.SALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL bit after reading AL = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

Table 21.20 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions

RIICnFER			RIICnSR2		
MALE	NALE	SALE	AL	Error	Arbitration-Lost Generation Source
1	—	—	1	Start condition issuance error	When internal SDA output state does not match SDA line level when a start condition is detected while the RIICnCR2.ST bit is 1
					When RIICnCR2.ST is set to 1 with RIICnCR2.BBSY set to 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
—	1	—	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master receive mode or slave receive mode
—	—	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

—: Don't care

START Flag (Start Condition Detection)

[Setting condition]

When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection)

[Setting condition]

When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

NACKF Flag (NACK Detection)

[Setting condition]

When acknowledge is not received (NACK is received) from the receive device in transmit mode with the RIICnFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to RIICnDRT in transmit mode or reading from RIICnDRR in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, clear the NACKF flag to 0.

RDRF Flag (Receive Complete)

[Setting conditions]

- When receive data has been transferred from RIICnDRS to RIICnDRR
  This flag is set to 1 at the rising edge of the eighth or ninth SCL clock cycle (selected by the RIICnMR3.RDRFS bit)
- When the received slave address matches the address enabled in RIICnSER after a start condition (or a restart condition) is detected with the RIICnCR2.TRS bit cleared to 0
- In master mode, transition to master reception while the R/W# bit appended to the slave address is set to 1

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from RIICnDRR
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

TEND Flag (Transmit End)

[Setting condition]

At the rising edge of the ninth SCL clock cycle while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to RIICnDRT
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty)

[Setting conditions]

- When data has been transferred from RIICnDRT to RIICnDRS and RIICnDRT becomes empty
- When the RIICnCR2.TRS bit is set to 1
  - When the RIICnCR2.MST bit is set to 1 after a start condition is detected
  - When the RIIC enters transmit mode from receive mode
  - When 1 is written to the RIICnCR2.TRS bit while the RIICnMR1.MTWP bit is 1
  - When the received slave address matches the address enabled in RIICnSER after a start condition including a restart condition is detected with the RIICnCR2.TRS bit set to 1
- When the received slave address matches while the TRS bit is 1

[Clearing conditions]

- When data is written to RIICnDRT
- When the RIICnCR2.TRS bit is cleared to 0
  - When a stop condition is detected
  - When the RIIC enters receive mode from transmit mode
  - When 0 is written to the RIICnCR2.TRS bit while the RIICnMR1.MTWP bit is 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

CAUTION

When the NACKF flag is set to 1 while the RIICnFER.NACKE bit is 1, the RIIC suspends data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the RIICnDRS register and the RIICnDRT register becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

21.3.12 RIICnSARy — I2C Slave Address Register y (y = 0 to 2)

Access: RIICnSARy is a 32-bit readable/writable register.
RIICnSARyL and RIICnSARyH are 16-bit readable/writable registers.
RIICnSARyLL, RIICnSARyLH, RIICnSARyHL, and RIICnSARyHH are 8-bit readable/writable registers.

Address: RIICnSAR0: <RIICn_base> + 0028_H

RIICnSAR0L: <RIICn_base> + 0028_H, RIICnSAR0H: <RIICn_base> + $002A_H$

RIICnSAR0LL: <RIICn_base> + 0028_H, RIICnSAR0LH: <RIICn_base> + 0029_H,
RIICnSAR0HL: <RIICn_base> + $002A_H$, RIICnSAR0HH: <RIICn_base> + $002B_H$

RIICnSAR1: <RIICn_base> + $002C_H$

RIICnSAR1L: <RIICn_base> + $002C_H$, RIICnSAR1H: <RIICn_base> + $002E_H$

RIICnSAR1LL: <RIICn_base> + $002C_H$, RIICnSAR1LH: <RIICn_base> + $002D_H$,
RIICnSAR1HL: <RIICn_base> + $002E_H$, RIICnSAR1HH: <RIICn_base> + $002F_H$

RIICnSAR2: <RIICn_base> + 0030_H

RIICnSAR2L: <RIICn_base> + 0030_H, RIICnSAR2H: <RIICn_base> + 0032_H

RIICnSAR2LL: <RIICn_base> + 0030_H, RIICnSAR2LH: <RIICn_base> + 0031_H,
RIICnSAR2HL: <RIICn_base> + 0032_H, RIICnSAR2HH: <RIICn_base> + 0033_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FSy	—	—	—	—	—	SVA[9:1]									SVA0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21.21 RIICnSARy Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
15	FSy	7-Bit/10-Bit Address Format Selection 0: The 7-bit address format is selected. 1: The 10-bit address format is selected.
14 to 10	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
9 to 1	SVA[9:1]	7-Bit Address/10-Bit Address Upper Bits A slave address is set. • When the FSy bit is 0 (7-bit address format), the SVA[7:1] bits are valid and form a 7-bit slave address. • When the FSy bit is 1 (10-bit address format), SVA[9:1] bits form a 10-bit slave address (combined with the SVA0 bit).
0	SVA0	10-Bit Address LSB The least significant bit (LSB) of a 10-bit slave address is set. • When the FSy bit is 0 (7-bit address format), this bit is invalid. • When the FSy bit is 1 (10-bit address format), this bit is a 10-bit slave address (combined with the SVA[9:1] bits).

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (RIICnSARy.FSy = 1), this bit functions as the LSB of a 10-bit address and forms a 10-bit address in combination with the SVA[9:1] bits.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, this bit is valid. While the RIICnSARy.FSy bit or SARy bit is 0, the setting of this bit is ignored.

SVA[9:1] Bits (7-Bit Address/10-Bit Address Upper Bits)

When the 7-bit address format is selected (RIICnSARy.FSy = 0), these bits function as a 7-bit address. When the 10-bit address format is selected (RIICnSARy.FSy = 1), these bits function as a 10-bit address in combination with the SVA0 bit.

While the RIICnSER.SARy bit is 0, the setting of these bits is ignored.

FSy Bit (7-Bit/10-Bit Address Format Selection)

This bit is used to select 7-bit address or 10-bit address for slave address y (in RIICnSARy).

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 0, the 7-bit address format is selected for slave address y, the RIICnSARy.SVA[7:1] setting is valid, and the settings of the SVA[9:8] bits and the RIICnSARy.SVA0 bit are ignored.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[9:1] bits and the SVA0 bit are valid.

While the RIICnSER.SARy bit is 0 (RIICnSARy disabled), the setting of the RIICnSARy.FSy bit is invalid.

21.3.13 RIICnBRL — I2C Bus Bit Rate Low-Level Register

Access: RIICnBRL is a 32-bit readable/writable register.
RIICnBRLL and RIICnBRLH are 16-bit readable/writable registers.
RIICnBRLLL, RIICnBRLLH, RIICnBRLHL, and RIICnBRLHH are 8-bit readable/writable registers.

Address: RIICnBRL: <RIICn_base> + 0034_H

RIICnBRLL: <RIICn_base> + 0034_H, RIICnBRLH: <RIICn_base> + 0036_H

RIICnBRLLL: <RIICn_base> + 0034_H, RIICnBRLLH: <RIICn_base> + 0035_H,
RIICnBRLHL: <RIICn_base> + 0036_H, RIICnBRLHH: <RIICn_base> + 0037_H

Value after reset: 0000 $00FF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	BRL[4:0]				
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 21.22 RIICnBRL Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
4 to 0	BRL[4:0]	Bit Rate Low-Level Period Low-level period of SCL clock

The RIICnBRL[4:0] register is a 5-bit register that is used to set the width at low level for the SCL clock.

It also works to generate the data setup time for automatic SCL low-hold operation (see Section 21.10, Automatic Low-Hold Function for SCL); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time*1.

RIICnBRL[4:0] counts the low-level period with the internal reference clock source (IICϕ) specified by the RIICnMR1.CKS[2:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns (up to 100 kbps: standard mode [Sm])
100 ns (up to 400 kbps: fast mode [Fm])

21.3.14 RIICnBRH — I2C Bus Bit Rate High-Level Register

Access: RIICnBRH is a 32-bit readable/writable register.
RIICnBRHL and RIICnBRHH are 16-bit readable/writable registers.
RIICnBRHLL, RIICnBRHLH, RIICnBRHHL, and RIICnBRHHH are 8-bit readable/writable registers.

Address: RIICnBRH: <RIICn_base> + 0038_H

RIICnBRHL: <RIICn_base> + 0038_H, RIICnBRHH: <RIICn_base> + $003A_H$

RIICnBRHLL: <RIICn_base> + 0038_H, RIICnBRHLH: <RIICn_base> + 0039_H,
RIICnBRHHL: <RIICn_base> + $003A_H$, RIICnBRHHH:<RIICn_base> + $003B_H$

Value after reset: 0000 $00FF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	BRH[4:0]				
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 21.23 RIICnBRH Register Contents

Bit Position	Bit Name	Function
31 to 5	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
4 to 0	BRH[4:0]	Bit Rate High-Level Period High-level period of SCL clock

RIICnBRH[4:0] is a 5-bit register to set the high-level period of SCL clock. RIICnICBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high-level period.

RIICnBRH[4:0] counts the high-level period with the internal reference clock source (IICϕ) specified by the RIICnMR1.CKS[2:0] bits.

The I2C transfer rate and the SCL clock duty are calculated using the following expression.

(1) When RIICnFER.SCLE = 0
Transfer rate = 1 / {[(RIICnBRH+1) + (RIICnBRL+1)] / IICφ*1 + tr + tf}
Duty cycle = {tr+[(RIICnBRH+1)/ IICφ]}/{ tr + tf + [(RIICnBRH+1) + (RIICnBRL+1)]/IICφ}

(2) When RIICnFER.SCLE=1, RIICnFER.NFE=0, IICφ = PCLK
Transfer rate = 1 / {[(RIICnBRH+3) + (RIICnBRL+3)] / IICφ*1 + tr + tf}
Duty cycle = {tr +[(RIICnBRH+3)/ IICφ]}/{ tr + tf + [(RIICnBRH+3) + (RIICnBRL+3)]/IICφ}

(3) When RIICnFER.SCLE=1, RIICnFER.NFE=1, IICφ = PCLK
Transfer rate =1 / {[(RIICnBRH+3+nf) + (RIICnBRL+3+nf)] / IICφ*1 + tr + tf}
Duty cycle = {tr +[(RIICnBRH+3+nf)/ IICφ]}/{ tr + tf + [(RIICnBRH+3+nf) + (RIICnBRL+3+nf)]/ IICφ}

(4) RIICnFER.SCLE=1, RIICnFER.NFE=0, IICφ < PCLK
Transfer rate =1 / {[(RIICnBRH+2) + (RIICnBRL+2)] / IICφ*1+ tr + tf}
Duty cycle = {tr +[(RIICnBRH+2)/ IICφ]}/{ tr + tf + [(RIICnBRH+2) + (RIICnBRL+2)]/IICφ}

(5) When RIICnFER.SCLE=1, RIICnFER.NFE=1, IICφ < PCLK
Transfer rate =1 / {[(RIICnBRH+2+nf) + (RIICnBRL+2+nf)] / IICφ*1+ tr + tf}
Duty cycle = {tr +[(RIICnBRH+2+nf)/ IICφ]}/{ tr + tf + [(RIICnBRH+2+nf) + (RIICnBRL+2+nf)]/ IICφ}

tf: SCL line falling time [ns]*2

tr: SCL line rising time [ns]*2

nf: Digital noise filter stage

Duty cycle: 0% < Duty < 100%

NOTES

1. As for IICφ, see CKS[2:0] in Section 21.3.4, RIICnMR1 — I2C Bus Mode Register 1.
2. The SCL line rising time [tr] and SCL line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I2C bus standard from NXP Semiconductors.

Table 21.24 lists examples of RIICnBRH/RIICnBRL settings.

Table 21.24 Examples of RIICnBRH/RIICnBRL Settings for Transfer Rate (1)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)							
	40				80			
	CKS[2:0]	CKS	RIICnBRH	RIICnBRL	CKS[2:0]	CKS	RIICnBRH	RIICnBRL
10	111b	128	14 (EEh)	15 (EFh)	111b	128	29 (FDh)	31 (FFh)
50	100b	16	21 (F5h)	24 (F8h)	101b	32	21 (F5h)	24 (F8h)
100	011b	8	20 (F4h)	22 (F6h)	100b	16	19 (F3h)	23 (F7h)
400	001b	2	17 (F1h)	19 (F3h)	011b	8	7 (E7h)	10 (EAh)

CAUTION

CBRH/ICBRL settings in these tables are calculated using the following values:
SCL line rising time (tr): 100 kbps or less, [Sm]: 1000 ns, 400 kbps or less, [Fm]: 300 ns
SCL line falling time (tf): 400 kbps or less, [Sm/Fm]: 300 ns
For the specified values of SCL line rising time (tr) and SCL line falling time (tf), see the I2C bus standard from NXP Semiconductors.

21.3.15 RIICnDRT — I2C Bus Transmit Data Register

Access: RIICnDRT is a 32-bit readable/writable register.
RIICnDRTL and RIICnDRTH are 16-bit readable/writable registers.
RIICnDRTLL, RIICnDRTLH, RIICnDRTHL, and RIICnDRTHH are 8-bit readable/writable registers.

Address: RIICnDRT: <RIICn_base> + $003C_H$

RIICnDRTL: <RIICn_base> + $003C_H$, RIICnDRTH: <RIICn_base> + $003E_H$

RIICnDRTLL: <RIICn_base> + $003C_H$, RIICnDRTLH: <RIICn_base> + $003D_H$,
RIICnDRTHL: <RIICn_base> + $003E_H$, RIICnDRTHH: <RIICn_base> + $003F_H$

Value after reset: $0000\ 00FF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

When RIICnDRT detects a space in the I2C bus shift register (RIICnDRS), it transfers the transmit data that has been written to RIICnDRT to RIICnDRS and starts transmitting data in transmit mode.

The double-buffer structure of RIICnDRT and RIICnDRS allows continuous transmit operation if the next transmit data has been written to RIICnDRT while the RIICnDRS data is being transmitted.

RIICnDRT can always be read and written. Write transmit data to RIICnDRT once when a transmit data empty interrupt (INTIICnTI) request is generated. When writing to bit 31 to 8, write 0.

21.3.16 RIICnDRR — I2C Bus Receive Data Register

Access: RIICnDRR is a 32-bit readable register.
RIICnDRRL and RIICnDRRH are 16-bit readable registers.
RIICnDRRLL, RIICnDRRLH, RIICnDRRHL, and RIICnDRRHH are 8-bit readable registers.

Address: RIICnDRR: <RIICn_base> + 0040_H

RIICnDRRL: <RIICn_base> + 0040_H, RIICnDRRH: <RIICn_base> + 0042_H

RIICnDRRLL: <RIICn_base> + 0040_H, RIICnDRRLH: <RIICn_base> + 0041_H,
RIICnDRRHL: <RIICn_base> + 0042_H, RIICnDRRHH: <RIICn_base> + 0043_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRR[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

When 1 byte of data has been received, the received data is transferred from the I2C bus shift register (RIICnDRS) to RIICnDRR to enable the next data to be received.

The double-buffer structure of RIICnDRS and RIICnDRR allows continuous receive operation if the received data has been read from RIICnDRR while RIICnDRS is receiving data.

RIICnDRR cannot be written. Read data from RIICnDRR once when a receive complete interrupt (INTIICnRI) request is generated.

If RIIC receives the next receive data before the current data is read from RIICnDRR (while the RIICnSR2.RDRF flag is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

21.3.17 RIICnDRS — I2C Bus Shift Register

Access: This register is not accessible.

Address: —

Value after reset: 0000 00FF$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DRS[7:0]							
Value after reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

RIICnDRS is a shift register to transmit and receive data.

During transmission, transmit data is transferred from RIICnDRT to RIICnDRS and is sent from the SDA pin. During reception, data is transferred from RIICnDRS to RIICnDRR after 1 byte of data has been received.

RIICnDRS cannot be accessed directly.

21.4 Interrupt Sources

The RIIC issues four types of interrupt request: transfer error or event generation (arbitration-lost detection, NACK detection, timeout detection, start condition detection, and stop condition detection), receive complete, transmit data empty, and transmit complete.

Table 21.25 lists details of the several interrupt requests. The receive complete and transmit data empty interrupt request are both capable of launching data transfer by the DMAC.

Table 21.25 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	DMACA Launching	Interrupt Condition
INTIICnTI	Transmit Data Empty	TDRE	Possible	TDRE = 1 and TIE = 1
INTIICnTEI	Transmit End	TEND	Not possible	TEND = 1 and TEIE = 1
INTIICnRI	Receive Complete	RDRF	Possible	RDRF = 1 and RIE = 1
INTIICnEE	Transfer Error/ Event Generation	AL	Not possible	AL = 1 and ALIE = 1
		NACKF		NACKF = 1 and NAKIE = 1
		TMOF		TMOF = 1 and TMOIE = 1
		START		START = 1 and STIE = 1
		STOP		STOP = 1 and SPIE = 1

Clear or mask the each flag during interrupt handling.

CAUTIONS

1. **There is a latency (delay) between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt processing. Returning from interrupt processing without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.**
2. **Since INTIICnTI is an edge-detected interrupt, it does not require clearing. Furthermore, the RIICnSR2.TDRE flag (a condition for INTIICnTI) is automatically cleared to 0 when data for transmission are written to RIICnDRT or a stop condition is detected (RIICnSR2.STOP flag = 1).**
3. **Since INTIICnRI is an edge-detected interrupt, it does not require clearing. Furthermore, the RIICnSR2.RDRF flag (a condition for INTIICnRI) is automatically cleared to 0 when data are read from RIICnDRR.**
4. **When using the INTIICnTEI interrupt, clear the RIICnSR2.TEND flag in the INTIICnTEI interrupt processing.**
   Note that the RIICnSR2.TEND flag is automatically cleared to 0 when data for transmission are written to RIICnDRT or a stop condition is detected (RIICnSR2.STOP flag = 1).

21.5 Operation

21.5.1 Communication Data Format

The I2C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start condition or restart condition is an address frame used to specify a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.

Figure 21.3 shows the I2C bus format, and **Figure 21.4** shows the I2C bus timing.

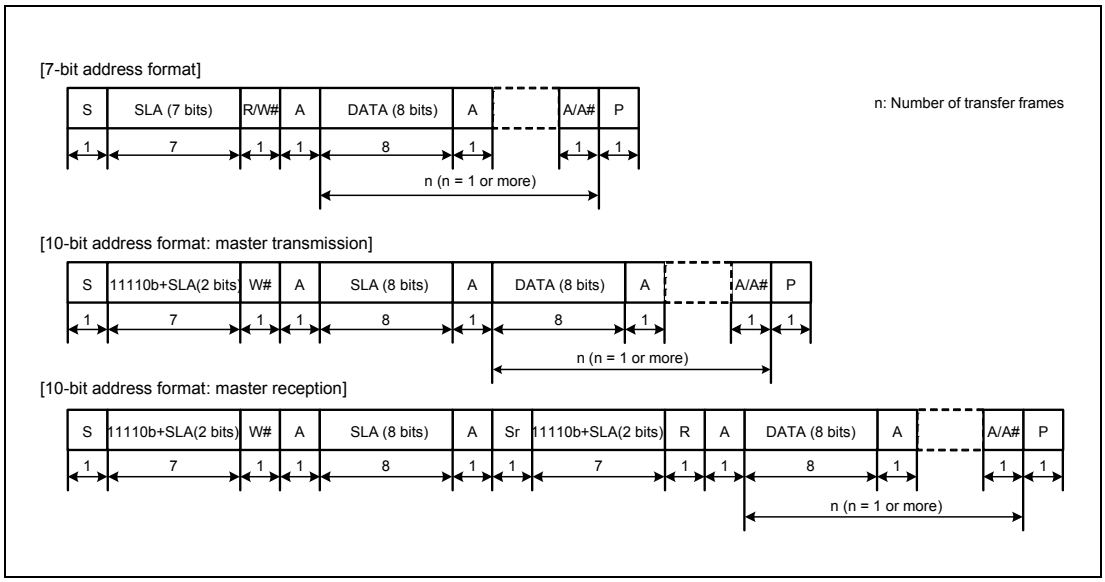


Figure 21.3 I2C Bus Format

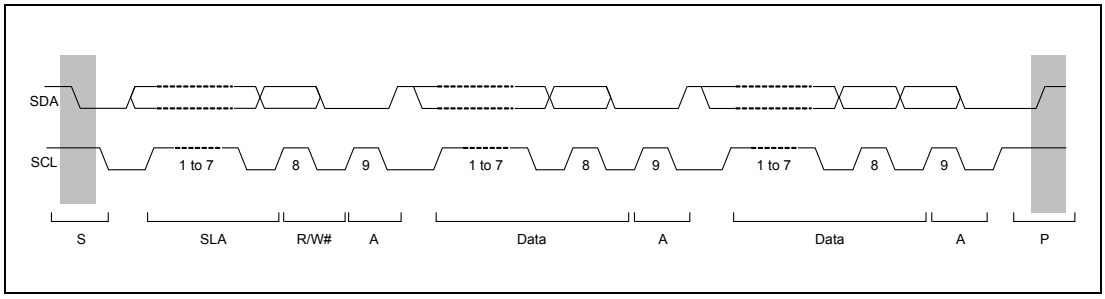


Figure 21.4 I2C Bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDA line low from high level while the SCL line is at a high level.

SLA: Slave address, by which the master device selects a slave device.

R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.

A: Acknowledge. The receive device drives the SDA line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not-acknowledge. The receiving device has not returned a response or is not present so the SDA line has remained at the high level.

Sr: Restart condition. The master device drives the SDA line low from the high level after the setup time has elapsed with the SCL line at the high level.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA line high from low level while the SCL line is at a high level.

21.5.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in **Figure 21.5**. Make initial settings for the RIIC once when starting the RIIC.

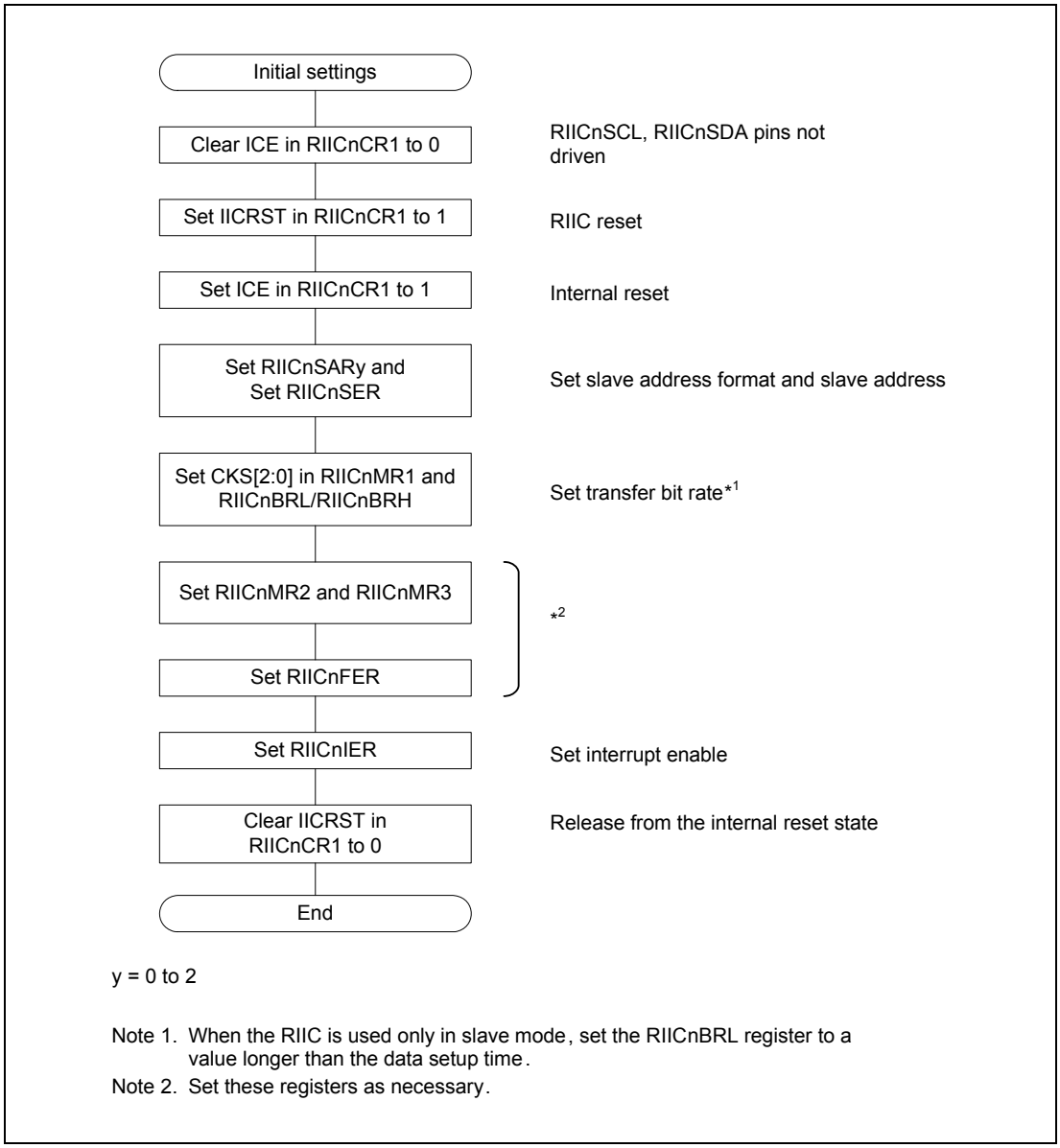


Figure 21.5 Example of RIIC Initialization Flowchart

21.5.3 Master Transmit Operation

In master transmit operation, the RIIC outputs the SCL (clock) and transmitted data signals as the master device, and the slave device returns acknowledgements. **Figure 21.6** shows an example of usage of master transmission and **Figure 21.7** to **Figure 21.9** show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

(1) Set the RIICnCR1.IICRST bit 1 to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see **Figure 21.5**). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS and MST bits to 1.

(3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. After the slave address including the R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.
Since the RIICnSR2.NACKF flag being 1 at this time indicates that the slave address has not been recognized or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0_B, the two higher-order bits of the slave address, and W# to RIICnDRT as the first address transmission. Then, as the second address transmission, write the eight lower-order bits of the slave address to RIICnDRT.

(4) After confirming that the RIICnSR2.TDRE flag is 1, write the data for transmission to the RIICnDRT register. The RIIC automatically holds the SCL line low until the data for transmission are ready or a stop condition is issued.

(5) After all bytes of data for transmission have been written to the RIICnDRT register, wait until the value of the RIICnSR2.TEND flag returns to 1, and then set the RIICnCR2.SP bit to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.

(6) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00_B and enters slave receive mode. Furthermore, it automatically clears the RIICnSR2.TDRE and TEND flags to 0, and sets the RIICnSR2.STOP flag in to 1.

(7) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

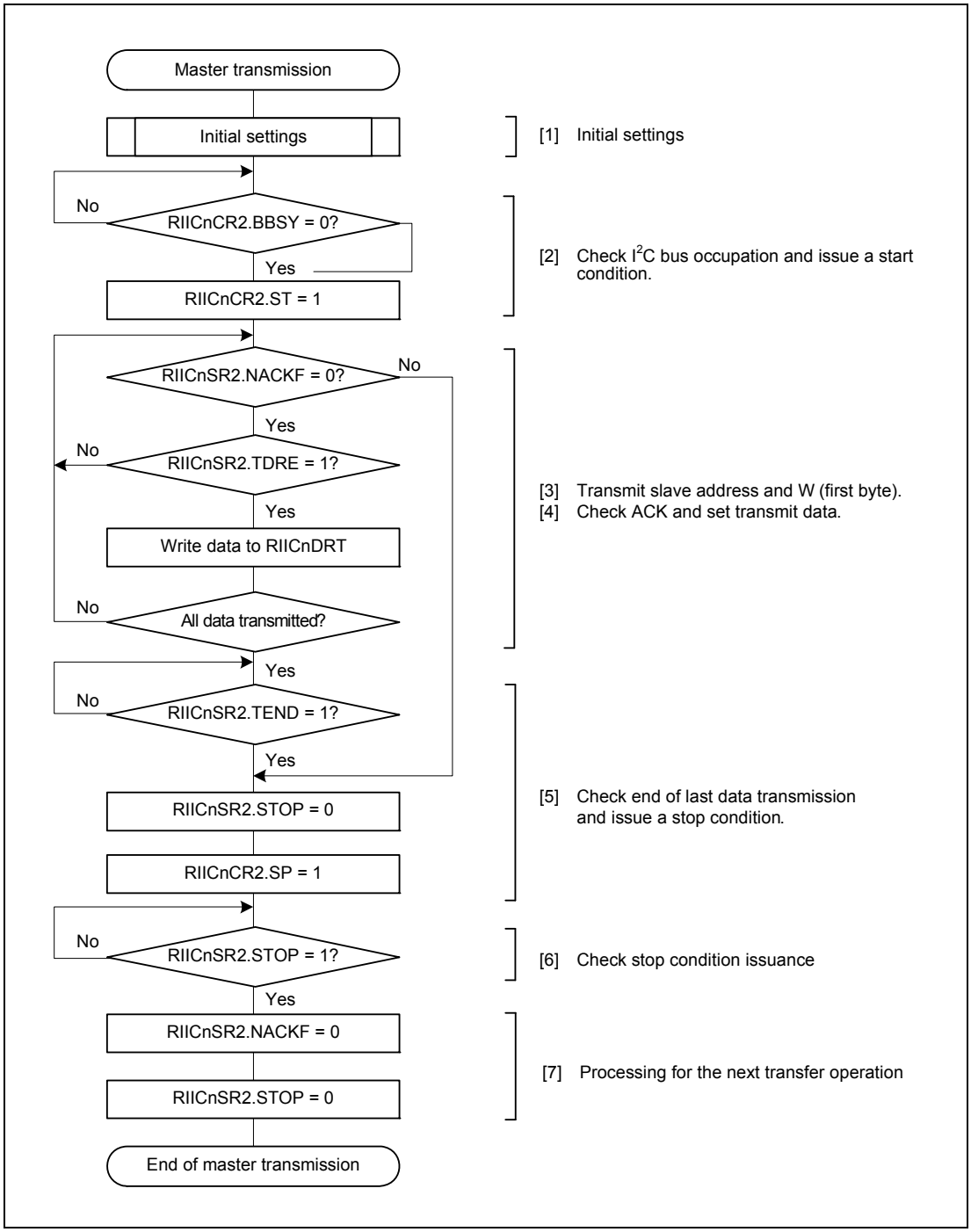


Figure 21.6 Example of Master Transmission Flowchart

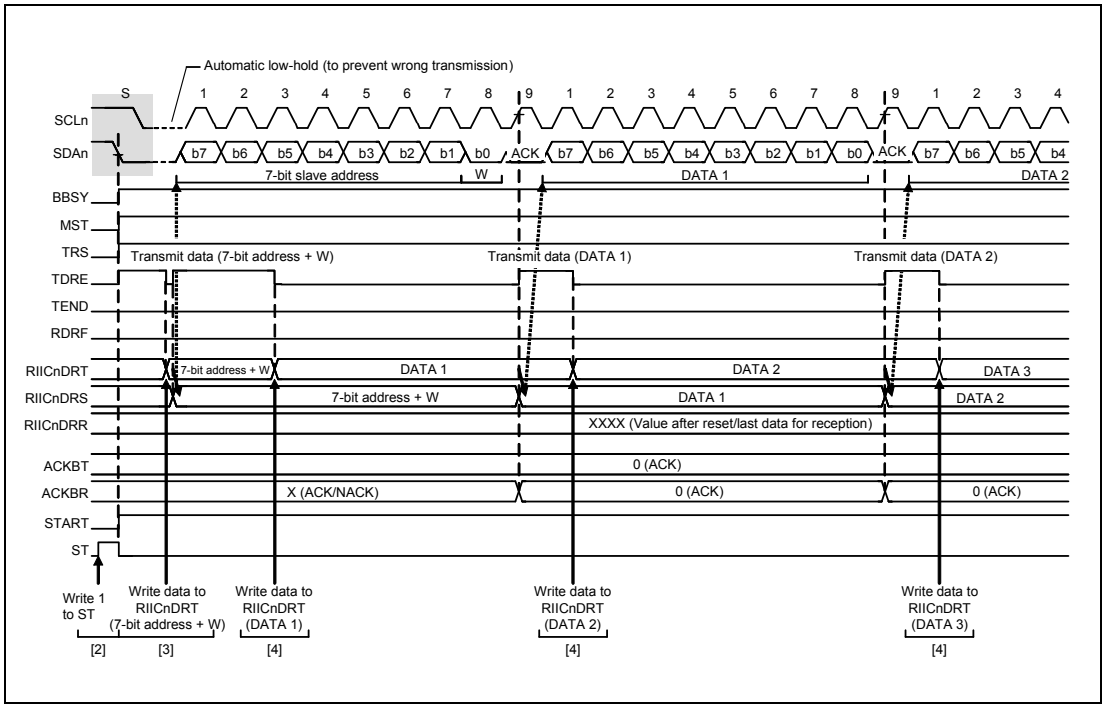


Figure 21.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

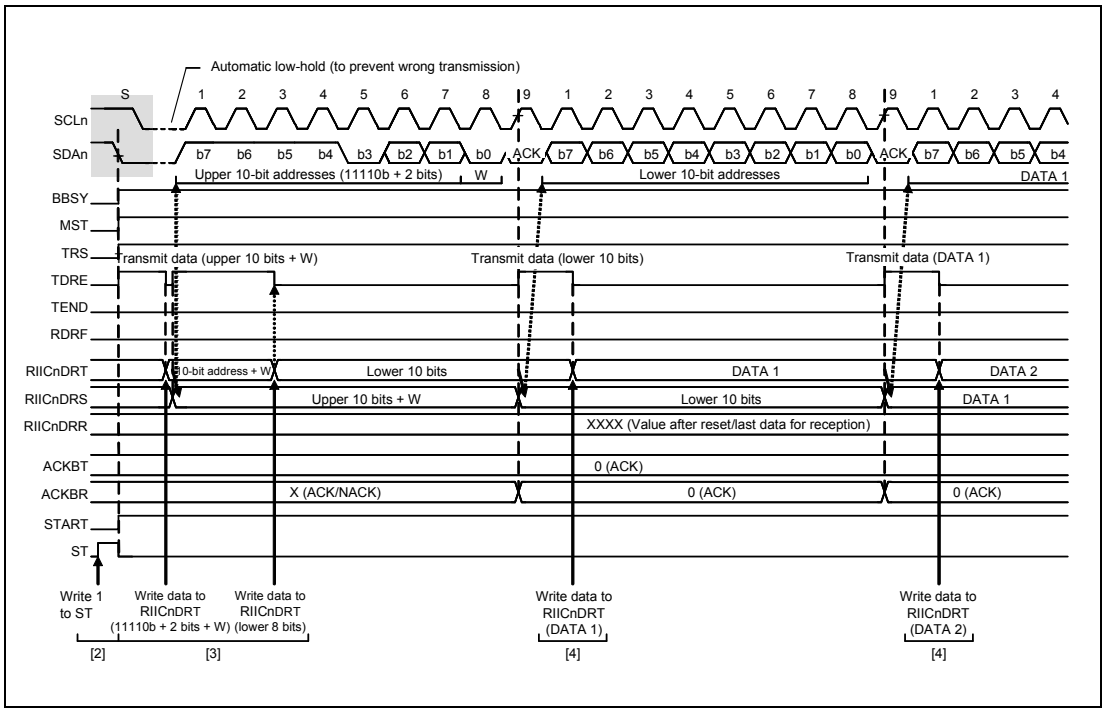


Figure 21.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

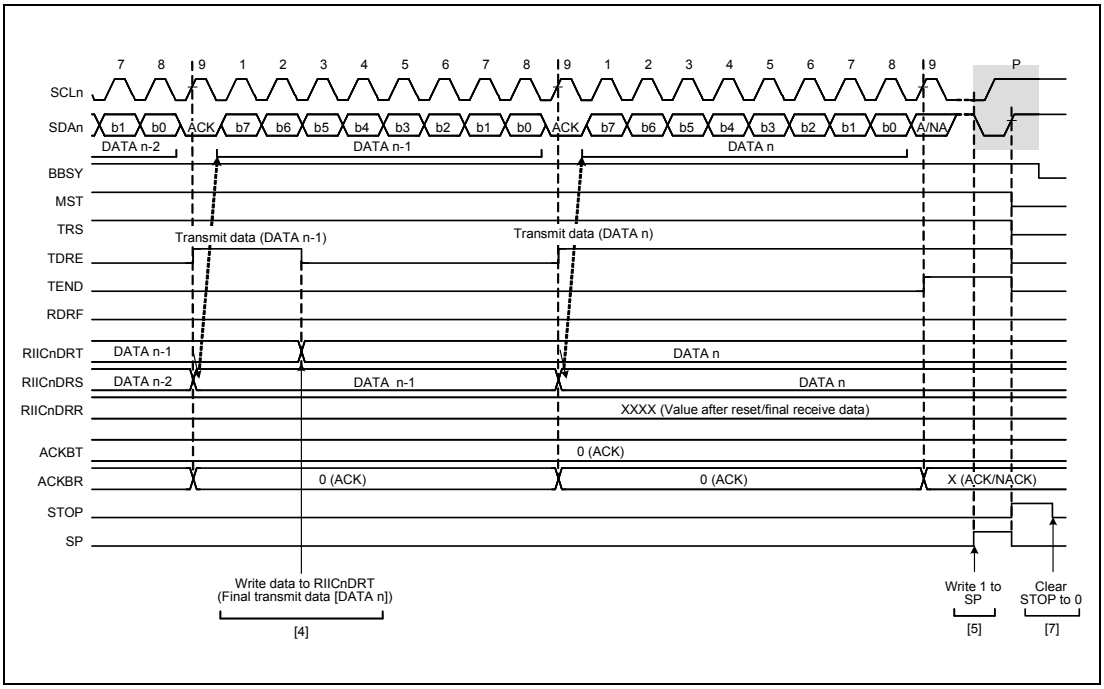


Figure 21.9 Master Transmit Operation Timing (3)

21.5.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL (clock) signal, receives data from the slave device, and returns acknowledgements. Since the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 21.10 shows an example of master reception flowchart (7-bit address format, 1 or 2 bytes), **Figure 21.11** shows an example of master reception flowchart (7-bit address format, 3 bytes or more), and **Figure 21.12** to **Figure 21.14** show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

(1) Set the RIICnCR1.IICRST bit to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see **Figure 21.5**). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. When the RIIC detects the start condition, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the slave address and value of the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. After the slave address including the R/W# bit has been transmitted, the value of the RIICnCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the RIICnCR2.TRS bit is cleared to 0 on the rising edge of the ninth cycle of SCL (the clock signal), placing the RIIC in master receive mode. At this time, the TDRE flag is automatically cleared to 0 and the RIICnSR2.RDRF flag is automatically set to 1.
Since the RIICnSR2.NACKF flag being 1 at this time indicates that the slave address has not been recognized or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to transmit the two higher-order bits of the slave address and then the eight lower-order bits of the slave address, and issue a restart condition following generation of the transmission end interrupt (or after TEND = 1) (see **Figure 21.13** for operation timing).
After that, transmitting 1111 0B plus the two higher-order bits of the slave address and the R bit places the RIIC in master receive mode.

(4) Dummy read RIICnDRR after confirming that the RIICnSR2.RDRF flag is 1; this makes the RIIC start output of the SCL (clock) signal and start data reception.

(5) After 1 byte of data has been received, the RIICnSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RIICnMR3.RDRFS bit. Reading out RIICnDRR at this time will produce the received data, and the RDRF flag is automatically cleared to 0 at the same time. Furthermore, the value of the acknowledgement field received during the ninth cycle of SCL clock is returned as the value set in the RIICnMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the RIICnMR3.WAIT bit to 1 (for wait insertion) before reading the RIICnDRR (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the RIICnMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a stop condition is possible.

(6) When the RIICnMR3.RDRFS bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the RIICnMR3.ACKBT bit to 1 (NACK).

(7) After reading out the byte before last from the RIICnDRR register, if the value of the RIICnSR2.RDRF flag is confirmed to be 1, write 1 to the RIICnCR2.SP bit (stop condition issuance request) and then read the last byte from RIICnDRR. When RIICnDRR is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCL line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00_B and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the RIICnSR2.STOP flag to 1.

(9) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

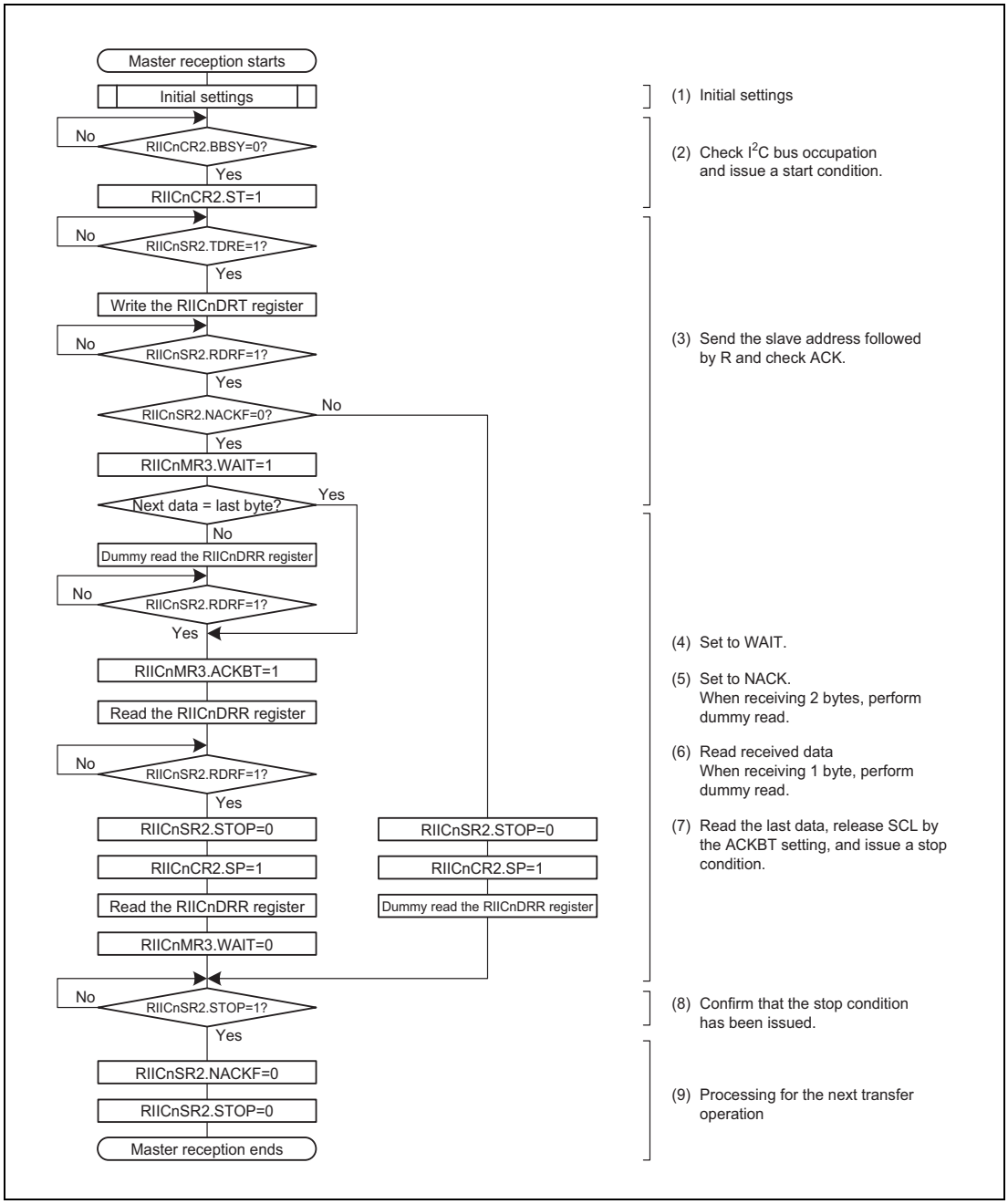


Figure 21.10 Example of Master Reception Flowchart (7-Bit Address Format, 1 or 2 Bytes)

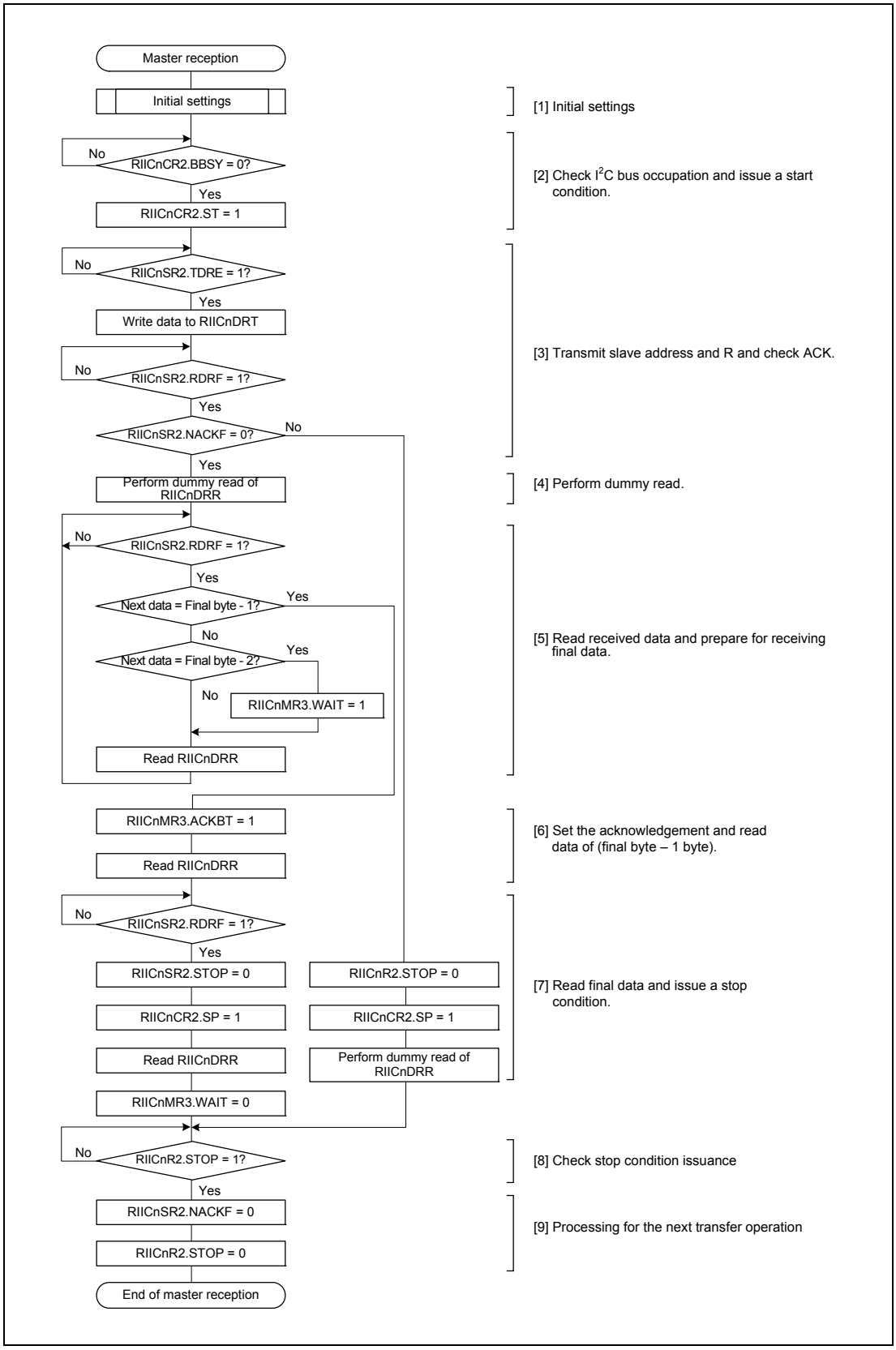


Figure 21.11 Example of Master Reception Flowchart (7-Bit Address Format, 3 Bytes or More)

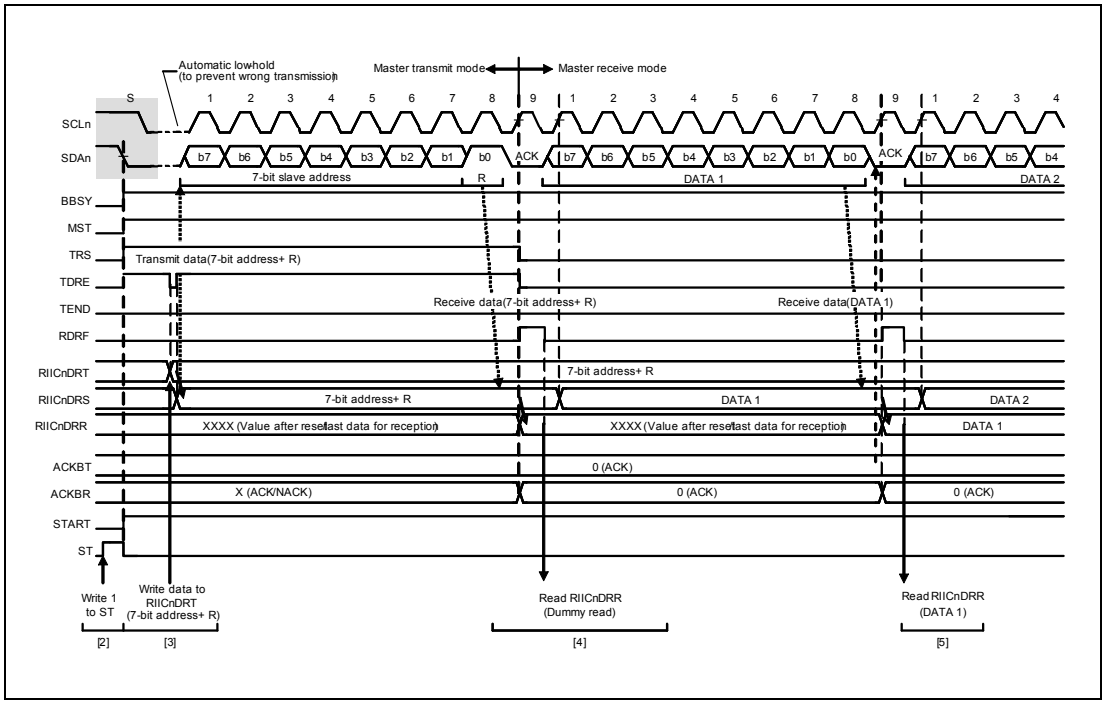


Figure 21.12 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

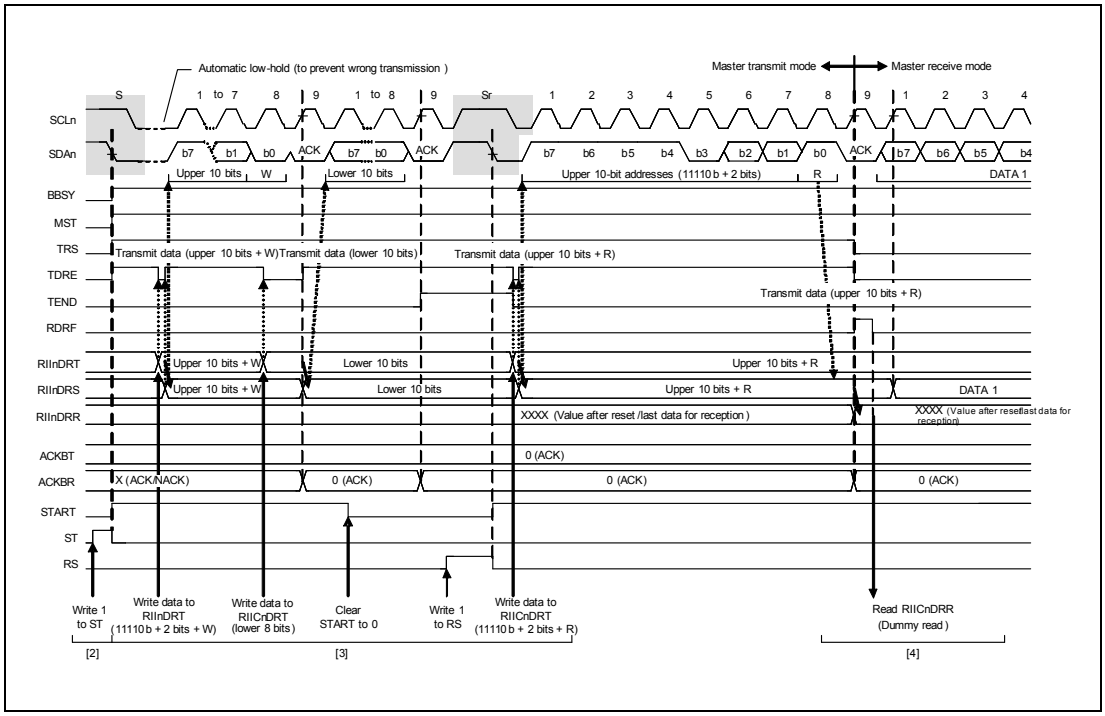


Figure 21.13 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)

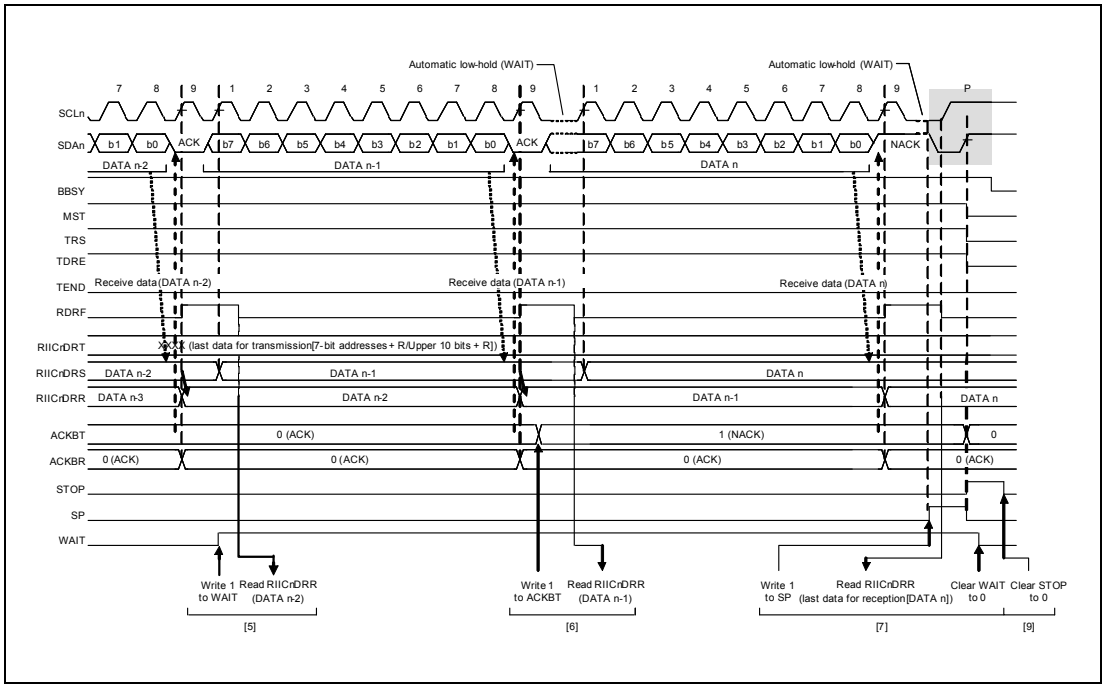


Figure 21.14 Master Receive Operation Timing (3) (when RDRFS = 0)

21.5.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL (clock) signal, the RIIC transmits data as a slave device, and the master device returns acknowledgements.

Figure 21.5 shows an example of usage of slave transmission and **Figure 21.16** and **Figure 21.17** show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Follow the procedure in **Figure 21.5** to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC automatically sets the BBSY and RIICnSR2.START flags to 1 and automatically clears the ST bit to 0 on detection of a start condition.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1.GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the RIICnCR2.TRS bit and the RIICnSR2.TDRE flag to 1.

(3) After the RIICnSR2.TDRE flag is confirmed to be 1, write the data for transmission to the RIICnDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives an NACK signal) while the RIICnFER.NACKE bit is 1, the RIIC suspends transfer of the next data.

(4) Wait until the RIICnSR2.TEND flag is set to 1 while the RIICnSR2.TDRE flag is 1, after the RIICnSR2.NACKF flag is set to 1 or the last byte for transmission is written to the RIICnDRT register. When the RIICnSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCL line low on the ninth falling edge of SCL clock.

(5) When the RIICnSR2.NACKF flag or the RIICnSR2.TEND flag is 1, dummy read RIICnDRR to complete the processing. This releases the SCL line.

(6) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.GCA, and AASy (y = 0 to 2), flags RIICnSR2.TDRE and TEND, and the RIICnCR2.TRS bit to 0, and enters slave receive mode. Furthermore, it automatically sets the RIICnSR2.STOP flag to 1.

(7) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

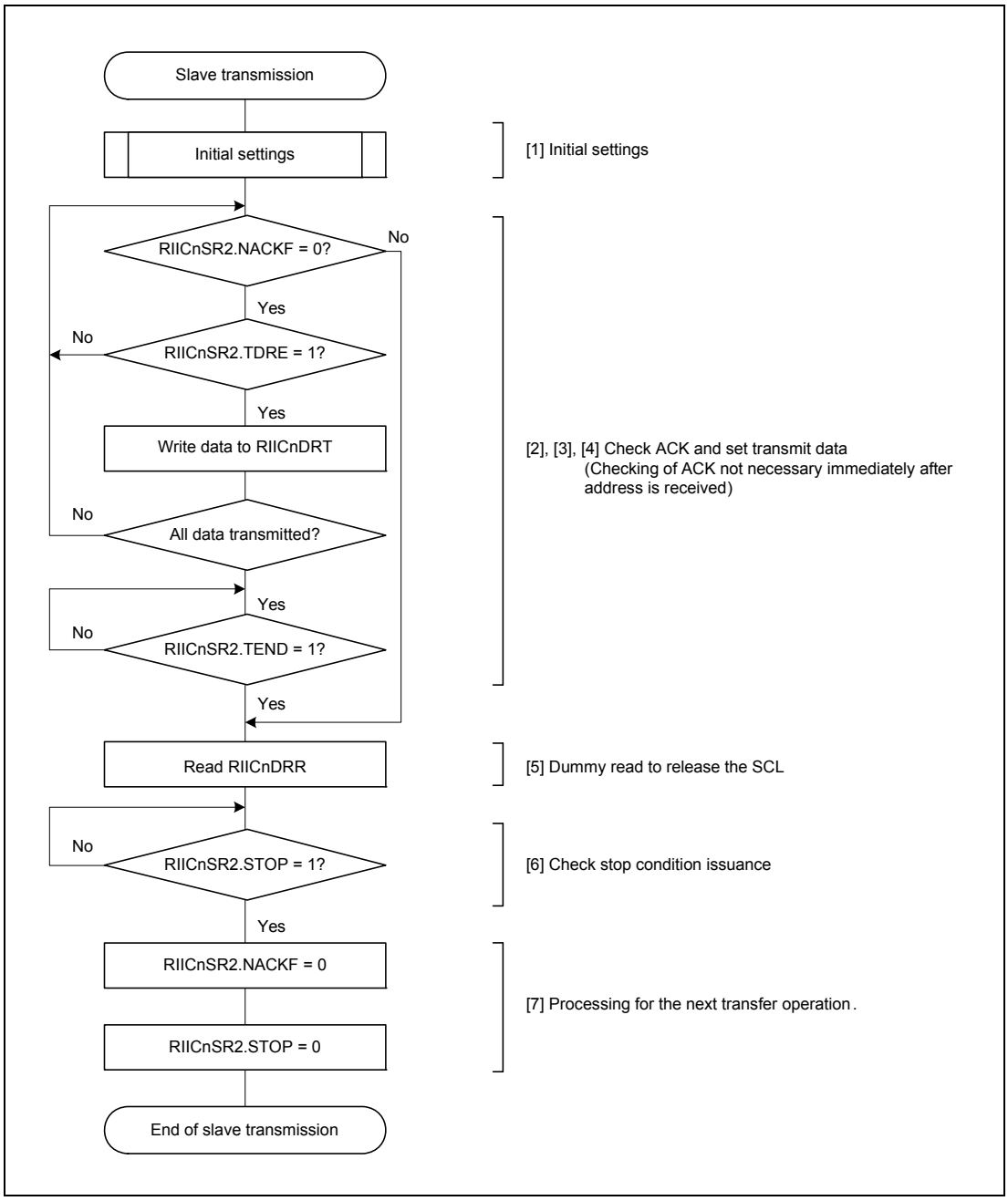


Figure 21.15 Example of Slave Transmission Flowchart

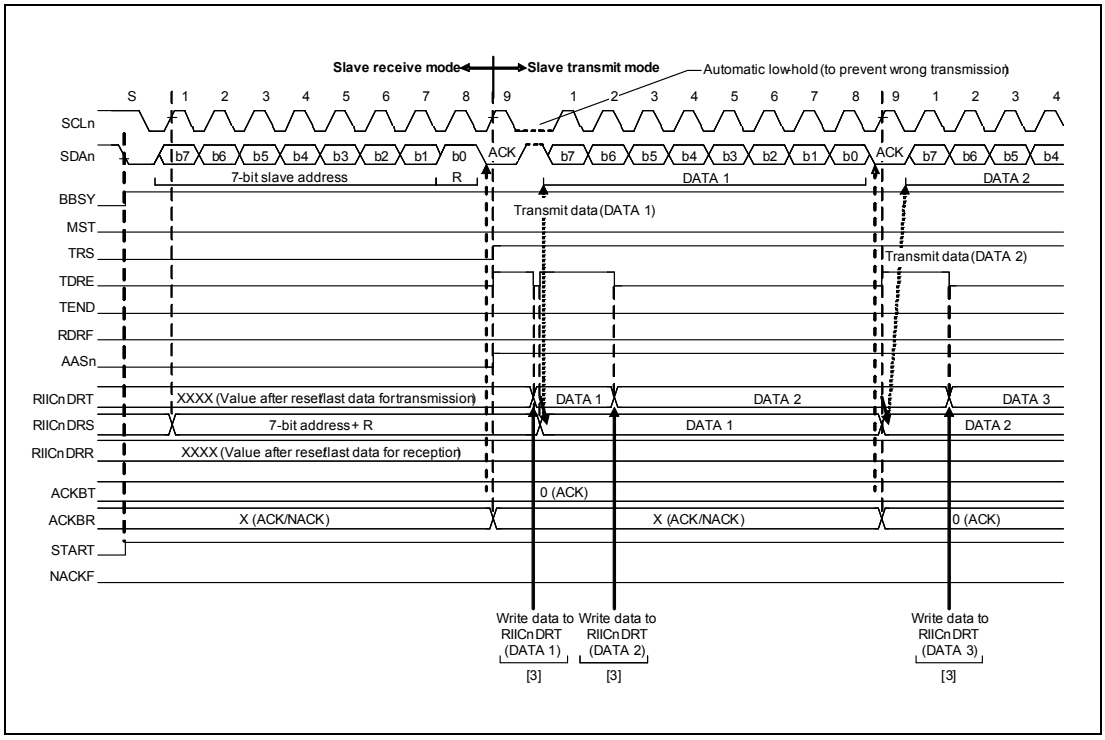


Figure 21.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

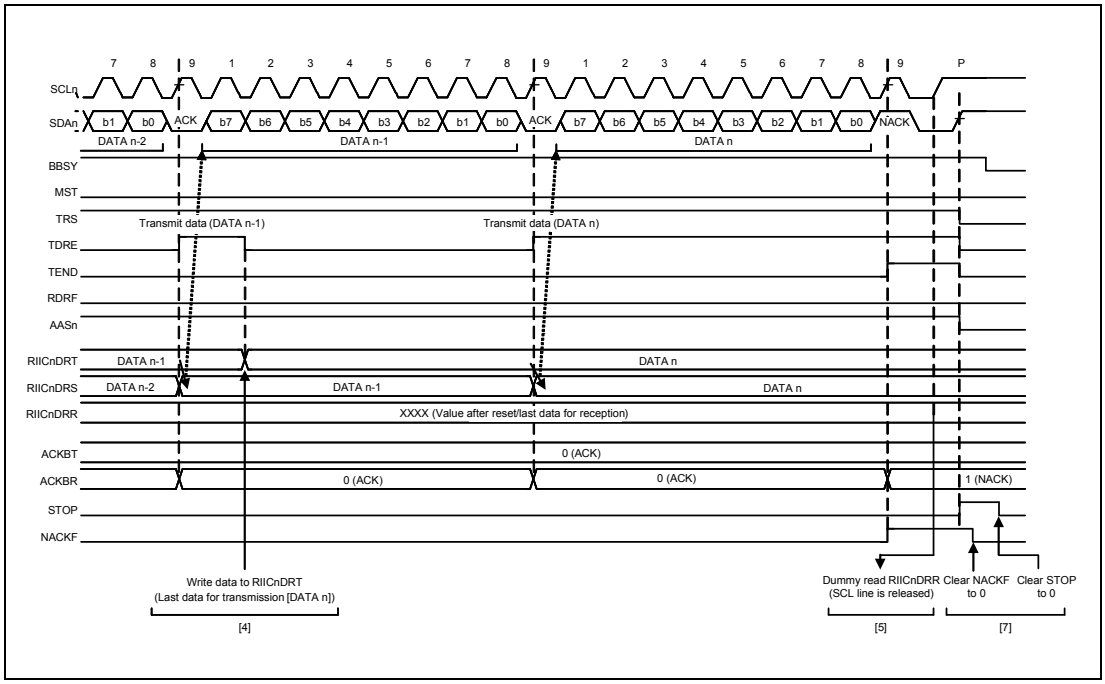


Figure 21.17 Slave Transmit Operation Timing (2)

21.5.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgements as a slave device.

Figure 21.18 shows an example of usage of slave reception and **Figure 21.19** and **Figure 21.20** show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Follow the procedure in **Figure 21.5** to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC automatically sets the BBSY and RIICnSR2.START flags to 1 and automatically clears the ST bit to 0 on detection of a start condition.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1.GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the RIICnSR2.RDRF flag to 1.

(3) After the RIICnSR2.STOP flag is confirmed to be 0 and the RIICnSR2.RDRF flag to be 1, dummy read RIICnDRR as the first read operation (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower eight bits when the 10-bit address format is selected).

(4) When RIICnDRR is read, the RIIC automatically clears the RIICnSR2.RDRF flag to 0. If reading of RIICnDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading RIICnDRR releases the SCL line from being held at the low level.
When the RIICnSR2.STOP flag is 1 and the RIICnSR2.RDRF flag is also 1, read RIICnDRR until all the data is completely received.

(5) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.GCA, and AASy (y = 0 to 2) to 0. Furthermore, it automatically sets the RIICnSR2.STOP flag to 1.

(6) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.STOP flag to 0 for the next transfer operation.

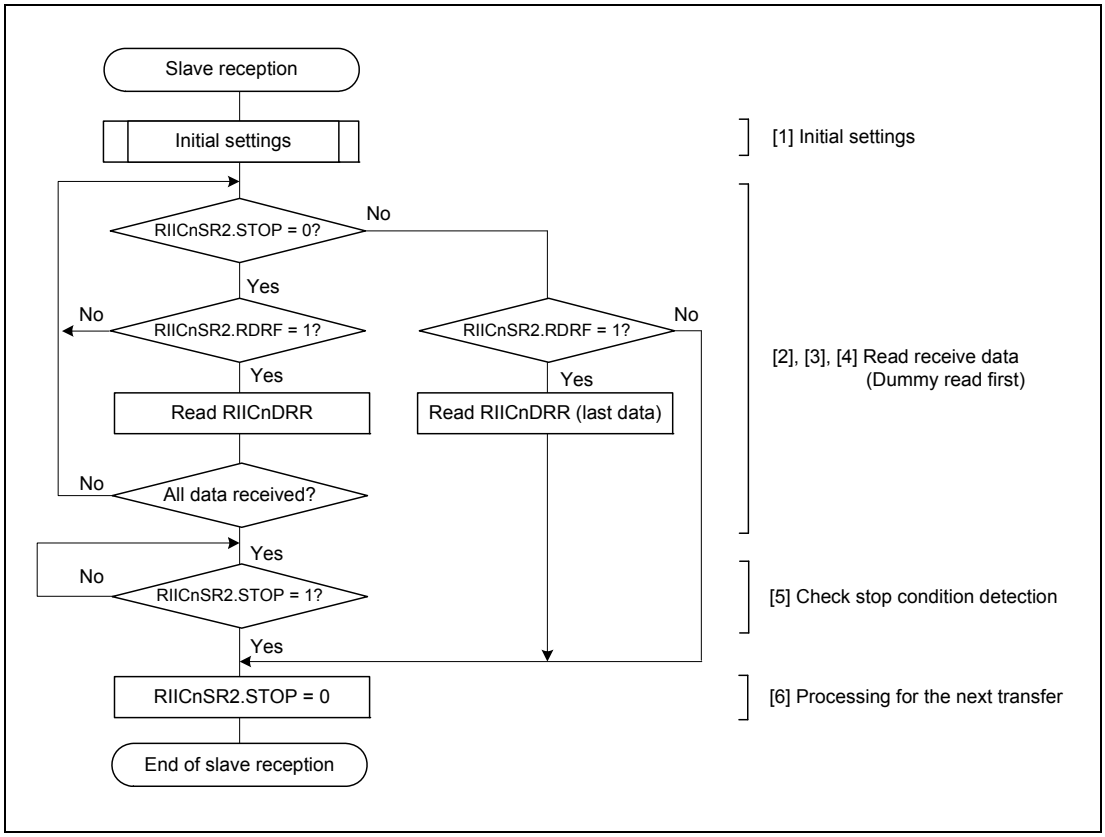


Figure 21.18 Example of Slave Reception Flowchart

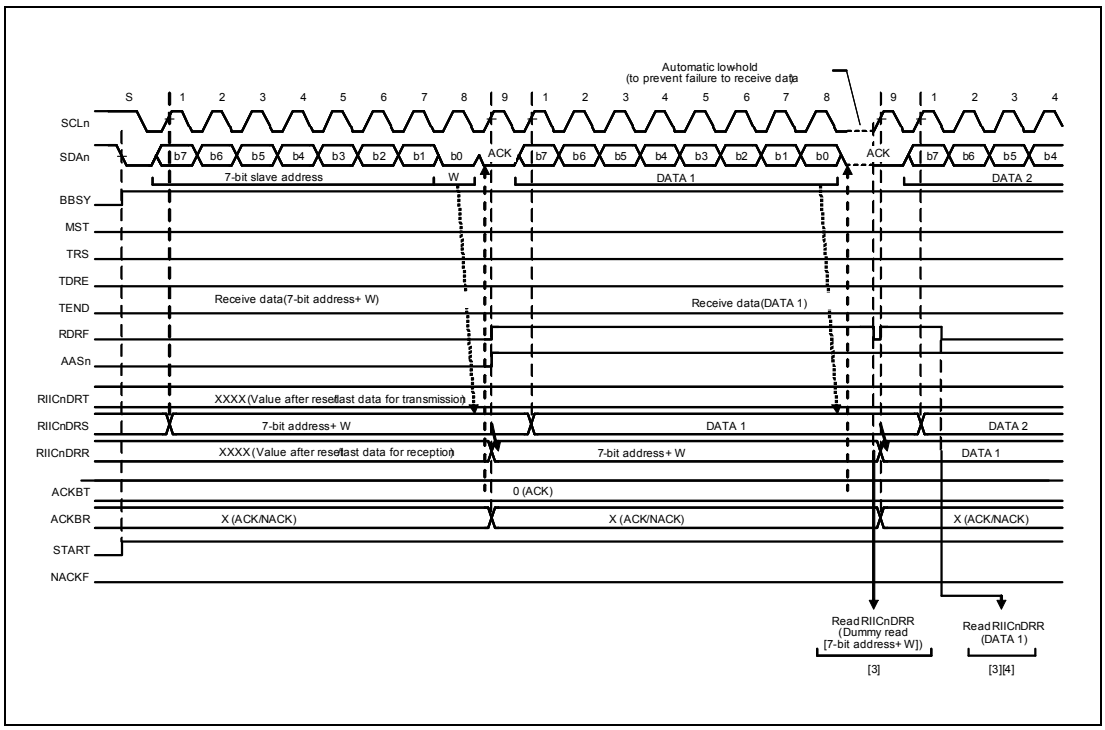


Figure 21.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

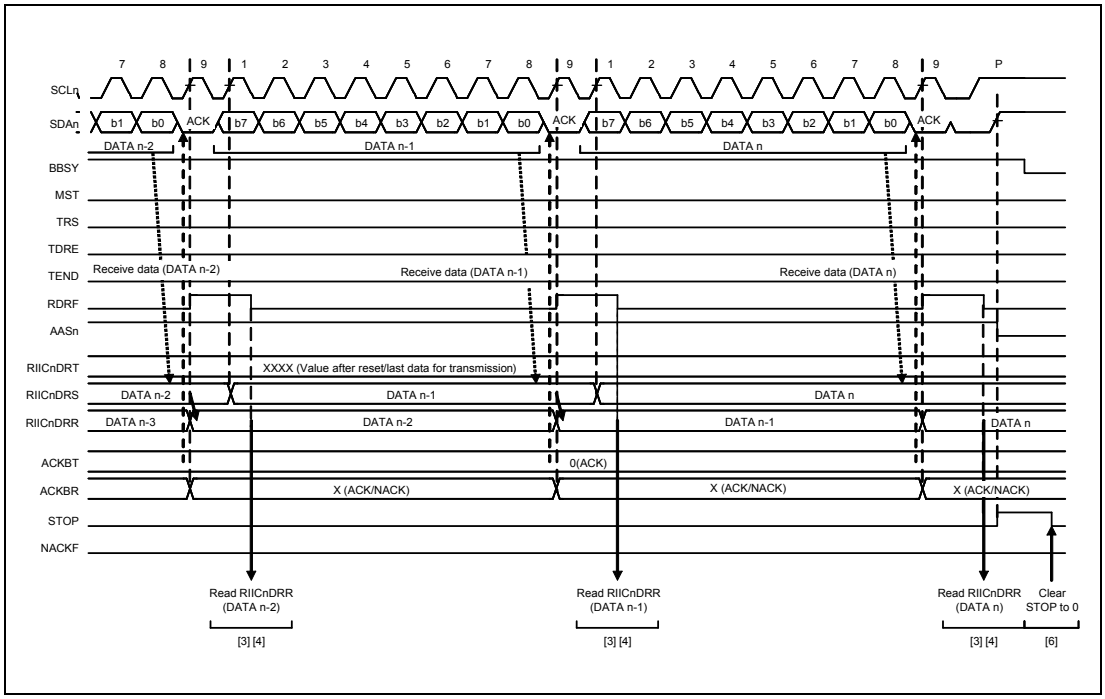


Figure 21.20 Slave Receive Operation Timing (2) (when RDRFS = 0)

21.6 SCL Synchronization Circuit

In generation of the SCL (clock) signal, the RIIC starts counting out the value for width at high level specified in RIICnBRH when it detects a rising edge on the SCL line and drives the SCL line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL line, it starts counting out the width at low level period specified in RIICnBRL, and then stops driving the SCL line (releases the line) once counting of the width at low level is complete. The SCL (clock) signal is thus generated.

If multiple master devices are connected to the I2C bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCL line during communication.

When the RIIC has detected a rising edge on the SCL line and thus started counting out the width at high level specified in RIICnBRH, and the level on the SCL line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL line low, and starts counting out the width at low level specified in RIICnBRL. When the RIIC finishes counting out the width at low level, it stops driving the SCL line to the low level (i.e. releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL line has been released. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the RIICnFER.SCLE bit is set to 1.

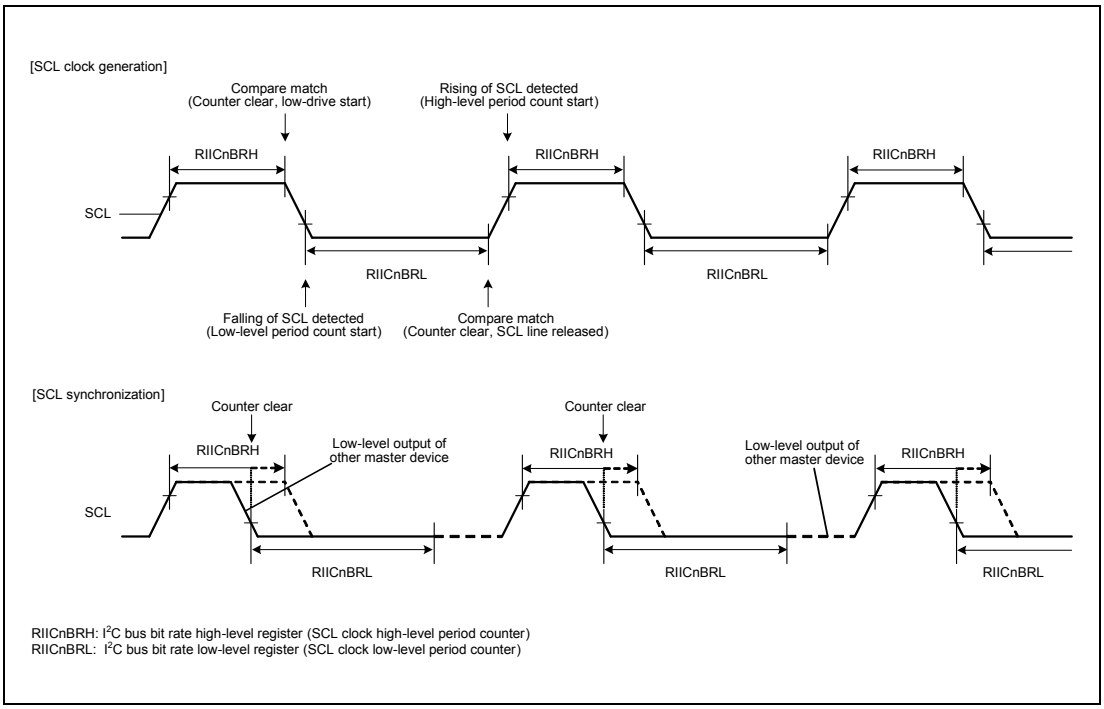


Figure 21.21 Generation and Synchronization of the SCL Signal from the RIIC

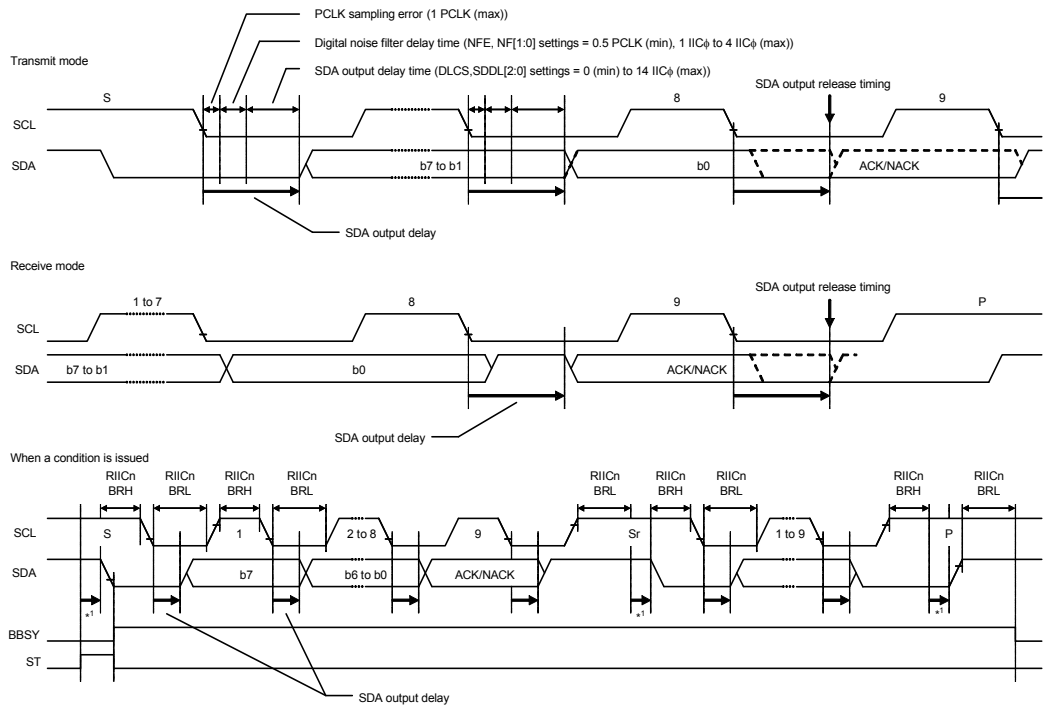
21.7 Facility for Delaying SDA Output

The RIIC module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL (clock) signal is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices.

The output delay facility is enabled by setting the RIICnMR2.SDDL[2:0] bits to any value other than 000_B, and disabled by setting the same bits to 000_B.

While the SDA output delay facility is enabled (i.e. while the SDDL[2:0] bits are set to any value other than 000_B), the RIICnMR2.DLCS bit selects the clock source for counting by the SDA output delay counter as the internal base clock (IICϕ) for the RIIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IICϕ/2). The counter counts the number of cycles set in the SDDL[2:0] bits. After counting of the set number of cycles of delay is completed, the RIIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.



Note: The output delay facility is set by the SDDL[2:0] bits when a start (S), restart (Sr), or stop (P) condition is issued.

Figure 21.22 SDA Output Delay Facility

21.8 Digital Noise-Filter Circuits

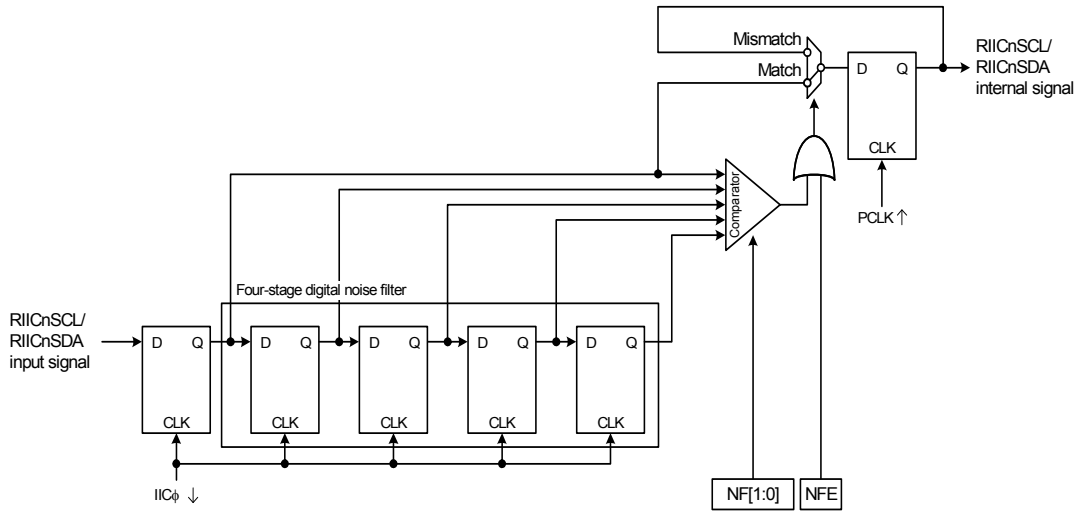
The states of the RIICnSCL and RIICnSDA pins are conveyed to the internal circuitry through the digital noise-filter circuit. **Figure 21.23** is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the RIIC consists of four flip-flop circuit stages connected in series and a match-detection circuit.

The number of effective stages in the digital noise filter is selected by the RIICnMR3.NF[1:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to four IICϕ cycles.

The input signal to the RIICnSCL pin (or RIICnSDA pin) is sampled on falling edges of the IICϕ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the RIICnMR3.NF[1:0] bits, the signal level is conveyed as an internal signal. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is small, the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, it is possible to disable the digital noise-filter circuit (by clearing the RIICnFER.NFE bit to 0).



Note: NFE: Digital noise filter circuit enable bit
NF[1:0]: Digital noise filter stage selection bit

Figure 21.23 Block Diagram of Digital Noise Filter Circuit

21.9 Address Match Detection

The RIIC can set three unique slave addresses in addition to the general call address and device ID address, and also can set 7-bit or 10-bit slave addresses.

21.9.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the RIICnSER.SARy bit (y = 0 to 2) is set to 1, the slave addresses set in RIICnSARy (y = 0 to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding RIICnSR1.AASy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle and returns the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. The RIICnSR2.RDRF flag or the RIICnSR2.TDRE flag is set to 1 by the following R/W# bit. This causes a receive complete interrupt (INTIICnRI) or transmit data empty interrupt (INTIICnTI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 21.24 to **Figure 21.26** show the AASy flag set timing in three cases.

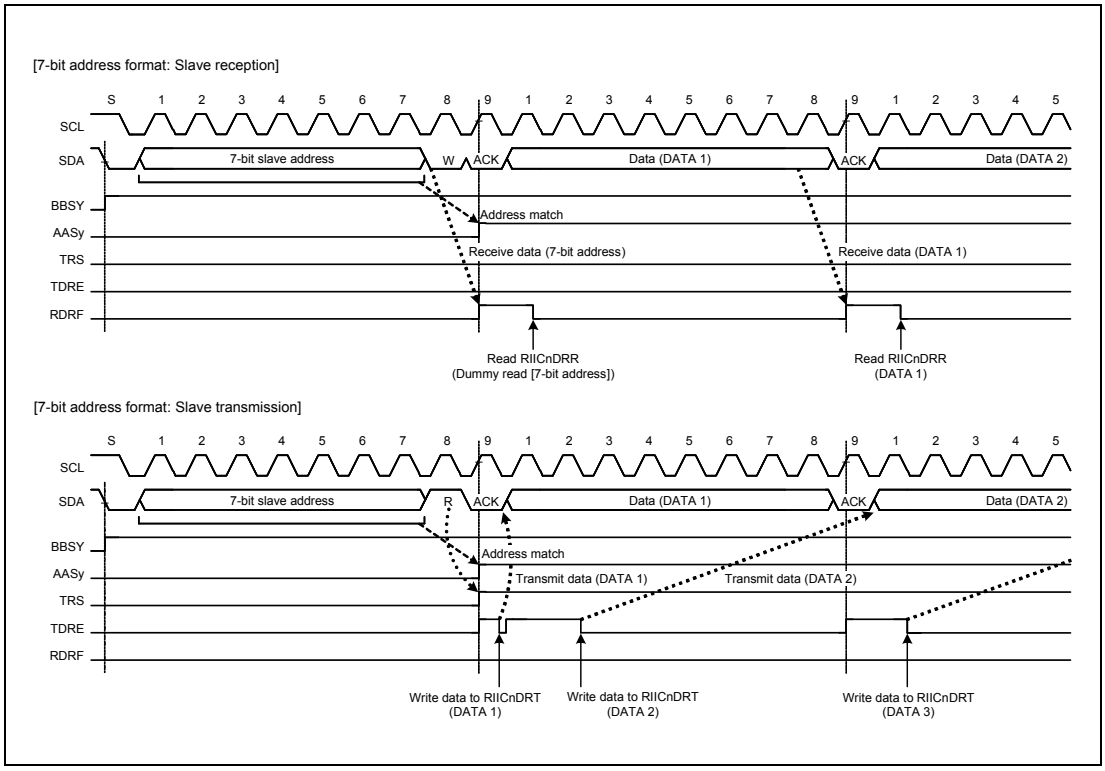


Figure 21.24 AASy Flag Set Timing with 7-Bit Address Format Selected

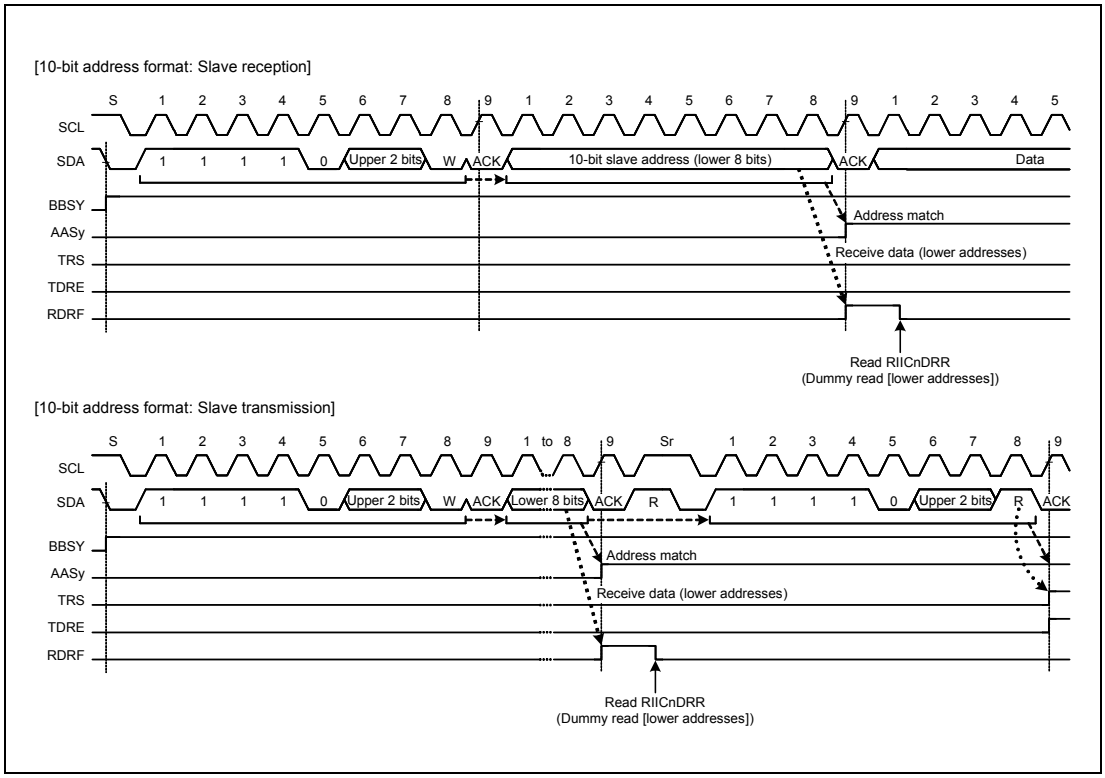


Figure 21.25 AASy Flag Set Timing with 10-Bit Address Format Selected

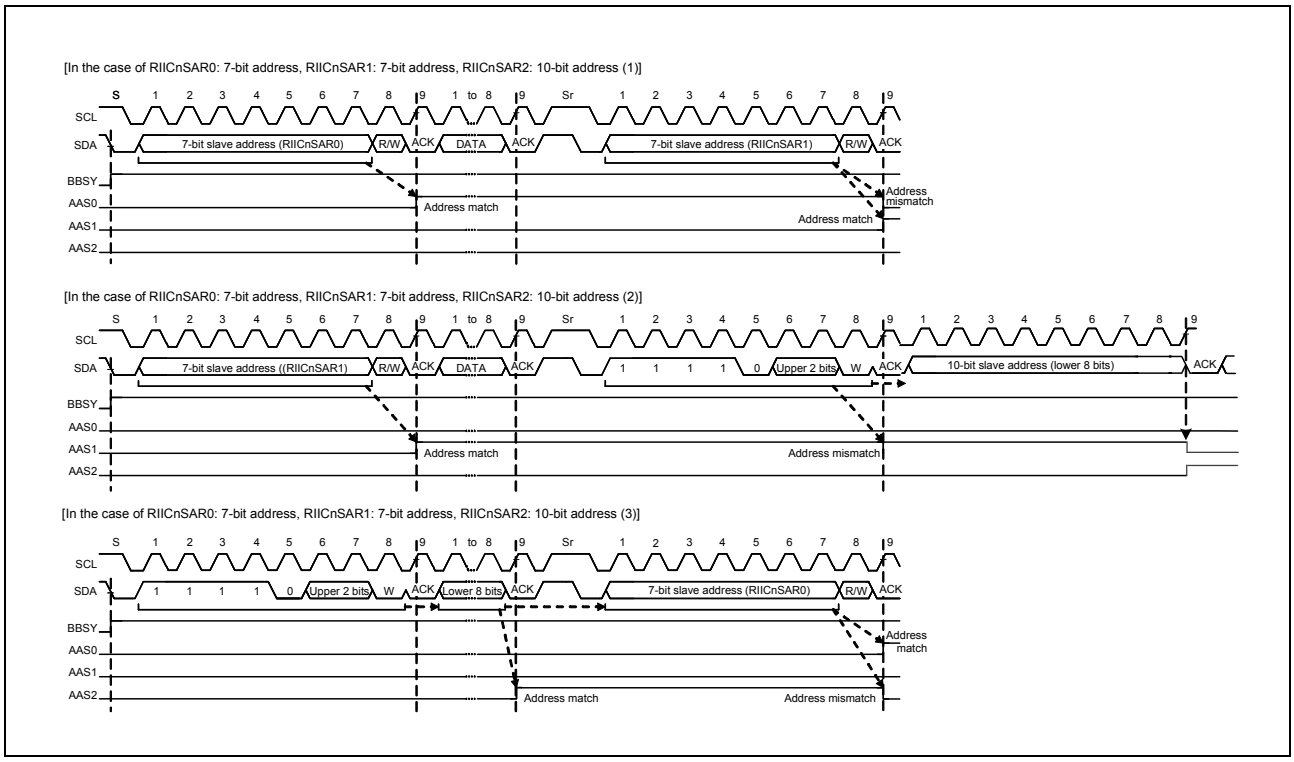


Figure 21.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

21.9.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address ($0000\ 000_B$ + 0 [W]). This is enabled by setting the RIICnSER.GCAE bit to 1.

If the address received after a start or restart condition is issued is 0000 000_B + 1[R] (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the RIICnSR1.GCA flag and the RIICnSR2.RDRF flag are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a receive complete interrupt (INTIICnRI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

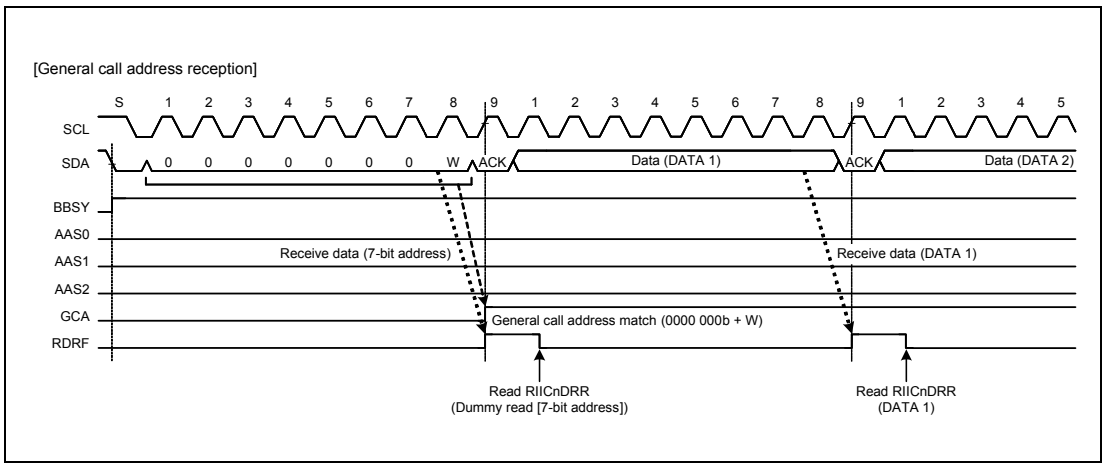


Figure 21.27 Timing of GCA Flag Setting during Reception of General Call Address

21.9.3 Device-ID Address Detection

The RIIC module has a facility for detecting device-ID addresses conforming with the I2C bus specification (Rev. 03). When the RIIC receives 1111 100_B as the first byte after a start condition or restart condition was issued with the RIICnSER.DIDE bit set to 1, the RIIC recognizes the address as a device ID, sets the RIICnSR1.DID flag to 1 on the rising edge of the ninth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding RIICnSR1.AASy flag (y = 0 to 2) to 1.

After that, when the first byte received after a start or restart condition is issued matches the device ID address (1111 100_B) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the RIICnSR2.TDRE flag to 1.

In the device-ID address detection function, the RIIC clears the DID flag to 0 if a match with the RIIC’s own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC’s own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100_B) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC’s slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Furthermore, prepare the device-ID fields (three bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details, see I2C Bus Standard from NXP Semiconductors.

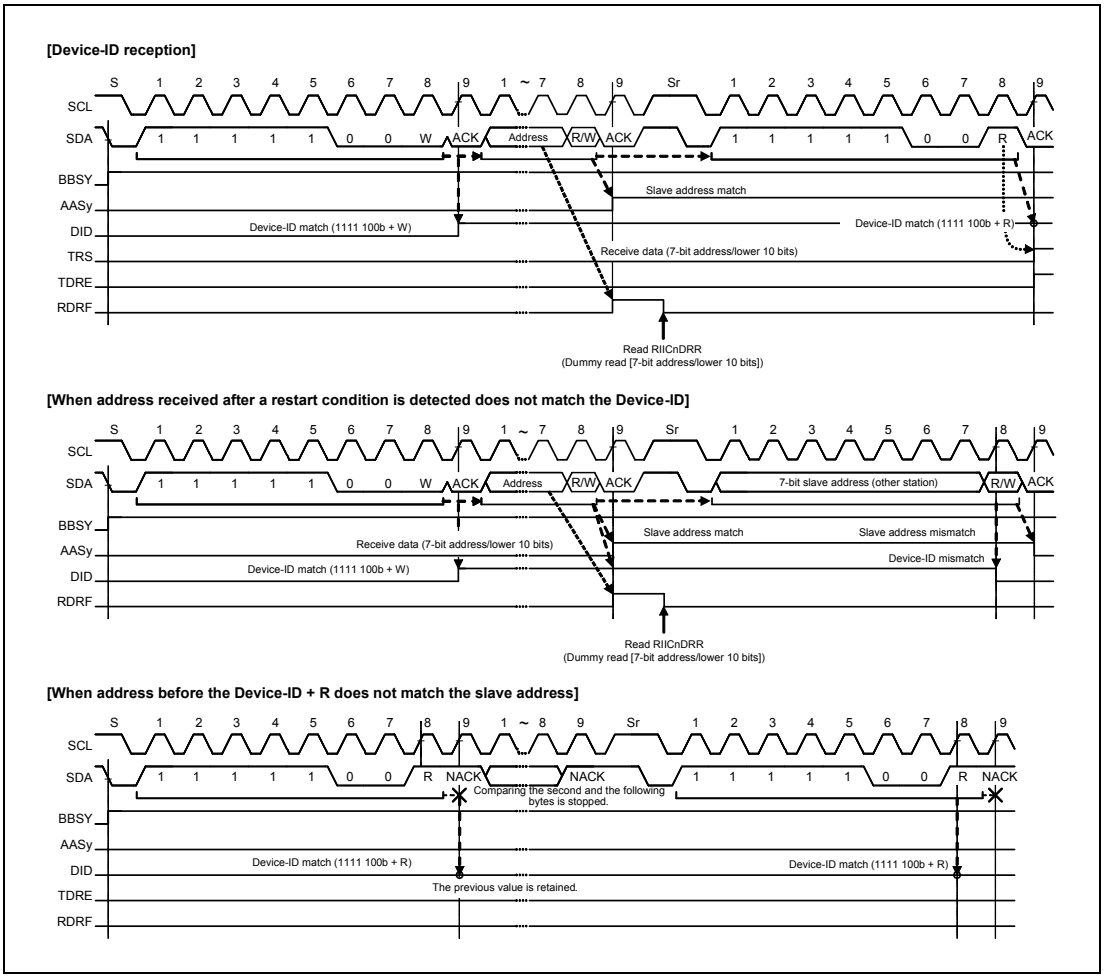


Figure 21.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

21.10 Automatic Low-Hold Function for SCL

21.10.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (RIICnDRS) is empty when data have not been written to the transmit data register (RIICnDRT) with the RIIC in transmission mode (RIICnCR2.TRS bit = 1), the SCL signal is automatically held at the low level over the intervals shown below. To prevent the unintended transmission of erroneous data, this low-hold period is extended until data for transmission have been written. In addition, the RIIC holds the SCL line low over the period until a stop condition is issued and also over the period until the RIICnDRR register is dummy read.

<Master transmit mode>

- Low-level interval after a start condition or restart condition is issued
- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next
- Low-level interval from the ninth clock cycle until a stop condition is issued

<Slave transmit mode>

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next
- Low-level interval from the ninth clock cycle and the RIICnDRR register is dummy read

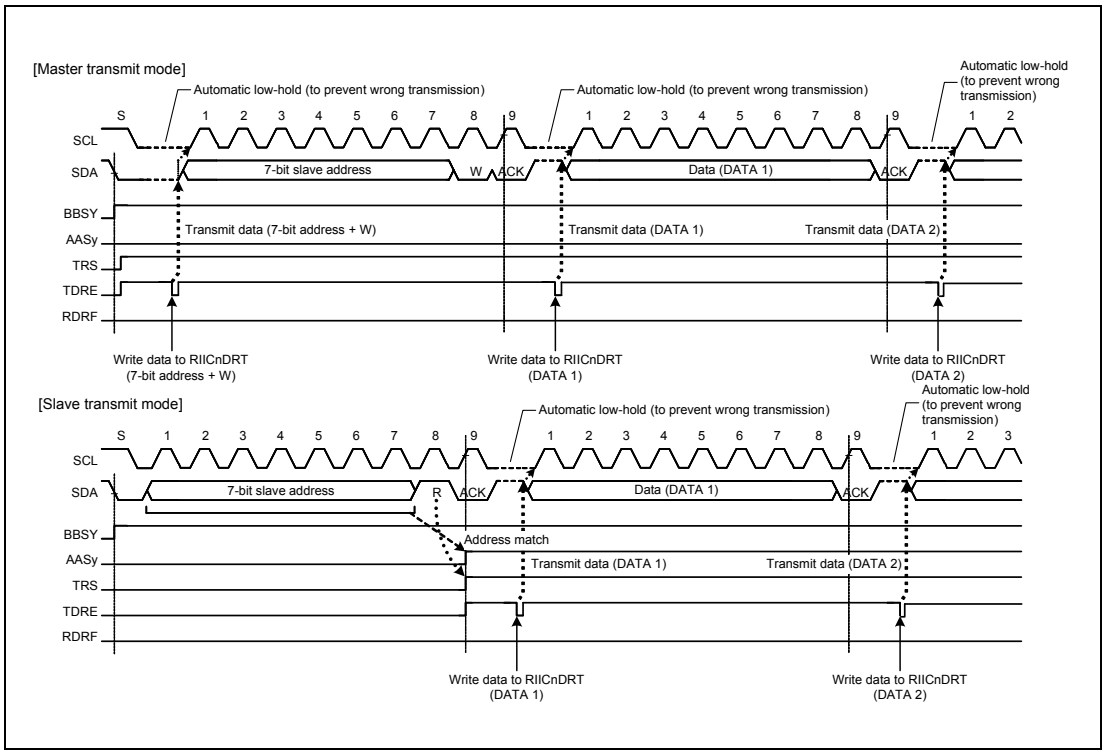


Figure 21.29 Automatic Low-Hold Operation in Transmit Mode

21.10.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (RIICnCR2.TRS bit = 1). This function is enabled when the RIICnFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (RIICnSR2.TDRE flag = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDA line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (RIICnSR2.NACKF flag = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to clear the NACKF flag to 0. In master transmit mode, clear the NACKF flag to 0, issue a restart or stop condition, and then issue a start condition again.

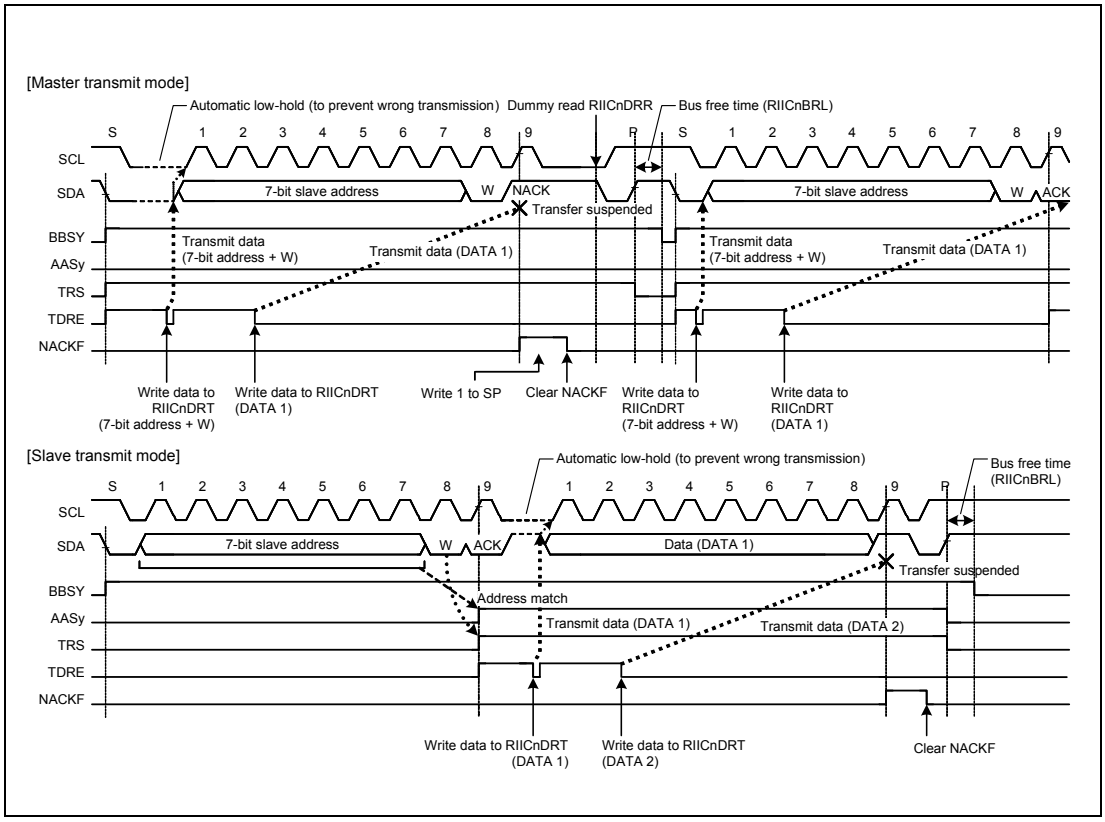


Figure 21.30 Suspension of Data Transfer when NACK is Received (NACKE = 1)

21.10.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (RIICnDRR) read is delayed for a period of one transfer frame or more with receive complete (RIICnSR2.RDRF flag = 1) in receive mode (RIICnCR2.TRS = 0), the RIIC holds the SCL line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is issued. This function does not disturb other communication because the RIIC does not hold the SCL line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCL line is held low can be selected with a combination of the RIICnMR3.WAIT and RDRFS bits.

(1) One-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the RIICnMR3.WAIT bit is set to 1, the RIIC performs one-byte receive operation using the WAIT bit function.
Furthermore, when the RIICnMR3.RDRFS bit is 0, the RIIC automatically sends the RIICnMR3.ACKBT bit value for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCL line low at the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from RIICnDRR, which enables bytewise receive operation.

The WAIT bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and device ID address) is obtained in master receive mode or slave receive mode.

(2) One-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the RIICnMR3.RDRFS bit is set to 1, the RIIC performs one-byte receive operation using the RDRFS bit function.
When the RIICnSR2.RDRFS bit is set to 1, the RDRF flag (receive complete) in RIICnSR2 is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCL line is automatically held low at the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the RIICnMR3.ACKBT bit, but cannot be released by reading data from RIICnDRR, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and device ID address) is obtained in master receive mode or slave receive mode.

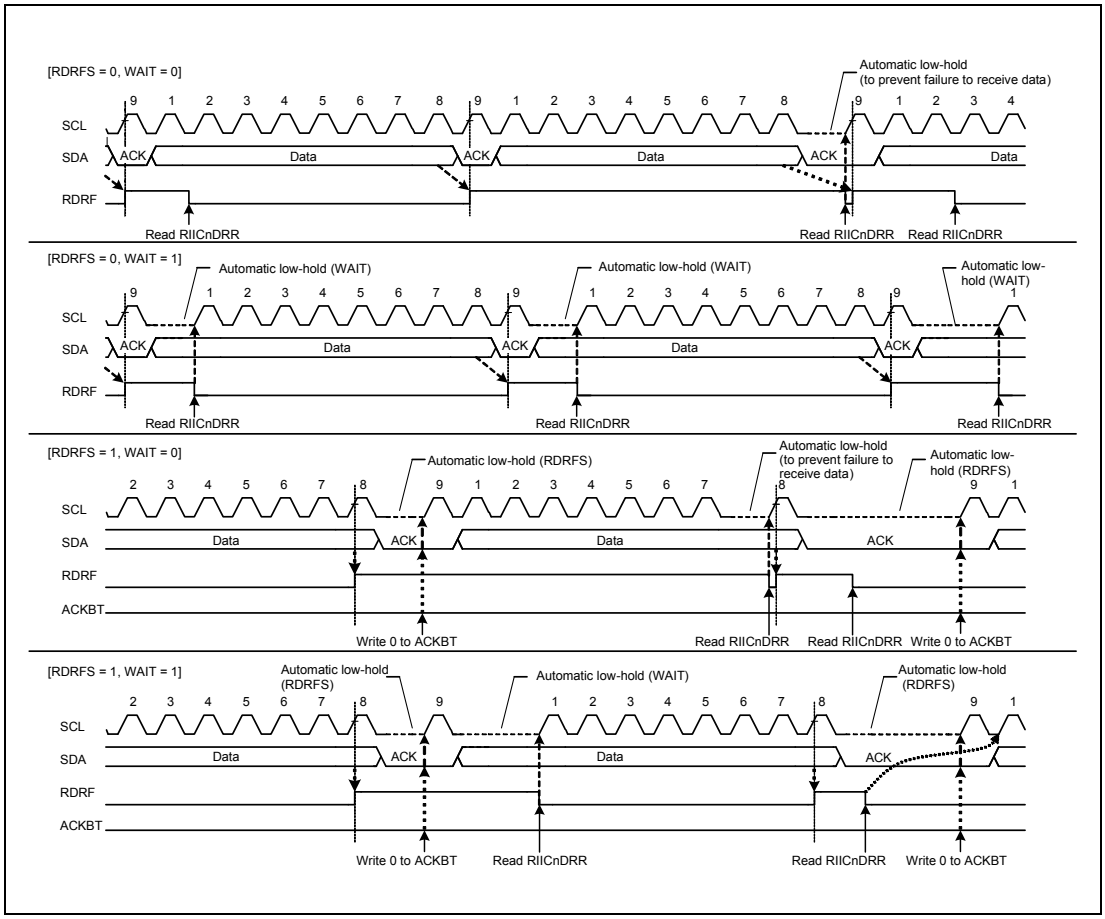


Figure 21.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

21.11 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C bus standard, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

21.11.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA line low to issue a start condition. However, if the SDA line has already been driven low by another master device issuing a start condition, the RIIC considers this a loss in arbitration, so priority is given to transfer by the other master device. Similarly, if the RIICnCR2.ST bit is set to 1 while the bus is busy (RIICnCR2.BBSY flag = 1), the RIIC considers itself to have lost in arbitration, so priority is given to transfer by the other master device and no start condition is generated.

When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the high impedance state, and the low level is detected on the SDA line), the RIIC loses in arbitration.

After a loss in arbitration of mastership, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the RIICnFER.MALE bit is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Non-matching of the internal level for output on SDA and the level on the SDA line after a start condition was issued by setting the RIICnCR2.ST bit to 1 while the RIICnCR2.BBSY flag was cleared to 0 (erroneous issuing of a start condition)
- Setting of the RIICnCR2.ST bit to 1 (start condition double-issue error) while the RIICnCR2.BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA line in master transmit mode (RIICnCR2.MST and TRS bits = 11_B)

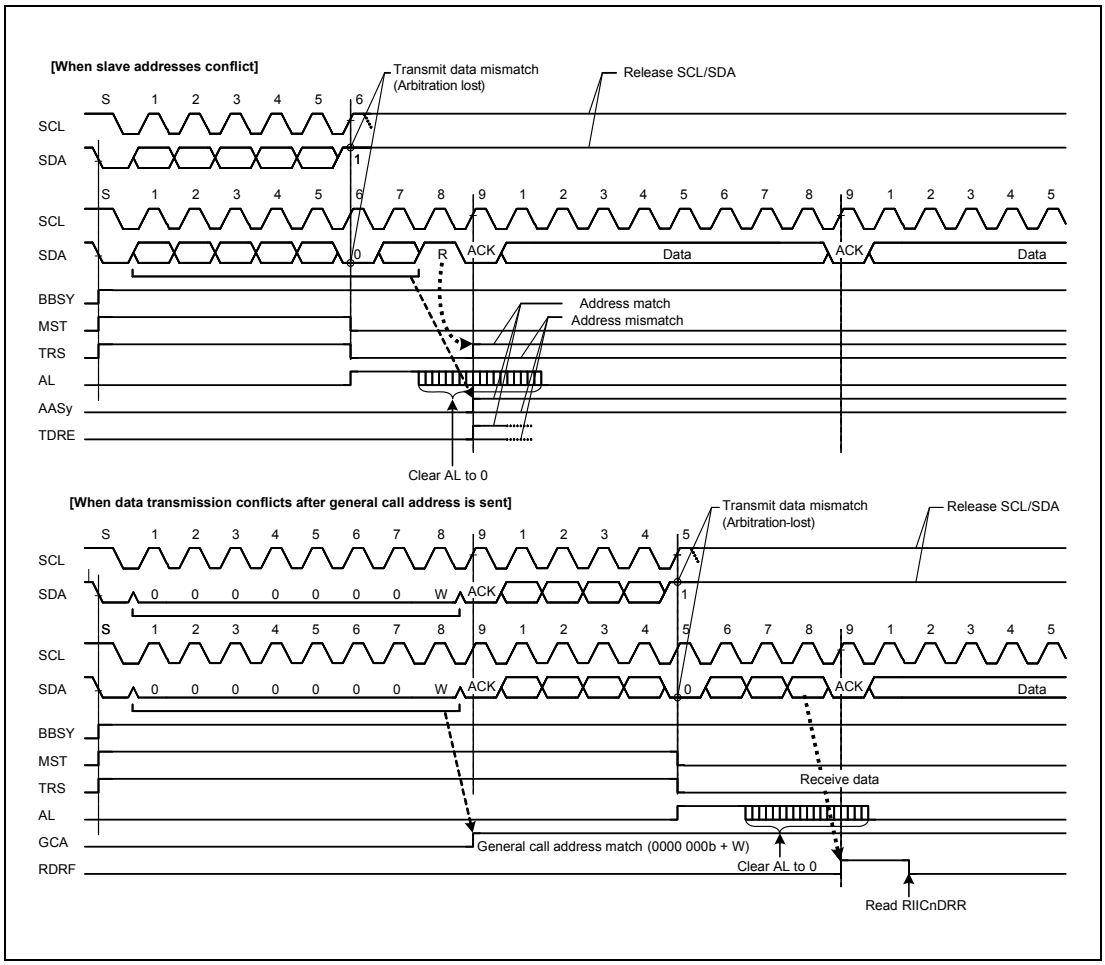


Figure 21.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

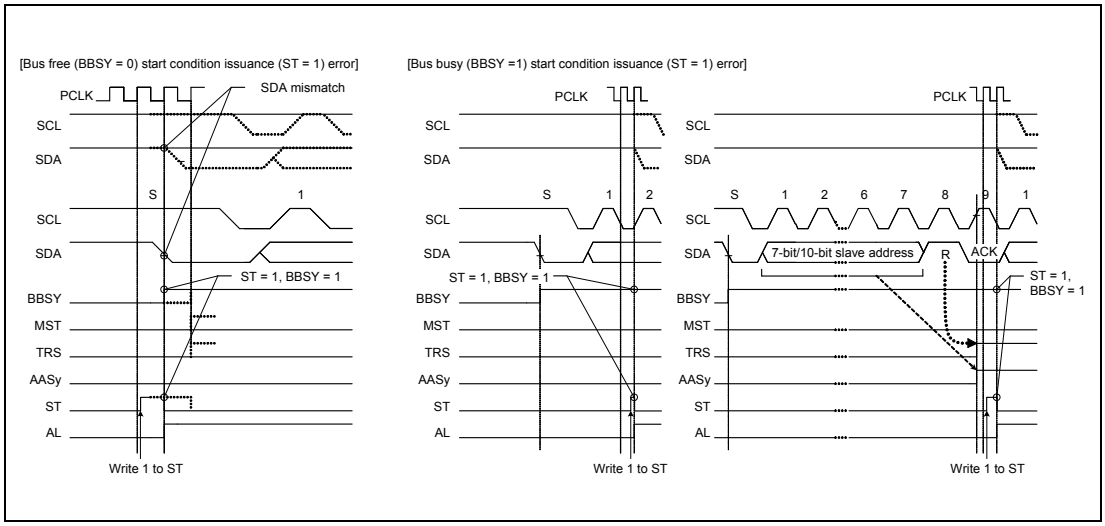


Figure 21.33 Arbitration-Lost when a Start Condition is Issued (MALE = 1)

The TRS bit is not cleared if 1 is written to the ST bit while the BBSY flag is 1 in slave transmit mode.

21.11.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA line (the high output as the internal SDA output; i.e. the SDA pin is in the high impedance state, and the low level is detected on the SDA line) during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. **Figure 21.34** shows an example of arbitration-lost detection during transmission of NACK.

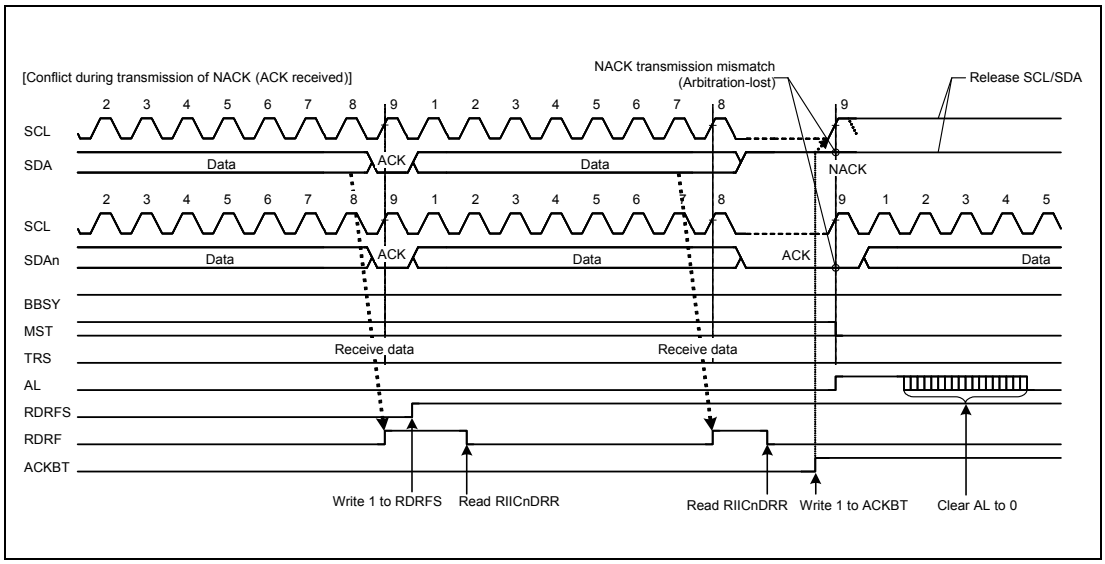


Figure 21.34 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives two bytes of data from the slave device, and master B receives four bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received two final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition.

Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When the RIIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, the RIIC enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

The RIIC detects arbitration-lost during transmission of NACK when the following condition is met with the RIICnFER.NALE bit set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

When the internal SDA output level does not match the SDA line (ACK is received) during transmission of NACK (RIICnMR3.ACKBT bit = 1)

21.11.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the high impedance state, and the low level is detected on the SDA line) in slave transmit mode.

When it loses slave arbitration, the RIIC enters slave receive mode.

The RIIC detects slave arbitration-lost when the following condition is met with the RIICnFER.SALE bit set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

When transmit data excluding acknowledge (internal SDA output level) does not match the SDA line in slave transmit mode (RIICnCR2.MST and TRS bits = 01_B)

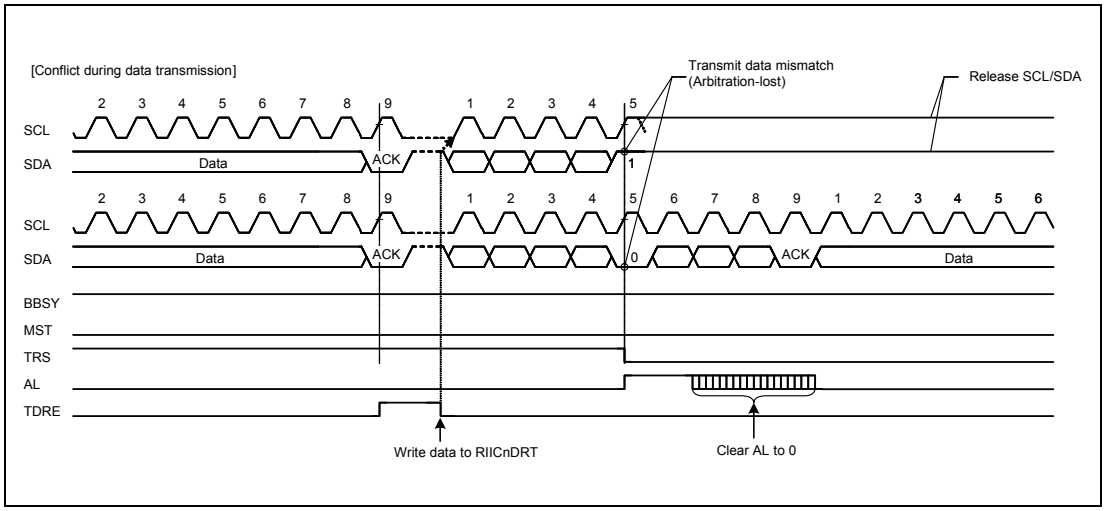


Figure 21.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

21.12 Start Condition/Restart Condition/Stop Condition Issuing Function

21.12.1 Issuing a Start Condition

The RIIC issues a start condition when the RIICnCR2.ST bit is set to 1.

When the ST bit is set to 1, a start condition issuance request is made and the RIIC issues a start condition when the RIICnCR2.BBSY flag is 0 (bus free). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.

A start condition is issued in the following sequence.

[Start condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the start condition hold time.
- Drive the SCL line low (high level to low level).
- Detect low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

21.12.2 Issuing a Restart Condition

The RIIC issues a restart condition when the RIICnCR2.RS bit is set to 1.

When the RS bit is set to 1, a restart condition issuance request is made even during communication and the RIIC issues a restart condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

(To detect the issuance of a restart condition, clear the RIICnSR2.START flag before a restart condition is issued.)

A restart condition is issued in the following sequence.

[Restart condition issuance]

- Release the SDA line.
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRL and the restart condition setup time.
- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the restart condition hold time.
- Drive the SCL line low (high level to low level).
- Detect a low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

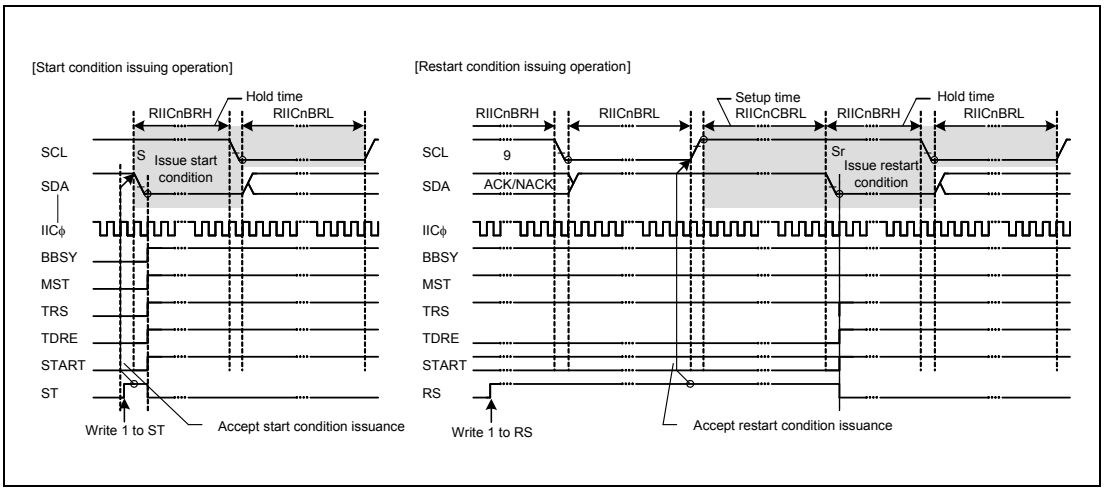


Figure 21.36 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

21.12.3 Issuing a Stop Condition

The RIIC issues a stop condition when the RIICnCR2.SP bit is set to 1.

When the SP bit is set to 1, a stop condition issuance request is made and the RIIC issues a stop condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRH and the stop condition setup time.
- Release the SDA line (low level to high level).
- Ensure the time set in RIICnBRL and the bus free time.
- Clear the BBSY flag to 0 (to release the bus mastership).

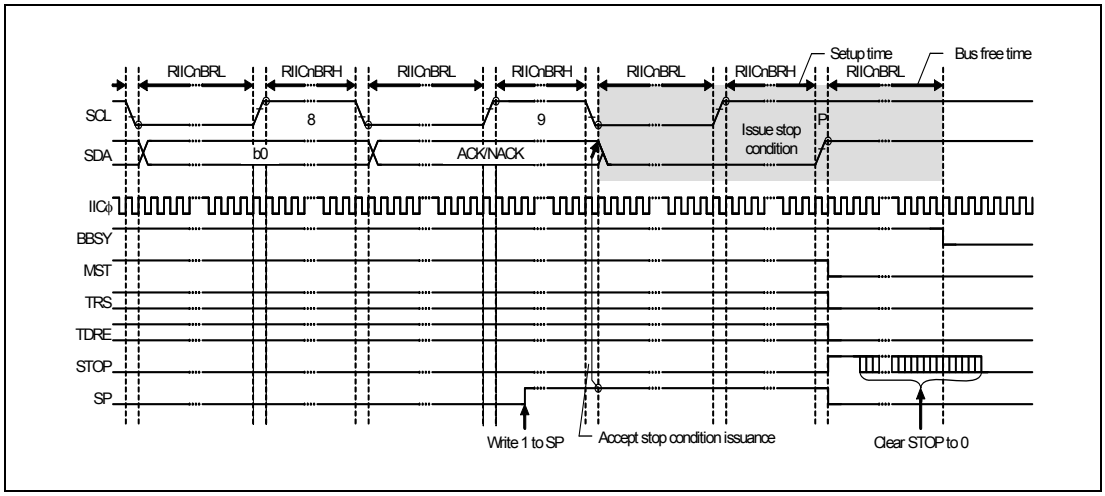


Figure 21.37 Stop Condition Issue Timing (SP Bit)

21.13 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I2C bus might hang with a fixed level on the SCL line and/or SDA line.

As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, and the RIIC/internal reset function.

By checking the RIICnCR1.SCLO, SDAO, SCLI, and SDAI bits, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL or SDA lines.

21.13.1 Timeout Function

The RIIC has the timeout function to detect an abnormality that the SCL line is held for a certain period of time. The RIIC can detect an abnormal bus state by monitoring that the SCL line is held low or high for a predetermined time.

The timeout function monitors the SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL line changes (rising or falling), but continues to count unless the SCL line changes. If the internal counter overflows due to no SCL line change, the RIIC can detect the timeout and report the bus abnormality.

The internal counter is cleared when one of the conditions is met.

(1) When RIICnMR2.TMOH=0, and RIICnMR2.TMOL=1:
The internal counter is cleared by SCL rising

(2) When RIICnMR2.TMOH=1, and RIICnMR2.TMOL=0:
The internal counter is cleared by SCL falling

(3) When RIICnMR2.TMOH=RIICnMR2.TMOL=1:
The internal counter is cleared by SCL rising or falling

This timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state that the SCL line is stuck low or high during the following conditions:

- The bus is busy (RIICnCR2.BBSY flag is 1) in master mode (RIICnCR2.MST bit is 1).
- The RIIC's own slave address matches (RIICnSR1 register is not 00_H) and the bus is busy (RIICnCR2.BBSY flag is 1) in slave mode (RIICnCR2.MST bit is 0).
- The bus is free (RIICnCR2.BBSY flag is 0) while generation of a START condition is requested (RIICnCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IICϕ) set by the RIICnMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (RIICnMR2.TMOS bit = 0) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCL line level (low/high or both levels) during which this counter is activated can be selected by the setting of the RIICnMR2.TMOH and TMOL bits. If both TMOL and TMOH bits are cleared to 0, the internal counter does not work.

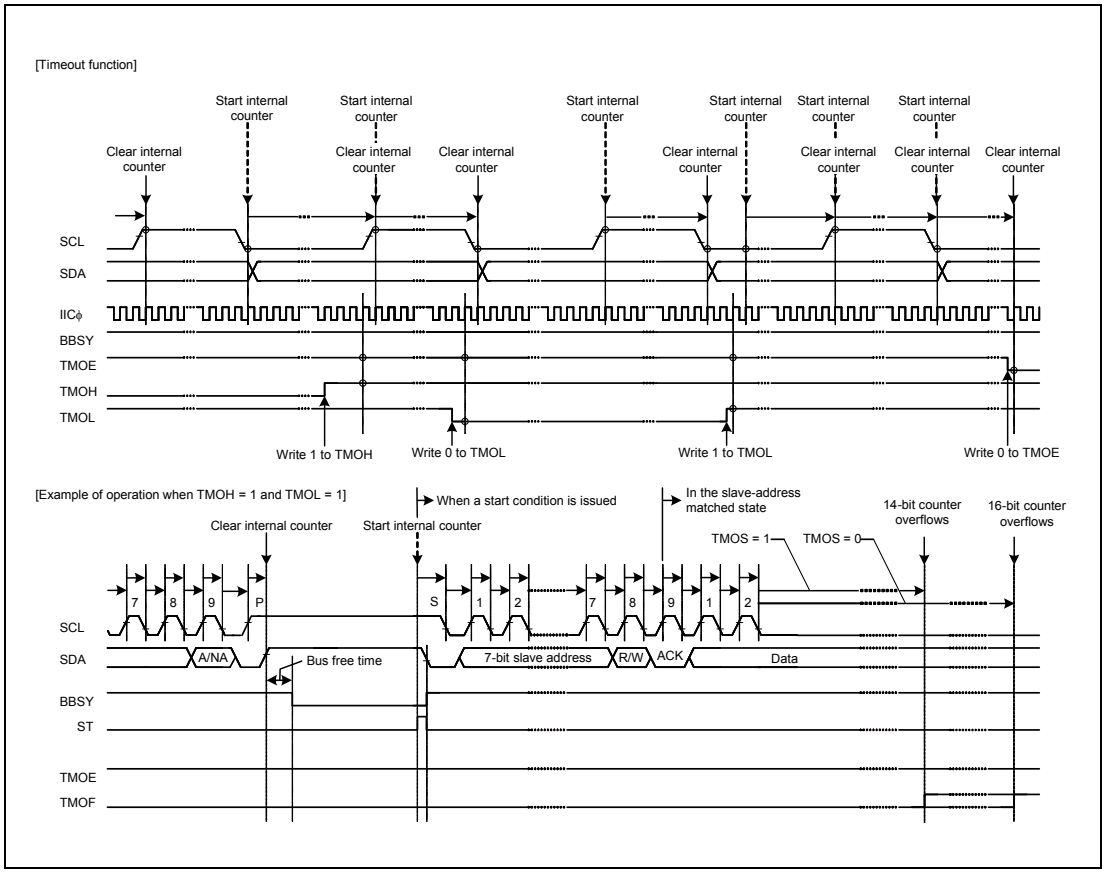


Figure 21.38 Timeout Function (TMOE, TMOS, TMOH, and TMOL Bits)

21.13.2 Extra SCL Clock Cycle Output Function

In master mode, the RIIC module has a facility for the output of extra SCL (clock) cycles to release the SDA line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the SDA line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from the RIIC with single cycles of the SCL (clock) signal as the unit in the case of a bus error where the RIIC cannot issue a stop condition because the slave device is holding the SDA line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the RIICnCR1.CLO bit is set to 1 in master mode, a single cycle of the SCL clock at the frequency corresponding to the transfer rate settings (settings of the RIICnMR1.CKS[2:0] bits, and of the RIICnBRH and RIICnBRL registers) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically cleared to 0. Therefore, further extra clock cycles can be output consecutively by the software program writing 1 to the CLO bit after having read CLO = 0.

When the RIIC module is in master mode and the slave device is holding the SDA line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The facility for output of an extra cycle of the SCL (clock) signal can be used to output extra cycles of SCL one by one to make the slave device release the SDA line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDA line by the slave device can be monitored by reading the RIICnCR1.SDAI bit. After confirming release of the SDA line by the slave device, complete communications by reissuing the stop condition.

Use this facility with the RIICnFER.MALE bit (master arbitration-lost detection disabled) cleared to 0. If the MALE bit is set to 1 (master arbitration-lost detection enabled), arbitration is lost when the value of the RIICnCR1.SDAO bit does not match the state of the SDA line, so take care on this point.

[Additional output conditions for the SCL clock]

- In master mode and when the bus is free
- In master mode and the SCL line is not held low (the bus is busy)

Figure 21.39 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

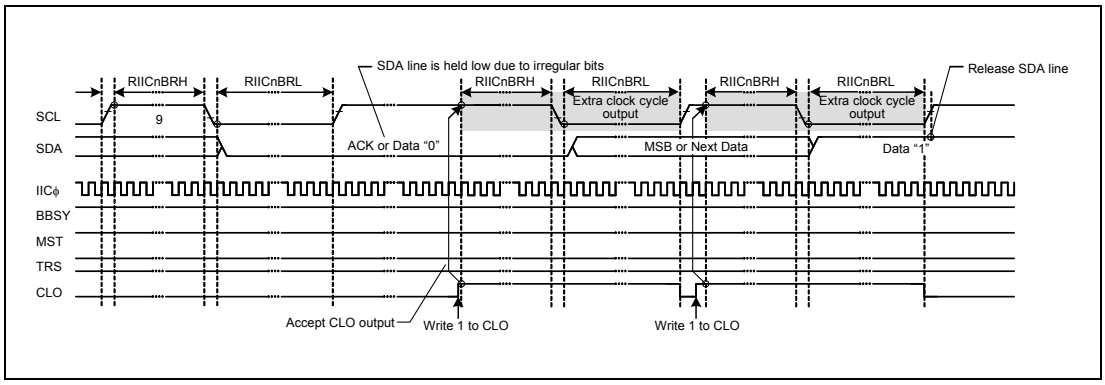


Figure 21.39 Extra SCL Clock Cycle Output Function (CLO Bit)

21.13.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the RIICnCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings.

After issuing a reset, be sure to clear the RIICnCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states since both restore the output state of the SCL and SDA pins to the high impedance state.

Issuing a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (RIICnCR1.ICE and IICRST bits = 01_B).

For a detailed description of the RIIC and internal resets, see Section 21.14, Reset Function of RIIC.

21.14 Reset Function of RIIC

The RIIC has RIIC reset, and internal reset functions. In addition RIIC is cleared by ISORES. **Table 21.26** lists the scope of each reset and reset conditions.

Table 21.26 RIIC Reset Functions (1/2)

UM		SYSRES/ wake-up from DEEPSTOP mode	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start/Restart Condition Detection	Stop Condition Detection
RIICnCR1	ICE	Initialized	0	1	Retained	Retained
	IICRST	Initialized	1	1	Retained	Retained
	CLO	Initialized	Initialized	Retained	Retained	Retained
	SOWP	Initialized	Initialized	Retained	Retained	Retained
	SCLO	Initialized	Initialized	Initialized	Retained	Retained
	SDAO	Initialized	Initialized	Initialized	Retained	Retained
	SCLI	Initialized	Initialized	Retained	Retained	Retained
	SDAI	Initialized	Initialized	Retained	Retained	Retained
RIICnCR2	BBSY	Initialized	Initialized	Initialized *1	Operation	Retained
	MST	Initialized	Initialized	Initialized	Operation (retained)	Initialized
	TRS	Initialized	Initialized	Initialized	Operation (retained)	Initialized
	SP	Initialized	Initialized	Initialized	Initialized	Initialized
	RS	Initialized	Initialized	Initialized	Initialized	Initialized
	ST	Initialized	Initialized	Initialized	Initialized	Retained
RIICnMR1	MTWP	Initialized	Initialized	Retained	Retained	Retained
	CKS[2:0]	Initialized	Initialized	Retained	Retained	Retained
	BCWP	Initialized	Initialized	Retained	Retained	Retained
	BC[2:0]	Initialized	Initialized	Initialized	Initialized	Retained
RIICnMR2		Initialized	Initialized	Retained	Retained	Retained
RIICnMR3	WAIT	Initialized	Initialized	Retained	Retained	Retained
	RDRFS	Initialized	Initialized	Retained	Retained	Retained
	ACKWP	Initialized	Initialized	Retained	Retained	Retained
	ACKBT	Initialized	Initialized	Retained	Retained	Initialized
	ACKBR	Initialized	Initialized	Retained	Retained	Retained
	NF[1:0]	Initialized	Initialized	Retained	Retained	Retained
RIICnFER		Initialized	Initialized	Retained	Retained	Retained
RIICnSER		Initialized	Initialized	Retained	Retained	Retained
RIICnIER		Initialized	Initialized	Retained	Retained	Retained
RIICnSR1	DID	Initialized	Initialized	Initialized	Retained	Initialized
	GCA	Initialized	Initialized	Initialized	Retained	Initialized
	AAS2	Initialized	Initialized	Initialized	Retained	Initialized
	AAS1	Initialized	Initialized	Initialized	Retained	Initialized
	AAS0	Initialized	Initialized	Initialized	Retained	Initialized
RIICnSR2	TDRE	Initialized	Initialized	Initialized	Retained	Initialized
	TEND	Initialized	Initialized	Initialized	Retained	Initialized
	RDRF	Initialized	Initialized	Initialized	Retained	Retained
	NACKF	Initialized	Initialized	Initialized	Retained	Retained
	STOP	Initialized	Initialized	Initialized	Retained	Operation
	START	Initialized	Initialized	Initialized	Operation	Initialized
	AL	Initialized	Initialized	Initialized	Retained	Retained
	TMOF	Initialized	Initialized	Initialized	Retained	Retained

Table 21.26 RIIC Reset Functions (2/2)

UM	SYSRES/ wake-up from DEEPSTOP mode	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start/Restart Condition Detection	Stop Condition Detection
RIICnSAR0, 1, 2	Initialized	Initialized	Retained	Retained	Retained
RIICnBRH, RIICnBRL	Initialized	Initialized	Retained	Retained	Retained
RIICnDRT	Initialized	Initialized	Retained	Retained	Retained
RIICnDRR	Initialized	Initialized	Retained	Retained	Retained
RIICnDRS	Initialized	Initialized	Initialized	Retained	Retained

Note 1. When an internal reset is applied while the bus is free after detection of a stop condition, the setting of the BBSY flag is 0 while the bus is free following de-assertion of the internal reset signal.
When an internal reset is applied while the bus is not free, the BBSY flag is not cleared.

Section 22 CAN Interface (RSCAN)

This section contains a generic description of the CAN Interface (RSCAN).

The first part of this section describes all RH850/D1L/D1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RSCAN.

22.1 Features of RH850/D1L/D1M RSCAN

22.1.1 Number of Units and Channels

This microcontroller has the following number of RSCAN units.

Table 22.1 Number of Units for D1L Devices

	D1L1		D1L2(H)	
Product Name	**R7F701401**	**R7F701421**	**R7F701402 R7F701403**	**R7F701422 R7F701423**
Number of Units	1	0	1	0
Name	RSCAN0	–	RSCAN0	–

Table 22.2 Number of Units for D1M Devices

	D1M1(H)	D1M1-V2		D1M1A		D1M2(H)	
Product Name	**R7F701404 R7F701405 R7F701406 R7F701407**	**R7F701442**	**R7F701462**	**R7F701441**	**R7F701461**	**R7F701408 R7F701410 R7F701411 R7F701412**	**R7F701428 R7F701430 R7F701431 R7F701432**
Number of Units	1	1	0	1	0	1	0
Name	RSCAN0	RSCAN0	–	RSCAN0	–	RSCAN0	–

The individual products have the CAN Interface Channel listed below.

Table 22.3 Unit Configurations and Channels for D1L Devices

	D1L1		D1L2(H)	
Channel Name	**R7F701401**	**R7F701421**	**R7F701402 R7F701403**	**R7F701422 R7F701423**
Unit Name: RSCAN0				
CAN0	√	–	√	–
CAN1	√	–	√	–
CAN2	√	–	√	–

Table 22.4 Unit Configurations and Channels for D1M Devices

	D1M1(H)	D1M1-V2		D1M1A		D1M2(H)	
Channel Name	R7F701404 R7F701405 R7F701406 R7F701407	R7F701442	R7F701462	R7F701441	R7F701461	R7F701408 R7F701410 R7F701411 R7F701412	R7F701428 R7F701430 R7F701431 R7F701432
Unit Name: RSCAN0							
CAN0	√	√	–	√	–	√	–
CAN1	√	√	–	√	–	√	–
CAN2	√	√	–	√	–	√	–

Table 22.5 Index

Index	Meaning
n	Throughout this section, the individual RSCAN units are generically indicated by the index “n” (n = 0); for example, RSCANnGCTR is the global control register of the RSCANn unit.
m	Throughout this section, the individual channels of RSCAN units are generically indicated by the index “m” (m = 0 to 2); for example, RSCAN0CmSTS is the channel m status register.
j	The individual registers associated with receive rule table are generically indicated by the index “j” (j = 0 to 15); for example, RSCAN0GAFLIDj is the receive rule ID register.
k	The individual transmit/receive FIFO buffers are generically indicated by the index “k” (k = 0 to 8); for example, RSCAN0CFCCk is the transmit/receive FIFO buffer configuration/control register.
x	The individual receive FIFO buffers in the RSCAN units are identified by the index “x” (x = 0 to 7): for example, RSCAN0RFSTSx is the receive FIFO buffer status register.
q	The individual receive buffers are generically indicated by the index “q” (q = 0 to 47); for example, RSCAN0RMIDq is the receive buffer ID register.
p	The individual transmit buffers are generically indicated by the index “p” (p = 0 to 47); for example, RSCAN0TMCp is the transmit buffer control register.
r	The individual RAM tests for CAN are generically indicated by the index “r” (r = 0 to 63); for example, RSCAN0RPGACCr is the RAM test page access register.
y	The registers not covered above are indicated by the index “y” (y = 0, 1); for example, RSCAN0RMNDy is a receive buffer new data register.

22.1.2 Register Base Address

RSCAN0 base addresses are listed in the following table.

RSCAN0 register addresses are given as offsets from the base addresses in general.

Table 22.6 Register Base Address

Base Address Name	Base Address
<RSCAN0_base>	$FFD0\ 0000_H$

22.1.3 Clock Supply

The RSCAN0 clock supply is shown in the following table.

Table 22.7 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RSCAN0	clk_xincan	Clock Controller C_ISO_RSCANXIN
	clkc	Clock Controller C_ISO_RSCAN
	PCLK	Clock Controller RSCANPCLK

The operating frequency of the RSCAN0 depends on the transfer rate and the number of channels in use. **Table 22.8** shows the range of the frequency.

Table 22.8 Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/D1L/D1M

Condition		Range of Operating Frequency	
Transfer Rate	No. of Channels in Use	pclk	clkc/clk_xincan*[1]
1 Mbps	3ch	pclk ≥ 32MHz	8 MHz ≤ clk_xincan ≤ pclk/2
	2ch	pclk ≥ 26MHz	
	1ch	pclk ≥ 18MHz	
500 kbps	3ch	pclk ≥ 16MHz	4 MHz ≤ clk_xincan ≤ pclk/2
	2ch	pclk ≥ 13MHz	
	1ch	pclk ≥ 8MHz	
125 kbps	3ch	pclk ≥ 8MHz	4 MHz ≤ clk_xincan ≤ pclk/2
	2ch		
	1ch		

Note 1. Setting the DCS bit in RSCAN0GCFG enables to select either clk_xincan or clkc. Set clocks less than or equal to pclk/2.

22.1.4 Interrupt Request

RSCAN0 interrupt requests are listed in the following table.

Table 22.9 Interrupt Requests

Unit Interrupt Signal	Outline	Connected to
RSCAN0		
INTRCANGERR	CAN global error interrupt	Interrupt Controller INTRCANGERR
INTRCANGRECC	CAN receive FIFO interrupt	Interrupt Controller INTRCANGRECC
RSCAN0		
INTRCANmERR (m = 0)	CAN0 error interrupt	Interrupt Controller INTRCAN0ERR
INTRCANmREC (m = 0)	CAN0 transmit/receive FIFO receive completion interrupt	Interrupt Controller INTRCAN0REC
INTRCANmTRX (m = 0)	CAN0 transmit interrupt	Interrupt Controller INTRCAN0TRX
RSCAN0		
INTRCANmERR (m = 1)	CAN1 error interrupt	Interrupt Controller INTRCAN1ERR
INTRCANmREC (m = 1)	CAN1 transmit/receive FIFO receive completion interrupt	Interrupt Controller INTRCAN1REC
INTRCANmTRX (m = 1)	CAN1 transmit interrupt	Interrupt Controller INTRCAN1TRX
RSCAN0		
INTRCANmERR (m = 2)	CAN2 error interrupt	Interrupt Controller INTRCAN2ERR
INTRCANmREC (m = 2)	CAN2 transmit/receive FIFO receive completion interrupt	Interrupt Controller INTRCAN2REC
INTRCANmTRX (m = 2)	CAN2 transmit interrupt	Interrupt Controller INTRCAN2TRX

22.1.5 Reset Sources

RSCAN0 reset sources are listed in the following table. RSCAN0 is initialized by these reset sources.

Table 22.10 Reset Sources

Unit Name	Reset Source
RSCAN0	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

22.1.6 External Input/Output Signals

External input/output signals of RSCAN0 are listed below.

Table 22.11 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
CAN0		
CANmRX (m = 0)	CAN0 receive data input	CAN0RX
CANmTX (m = 0)	CAN0 transmit data output	CAN0TX
CAN1		
CANmRX (m = 1)	CAN1 receive data input	CAN1RX
CANmTX (m = 1)	CAN1 transmit data output	CAN1TX
CAN2		
CANmRX (m = 2)	CAN2 receive data input	CAN2RX
CANmTX (m = 2)	CAN2 transmit data output	CAN2TX

22.2 Overview

22.2.1 Functional Overview

The RH850/D1L/D1M incorporates one unit of the CAN interface (RSCAN) which consists of three channels (CAN0 to CAN2) of the CAN controller conforming to the ISO11898-1 specifications. **Table 22.12** shows the RSCAN module specifications. **Figure 22.1** shows the RSCAN module block diagram.

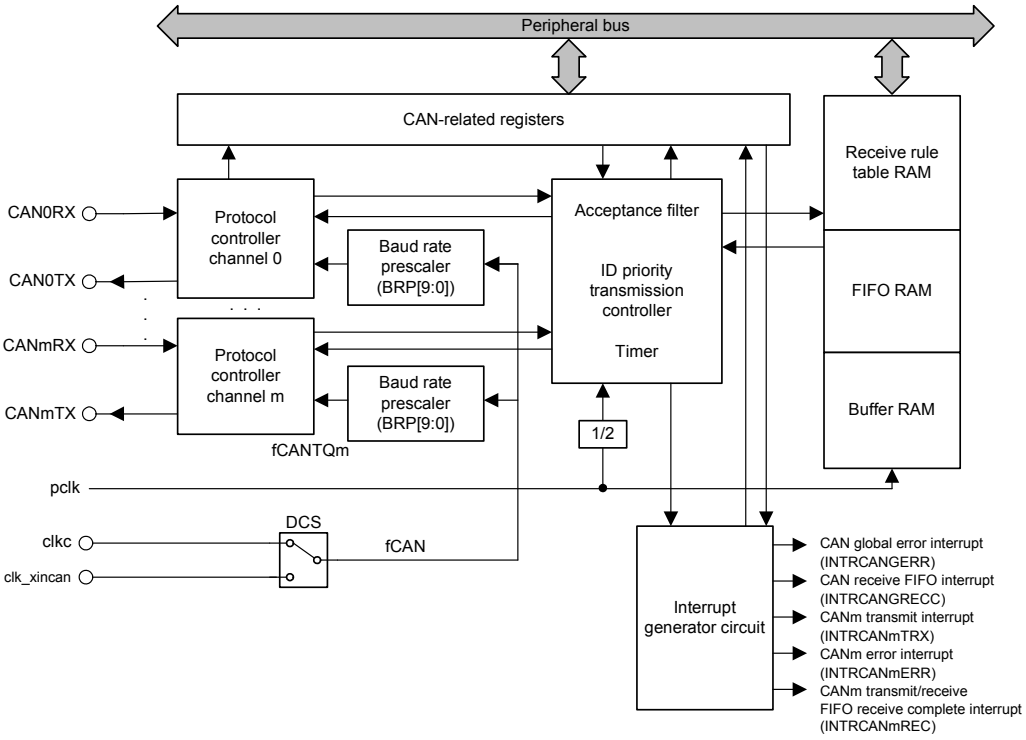
Table 22.12 RSCAN Module Specifications (1/2)

Item	Specification
Number of channels	3
Protocol	ISO11898-1 compliant
Communication speed	• Maximum 1 Mbps $\text{Commnucation speed (CANm bit time clock)} = \frac{1}{\text{CANm bit time}}$ CANm bit time = CANmTq × Tq count per bit $\text{CANmTq} = \frac{\text{(BRP[9:0] bits in the RSCAN0CmCFG register + 1)}}{\text{fCAN}}$ m = 0 to 2 Tq: Time quantum fCAN: Frequency of CAN clock (selected by the DCS bit in the RSCAN0GCFG register)
Buffer	240 buffers in total • Individual buffers: 48 buffers (16 buffers × 3 channels) Transmit buffer: 16 buffers per channel Transmit queue: Single queue per channel (shared with the transmit buffer; up to 16 buffers allocatable) • Shared buffers: 192 buffers for all channels Receive buffer: 48 buffers Receive FIFO buffer: 8 FIFO buffers (up to 128 buffers allocatable to each) Transmit/receive FIFO buffer: 3 FIFO buffers per channel (up to 128 buffers allocatable to each) • ECC included The 7-bit ECC value is calculated for each 32-bit transfer.
Reception function	• Receives data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be received. • Sets interrupt enable/disable for each FIFO. • Mirror function (reception of messages transmitted from the own CAN node) • Timestamp function (to record message reception time as a 16-bit timer value)
Reception filter function	• Selects receive messages according to 192 receive rules. • Sets the number of receive rules (0 to 127) for each channel. • Acceptance filter processing: Sets ID and mask for each receive rule. • DLC filter processing: Enables DLC filter check for each acceptance rule.
Receive message transfer function	• Routing function Transfers receive messages to arbitrary destinations (can be transferred to up to 8 buffers) Transfer destination: Receive buffer, receive FIFO buffer, and/or transmit/receive FIFO buffer • Label addition function Stores label information together with a message in a receive buffer and FIFO buffer.
Transmission function	• Transmits data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be transmitted. • Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer. • Selects ID priority transmission or transmit buffer number priority transmission. • Transmit request can be aborted (possible to confirm with a flag) • One-shot transmission function

Table 22.12 RSCAN Module Specifications (2/2)

Item	Specification
Interval transmission function	Transmit messages at configurable intervals (transmit mode or gateway mode of transmit/receive FIFO buffers)
Transmit queue function	Transmits all stored messages according to the ID priority.
Transmit history function	Stores the history information of transmission-completed messages
Gateway function	Transmits a received message automatically.
Bus off recovery mode selection	Selects the method for returning from bus off state. • ISO11898-1 compliant • Automatic entry to channel halt mode at bus-off entry • Automatic entry to channel halt mode at bus-off end • Transition to channel halt mode by program request • Transition to the error-active state by program request (forcible return from the bus off state)
Error status monitoring	• Monitors CAN protocol errors (stuff error, form error, ACK error, CRC error, bit error, ACK delimiter error, and bus dominant lock). • Detects error status transitions (error warning, error passive, bus off entry, and bus off recovery) • Reads the error counter. • Monitors DLC errors.
Interrupt source	11 sources • Global Interrupts (2 sources) Receive FIFO interrupt Global error interrupt • Channel interrupts (3 sources/channel) CANm transmit interrupt (m = 0 to 2) – CANm transmit complete interrupt – CANm transmit abort interrupt – CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode) – CANm transmit history interrupt – CANm transmit queue interrupt CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode) CANm error interrupt
CAN stop mode	Reduces power consumption by stopping clock supply to the RSCAN module.
CAN clock source	Selects the clkc or the clk_xincan. As for the range of operating frequency, see **Table 22.8**.
Test function	Test function for user evaluation • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback) • RAM test (read/write test) • Inter-channel communication test

22.2.2 Block Diagram



Note: m = 0 to 2
BRP[9:0]: Bits in the RSCAN0CmCFG register
DCS: Bits in the RSCAN0GCFG register
fCANTQm: CANmTq clock
fCAN: CAN clock

Figure 22.1 RSCAN Module Block Diagram

22.3 Registers

22.3.1 List of Registers

RSCAN registers are listed in the following table.

For details about <RSCAN0_base>, see Section 22.1.2, Register Base Address.

Table 22.13 Registers (1/19)

Module	Register	Symbol	Address
RSCAN0	Channel 0 configuration register	RSCAN0C0CFG	<RSCAN0_base> + 0000_H
RSCAN0	Channel 0 control register	RSCAN0C0CTR	<RSCAN0_base> + 0004_H
RSCAN0	Channel 0 status register	RSCAN0C0STS	<RSCAN0_base> + 0008_H
RSCAN0	Channel 0 error flag register	RSCAN0C0ERFL	<RSCAN0_base> + $000C_H$
RSCAN0	Channel 1 configuration register	RSCAN0C1CFG	<RSCAN0_base> + 0010_H
RSCAN0	Channel 1 control register	RSCAN0C1CTR	<RSCAN0_base> + 0014_H
RSCAN0	Channel 1 status register	RSCAN0C1STS	<RSCAN0_base> + 0018_H
RSCAN0	Channel 1 error flag register	RSCAN0C1ERFL	<RSCAN0_base> + $001C_H$
RSCAN0	Channel 2 configuration register	RSCAN0C2CFG	<RSCAN0_base> + 0020_H
RSCAN0	Channel 2 control register	RSCAN0C2CTR	<RSCAN0_base> + 0024_H
RSCAN0	Channel 2 status register	RSCAN0C2STS	<RSCAN0_base> + 0028_H
RSCAN0	Channel 2 error flag register	RSCAN0C2ERFL	<RSCAN0_base> + $002C_H$
RSCAN0	Global configuration register	RSCAN0GCFG	<RSCAN0_base> + 0084_H
RSCAN0	Global control register	RSCAN0GCTR	<RSCAN0_base> + 0088_H
RSCAN0	Global status register	RSCAN0GSTS	<RSCAN0_base> + $008C_H$
RSCAN0	Global error flag register	RSCAN0GERFL	<RSCAN0_base> + 0090_H
RSCAN0	Global timestamp counter register	RSCAN0GTSC	<RSCAN0_base> + 0094_H
RSCAN0	Receive rule entry control register	RSCAN0GAFLECTR	<RSCAN0_base> + 0098_H
RSCAN0	Receive rule configuration register 0	RSCAN0GAFLCFG0	<RSCAN0_base> + $009C_H$
RSCAN0	Receive buffer number register	RSCAN0RMNB	<RSCAN0_base> + $00A4_H$
RSCAN0	Receive buffer new data register 0	RSCAN0RMND0	<RSCAN0_base> + $00A8_H$
RSCAN0	Receive buffer new data register 1	RSCAN0RMND1	<RSCAN0_base> + $00AC_H$
RSCAN0	Receive FIFO buffer configuration and control register 0	RSCAN0RFCC0	<RSCAN0_base> + $00B8_H$
RSCAN0	Receive FIFO buffer configuration and control register 1	RSCAN0RFCC1	<RSCAN0_base> + $00BC_H$
RSCAN0	Receive FIFO buffer configuration and control register 2	RSCAN0RFCC2	<RSCAN0_base> + $00C0_H$
RSCAN0	Receive FIFO buffer configuration and control register 3	RSCAN0RFCC3	<RSCAN0_base> + $00C4_H$
RSCAN0	Receive FIFO buffer configuration and control register 4	RSCAN0RFCC4	<RSCAN0_base> + $00C8_H$
RSCAN0	Receive FIFO buffer configuration and control register 5	RSCAN0RFCC5	<RSCAN0_base> + $00CC_H$
RSCAN0	Receive FIFO buffer configuration and control register 6	RSCAN0RFCC6	<RSCAN0_base> + $00D0_H$
RSCAN0	Receive FIFO buffer configuration and control register 7	RSCAN0RFCC7	<RSCAN0_base> + $00D4_H$
RSCAN0	Receive FIFO buffer status register 0	RSCAN0RFSTS0	<RSCAN0_base> + $00D8_H$
RSCAN0	Receive FIFO buffer status register 1	RSCAN0RFSTS1	<RSCAN0_base> + $00DC_H$
RSCAN0	Receive FIFO buffer status register 2	RSCAN0RFSTS2	<RSCAN0_base> + $00E0_H$
RSCAN0	Receive FIFO buffer status register 3	RSCAN0RFSTS3	<RSCAN0_base> + $00E4_H$
RSCAN0	Receive FIFO buffer status register 4	RSCAN0RFSTS4	<RSCAN0_base> + $00E8_H$
RSCAN0	Receive FIFO buffer status register 5	RSCAN0RFSTS5	<RSCAN0_base> + $00EC_H$
RSCAN0	Receive FIFO buffer status register 6	RSCAN0RFSTS6	<RSCAN0_base> + $00F0_H$
RSCAN0	Receive FIFO buffer status register 7	RSCAN0RFSTS7	<RSCAN0_base> + $00F4_H$

Table 22.13 Registers (2/19)

Module	Register	Symbol	Address
RSCAN0	Receive FIFO buffer pointer control register 0	RSCAN0RFPCTR0	<RSCAN0_base> + $00F8_H$
RSCAN0	Receive FIFO buffer pointer control register 1	RSCAN0RFPCTR1	<RSCAN0_base> + $00FC_H$
RSCAN0	Receive FIFO buffer pointer control register 2	RSCAN0RFPCTR2	<RSCAN0_base> + 0100_H
RSCAN0	Receive FIFO buffer pointer control register 3	RSCAN0RFPCTR3	<RSCAN0_base> + 0104_H
RSCAN0	Receive FIFO buffer pointer control register 4	RSCAN0RFPCTR4	<RSCAN0_base> + 0108_H
RSCAN0	Receive FIFO buffer pointer control register 5	RSCAN0RFPCTR5	<RSCAN0_base> + $010C_H$
RSCAN0	Receive FIFO buffer pointer control register 6	RSCAN0RFPCTR6	<RSCAN0_base> + 0110_H
RSCAN0	Receive FIFO buffer pointer control register 7	RSCAN0RFPCTR7	<RSCAN0_base> + 0114_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 0	RSCAN0CFCC0	<RSCAN0_base> + 0118_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 1	RSCAN0CFCC1	<RSCAN0_base> + $011C_H$
RSCAN0	Transmit/receive FIFO buffer configuration and control register 2	RSCAN0CFCC2	<RSCAN0_base> + 0120_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 3	RSCAN0CFCC3	<RSCAN0_base> + 0124_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 4	RSCAN0CFCC4	<RSCAN0_base> + 0128_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 5	RSCAN0CFCC5	<RSCAN0_base> + $012C_H$
RSCAN0	Transmit/receive FIFO buffer configuration and control register 6	RSCAN0CFCC6	<RSCAN0_base> + 0130_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 7	RSCAN0CFCC7	<RSCAN0_base> + 0134_H
RSCAN0	Transmit/receive FIFO buffer configuration and control register 8	RSCAN0CFCC8	<RSCAN0_base> + 0138_H
RSCAN0	Transmit/receive FIFO buffer status register 0	RSCAN0CFSTS0	<RSCAN0_base> + 0178_H
RSCAN0	Transmit/receive FIFO buffer status register 1	RSCAN0CFSTS1	<RSCAN0_base> + $017C_H$
RSCAN0	Transmit/receive FIFO buffer status register 2	RSCAN0CFSTS2	<RSCAN0_base> + 0180_H
RSCAN0	Transmit/receive FIFO buffer status register 3	RSCAN0CFSTS3	<RSCAN0_base> + 0184_H
RSCAN0	Transmit/receive FIFO buffer status register 4	RSCAN0CFSTS4	<RSCAN0_base> + 0188_H
RSCAN0	Transmit/receive FIFO buffer status register 5	RSCAN0CFSTS5	<RSCAN0_base> + $018C_H$
RSCAN0	Transmit/receive FIFO buffer status register 6	RSCAN0CFSTS6	<RSCAN0_base> + 0190_H
RSCAN0	Transmit/receive FIFO buffer status register 7	RSCAN0CFSTS7	<RSCAN0_base> + 0194_H
RSCAN0	Transmit/receive FIFO buffer status register 8	RSCAN0CFSTS8	<RSCAN0_base> + 0198_H
RSCAN0	Transmit/receive FIFO buffer pointer control register 0	RSCAN0CFPCTR0	<RSCAN0_base> + $01D8_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 1	RSCAN0CFPCTR1	<RSCAN0_base> + $01DC_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 2	RSCAN0CFPCTR2	<RSCAN0_base> + $01E0_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 3	RSCAN0CFPCTR3	<RSCAN0_base> + $01E4_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 4	RSCAN0CFPCTR4	<RSCAN0_base> + $01E8_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 5	RSCAN0CFPCTR5	<RSCAN0_base> + $01EC_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 6	RSCAN0CFPCTR6	<RSCAN0_base> + $01F0_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 7	RSCAN0CFPCTR7	<RSCAN0_base> + $01F4_H$
RSCAN0	Transmit/receive FIFO buffer pointer control register 8	RSCAN0CFPCTR8	<RSCAN0_base> + $01F8_H$
RSCAN0	FIFO empty status register	RSCAN0FESTS	<RSCAN0_base> + 0238_H
RSCAN0	FIFO full status register	RSCAN0FFSTS	<RSCAN0_base> + $023C_H$
RSCAN0	FIFO Msg lost status register	RSCAN0FMSTS	<RSCAN0_base> + 0240_H
RSCAN0	Receive FIFO buffer interrupt flag status register	RSCAN0RFISTS	<RSCAN0_base> + 0244_H
RSCAN0	Transmit/receive FIFO buffer RX interrupt flag status register	RSCAN0CFRISTS	<RSCAN0_base> + 0248_H
RSCAN0	Transmit/receive FIFO buffer TX interrupt flag status register	RSCAN0CFTISTS	<RSCAN0_base> + $024C_H$
RSCAN0	Transmit buffer control register 0	RSCAN0TMC0	<RSCAN0_base> + 0250_H
RSCAN0	Transmit buffer control register 1	RSCAN0TMC1	<RSCAN0_base> + 0251_H
RSCAN0	Transmit buffer control register 2	RSCAN0TMC2	<RSCAN0_base> + 0252_H

Table 22.13 Registers (3/19)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer control register 3	RSCAN0TMC3	<RSCAN0_base> + 0253_H
RSCAN0	Transmit buffer control register 4	RSCAN0TMC4	<RSCAN0_base> + 0254_H
RSCAN0	Transmit buffer control register 5	RSCAN0TMC5	<RSCAN0_base> + 0255_H
RSCAN0	Transmit buffer control register 6	RSCAN0TMC6	<RSCAN0_base> + 0256_H
RSCAN0	Transmit buffer control register 7	RSCAN0TMC7	<RSCAN0_base> + 0257_H
RSCAN0	Transmit buffer control register 8	RSCAN0TMC8	<RSCAN0_base> + 0258_H
RSCAN0	Transmit buffer control register 9	RSCAN0TMC9	<RSCAN0_base> + 0259_H
RSCAN0	Transmit buffer control register 10	RSCAN0TMC10	<RSCAN0_base> + $025A_H$
RSCAN0	Transmit buffer control register 11	RSCAN0TMC11	<RSCAN0_base> + $025B_H$
RSCAN0	Transmit buffer control register 12	RSCAN0TMC12	<RSCAN0_base> + $025C_H$
RSCAN0	Transmit buffer control register 13	RSCAN0TMC13	<RSCAN0_base> + $025D_H$
RSCAN0	Transmit buffer control register 14	RSCAN0TMC14	<RSCAN0_base> + $025E_H$
RSCAN0	Transmit buffer control register 15	RSCAN0TMC15	<RSCAN0_base> + $025F_H$
RSCAN0	Transmit buffer control register 16	RSCAN0TMC16	<RSCAN0_base> + 0260_H
RSCAN0	Transmit buffer control register 17	RSCAN0TMC17	<RSCAN0_base> + 0261_H
RSCAN0	Transmit buffer control register 18	RSCAN0TMC18	<RSCAN0_base> + 0262_H
RSCAN0	Transmit buffer control register 19	RSCAN0TMC19	<RSCAN0_base> + 0263_H
RSCAN0	Transmit buffer control register 20	RSCAN0TMC20	<RSCAN0_base> + 0264_H
RSCAN0	Transmit buffer control register 21	RSCAN0TMC21	<RSCAN0_base> + 0265_H
RSCAN0	Transmit buffer control register 22	RSCAN0TMC22	<RSCAN0_base> + 0266_H
RSCAN0	Transmit buffer control register 23	RSCAN0TMC23	<RSCAN0_base> + 0267_H
RSCAN0	Transmit buffer control register 24	RSCAN0TMC24	<RSCAN0_base> + 0268_H
RSCAN0	Transmit buffer control register 25	RSCAN0TMC25	<RSCAN0_base> + 0269_H
RSCAN0	Transmit buffer control register 26	RSCAN0TMC26	<RSCAN0_base> + $026A_H$
RSCAN0	Transmit buffer control register 27	RSCAN0TMC27	<RSCAN0_base> + $026B_H$
RSCAN0	Transmit buffer control register 28	RSCAN0TMC28	<RSCAN0_base> + $026C_H$
RSCAN0	Transmit buffer control register 29	RSCAN0TMC29	<RSCAN0_base> + $026D_H$
RSCAN0	Transmit buffer control register 30	RSCAN0TMC30	<RSCAN0_base> + $026E_H$
RSCAN0	Transmit buffer control register 31	RSCAN0TMC31	<RSCAN0_base> + $026F_H$
RSCAN0	Transmit buffer control register 32	RSCAN0TMC32	<RSCAN0_base> + 0270_H
RSCAN0	Transmit buffer control register 33	RSCAN0TMC33	<RSCAN0_base> + 0271_H
RSCAN0	Transmit buffer control register 34	RSCAN0TMC34	<RSCAN0_base> + 0272_H
RSCAN0	Transmit buffer control register 35	RSCAN0TMC35	<RSCAN0_base> + 0273_H
RSCAN0	Transmit buffer control register 36	RSCAN0TMC36	<RSCAN0_base> + 0274_H
RSCAN0	Transmit buffer control register 37	RSCAN0TMC37	<RSCAN0_base> + 0275_H
RSCAN0	Transmit buffer control register 38	RSCAN0TMC38	<RSCAN0_base> + 0276_H
RSCAN0	Transmit buffer control register 39	RSCAN0TMC39	<RSCAN0_base> + 0277_H
RSCAN0	Transmit buffer control register 40	RSCAN0TMC40	<RSCAN0_base> + 0278_H
RSCAN0	Transmit buffer control register 41	RSCAN0TMC41	<RSCAN0_base> + 0279_H
RSCAN0	Transmit buffer control register 42	RSCAN0TMC42	<RSCAN0_base> + $027A_H$
RSCAN0	Transmit buffer control register 43	RSCAN0TMC43	<RSCAN0_base> + $027B_H$
RSCAN0	Transmit buffer control register 44	RSCAN0TMC44	<RSCAN0_base> + $027C_H$
RSCAN0	Transmit buffer control register 45	RSCAN0TMC45	<RSCAN0_base> + $027D_H$
RSCAN0	Transmit buffer control register 46	RSCAN0TMC46	<RSCAN0_base> + $027E_H$

Table 22.13 Registers (4/19)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer control register 47	RSCAN0TMC47	<RSCAN0_base> + $027F_H$
RSCAN0	Transmit buffer status register 0	RSCAN0TMSTS0	<RSCAN0_base> + $02D0_H$
RSCAN0	Transmit buffer status register 1	RSCAN0TMSTS1	<RSCAN0_base> + $02D1_H$
RSCAN0	Transmit buffer status register 2	RSCAN0TMSTS2	<RSCAN0_base> + $02D2_H$
RSCAN0	Transmit buffer status register 3	RSCAN0TMSTS3	<RSCAN0_base> + $02D3_H$
RSCAN0	Transmit buffer status register 4	RSCAN0TMSTS4	<RSCAN0_base> + $02D4_H$
RSCAN0	Transmit buffer status register 5	RSCAN0TMSTS5	<RSCAN0_base> + $02D5_H$
RSCAN0	Transmit buffer status register 6	RSCAN0TMSTS6	<RSCAN0_base> + $02D6_H$
RSCAN0	Transmit buffer status register 7	RSCAN0TMSTS7	<RSCAN0_base> + $02D7_H$
RSCAN0	Transmit buffer status register 8	RSCAN0TMSTS8	<RSCAN0_base> + $02D8_H$
RSCAN0	Transmit buffer status register 9	RSCAN0TMSTS9	<RSCAN0_base> + $02D9_H$
RSCAN0	Transmit buffer status register 10	RSCAN0TMSTS10	<RSCAN0_base> + $02DA_H$
RSCAN0	Transmit buffer status register 11	RSCAN0TMSTS11	<RSCAN0_base> + $02DB_H$
RSCAN0	Transmit buffer status register 12	RSCAN0TMSTS12	<RSCAN0_base> + $02DC_H$
RSCAN0	Transmit buffer status register 13	RSCAN0TMSTS13	<RSCAN0_base> + $02DD_H$
RSCAN0	Transmit buffer status register 14	RSCAN0TMSTS14	<RSCAN0_base> + $02DE_H$
RSCAN0	Transmit buffer status register 15	RSCAN0TMSTS15	<RSCAN0_base> + $02DF_H$
RSCAN0	Transmit buffer status register 16	RSCAN0TMSTS16	<RSCAN0_base> + $02E0_H$
RSCAN0	Transmit buffer status register 17	RSCAN0TMSTS17	<RSCAN0_base> + $02E1_H$
RSCAN0	Transmit buffer status register 18	RSCAN0TMSTS18	<RSCAN0_base> + $02E2_H$
RSCAN0	Transmit buffer status register 19	RSCAN0TMSTS19	<RSCAN0_base> + $02E3_H$
RSCAN0	Transmit buffer status register 20	RSCAN0TMSTS20	<RSCAN0_base> + $02E4_H$
RSCAN0	Transmit buffer status register 21	RSCAN0TMSTS21	<RSCAN0_base> + $02E5_H$
RSCAN0	Transmit buffer status register 22	RSCAN0TMSTS22	<RSCAN0_base> + $02E6_H$
RSCAN0	Transmit buffer status register 23	RSCAN0TMSTS23	<RSCAN0_base> + $02E7_H$
RSCAN0	Transmit buffer status register 24	RSCAN0TMSTS24	<RSCAN0_base> + $02E8_H$
RSCAN0	Transmit buffer status register 25	RSCAN0TMSTS25	<RSCAN0_base> + $02E9_H$
RSCAN0	Transmit buffer status register 26	RSCAN0TMSTS26	<RSCAN0_base> + $02EA_H$
RSCAN0	Transmit buffer status register 27	RSCAN0TMSTS27	<RSCAN0_base> + $02EB_H$
RSCAN0	Transmit buffer status register 28	RSCAN0TMSTS28	<RSCAN0_base> + $02EC_H$
RSCAN0	Transmit buffer status register 29	RSCAN0TMSTS29	<RSCAN0_base> + $02ED_H$
RSCAN0	Transmit buffer status register 30	RSCAN0TMSTS30	<RSCAN0_base> + $02EE_H$
RSCAN0	Transmit buffer status register 31	RSCAN0TMSTS31	<RSCAN0_base> + $02EF_H$
RSCAN0	Transmit buffer status register 32	RSCAN0TMSTS32	<RSCAN0_base> + $02F0_H$
RSCAN0	Transmit buffer status register 33	RSCAN0TMSTS33	<RSCAN0_base> + $02F1_H$
RSCAN0	Transmit buffer status register 34	RSCAN0TMSTS34	<RSCAN0_base> + $02F2_H$
RSCAN0	Transmit buffer status register 35	RSCAN0TMSTS35	<RSCAN0_base> + $02F3_H$
RSCAN0	Transmit buffer status register 36	RSCAN0TMSTS36	<RSCAN0_base> + $02F4_H$
RSCAN0	Transmit buffer status register 37	RSCAN0TMSTS37	<RSCAN0_base> + $02F5_H$
RSCAN0	Transmit buffer status register 38	RSCAN0TMSTS38	<RSCAN0_base> + $02F6_H$
RSCAN0	Transmit buffer status register 39	RSCAN0TMSTS39	<RSCAN0_base> + $02F7_H$
RSCAN0	Transmit buffer status register 40	RSCAN0TMSTS40	<RSCAN0_base> + $02F8_H$
RSCAN0	Transmit buffer status register 41	RSCAN0TMSTS41	<RSCAN0_base> + $02F9_H$
RSCAN0	Transmit buffer status register 42	RSCAN0TMSTS42	<RSCAN0_base> + $02FA_H$

Table 22.13 Registers (5/19)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer status register 43	RSCAN0TMSTS43	<RSCAN0_base> + $02FB_H$
RSCAN0	Transmit buffer status register 44	RSCAN0TMSTS44	<RSCAN0_base> + $02FC_H$
RSCAN0	Transmit buffer status register 45	RSCAN0TMSTS45	<RSCAN0_base> + $02FD_H$
RSCAN0	Transmit buffer status register 46	RSCAN0TMSTS46	<RSCAN0_base> + $02FE_H$
RSCAN0	Transmit buffer status register 47	RSCAN0TMSTS47	<RSCAN0_base> + $02FF_H$
RSCAN0	Transmit buffer transmit request status register 0	RSCAN0TMTRSTS0	<RSCAN0_base> + 0350_H
RSCAN0	Transmit buffer transmit request status register 1	RSCAN0TMTRSTS1	<RSCAN0_base> + 0354_H
RSCAN0	Transmit buffer transmit abort request status register 0	RSCAN0TMTARSTS0	<RSCAN0_base> + 0360_H
RSCAN0	Transmit buffer transmit abort request status register 1	RSCAN0TMTARSTS1	<RSCAN0_base> + 0364_H
RSCAN0	Transmit buffer transmit complete status register 0	RSCAN0TMTCSTS0	<RSCAN0_base> + 0370_H
RSCAN0	Transmit buffer transmit complete status register 1	RSCAN0TMTCSTS1	<RSCAN0_base> + 0374_H
RSCAN0	Transmit buffer transmit abort status register 0	RSCAN0TMTASTS0	<RSCAN0_base> + 0380_H
RSCAN0	Transmit buffer transmit abort status register 1	RSCAN0TMTASTS1	<RSCAN0_base> + 0384_H
RSCAN0	Transmit buffer interrupt enable configuration register 0	RSCAN0TMIEC0	<RSCAN0_base> + 0390_H
RSCAN0	Transmit buffer interrupt enable configuration register 1	RSCAN0TMIEC1	<RSCAN0_base> + 0394_H
RSCAN0	Transmit queue configuration and control register 0	RSCAN0TXQCC0	<RSCAN0_base> + $03A0_H$
RSCAN0	Transmit queue configuration and control register 1	RSCAN0TXQCC1	<RSCAN0_base> + $03A4_H$
RSCAN0	Transmit queue configuration and control register 2	RSCAN0TXQCC2	<RSCAN0_base> + $03A8_H$
RSCAN0	Transmit queue status register 0	RSCAN0TXQSTS0	<RSCAN0_base> + $03C0_H$
RSCAN0	Transmit queue status register 1	RSCAN0TXQSTS1	<RSCAN0_base> + $03C4_H$
RSCAN0	Transmit queue status register 2	RSCAN0TXQSTS2	<RSCAN0_base> + $03C8_H$
RSCAN0	Transmit queue pointer control register 0	RSCAN0TXQPCTR0	<RSCAN0_base> + $03E0_H$
RSCAN0	Transmit queue pointer control register 1	RSCAN0TXQPCTR1	<RSCAN0_base> + $03E4_H$
RSCAN0	Transmit queue pointer control register 2	RSCAN0TXQPCTR2	<RSCAN0_base> + $03E8_H$
RSCAN0	Transmit history configuration and control register 0	RSCAN0THLCC0	<RSCAN0_base> + 0400_H
RSCAN0	Transmit history configuration and control register 1	RSCAN0THLCC1	<RSCAN0_base> + 0404_H
RSCAN0	Transmit history configuration and control register 2	RSCAN0THLCC2	<RSCAN0_base> + 0408_H
RSCAN0	Transmit history status register 0	RSCAN0THLSTS0	<RSCAN0_base> + 0420_H
RSCAN0	Transmit history status register 1	RSCAN0THLSTS1	<RSCAN0_base> + 0424_H
RSCAN0	Transmit history status register 2	RSCAN0THLSTS2	<RSCAN0_base> + 0428_H
RSCAN0	Transmit history pointer control register 0	RSCAN0THLPCTR0	<RSCAN0_base> + 0440_H
RSCAN0	Transmit history pointer control register 1	RSCAN0THLPCTR1	<RSCAN0_base> + 0444_H
RSCAN0	Transmit history pointer control register 2	RSCAN0THLPCTR2	<RSCAN0_base> + 0448_H
RSCAN0	Global TX interrupt status register 0	RSCAN0GTINTSTS0	<RSCAN0_base> + 0460_H
RSCAN0	Global test configuration register	RSCAN0GTSTCFG	<RSCAN0_base> + 0468_H
RSCAN0	Global test control register	RSCAN0GTSTCTR	<RSCAN0_base> + $046C_H$
RSCAN0	Global lock key register	RSCAN0GLOCKK	<RSCAN0_base> + $047C_H$
RSCAN0	Receive rule ID register 0	RSCAN0GAFLID0	<RSCAN0_base> + 0500_H
RSCAN0	Receive rule mask register 0	RSCAN0GAFLM0	<RSCAN0_base> + 0504_H
RSCAN0	Receive rule pointer 0 register 0	RSCAN0GAFLP00	<RSCAN0_base> + 0508_H
RSCAN0	Receive rule pointer 1 register 0	RSCAN0GAFLP10	<RSCAN0_base> + $050C_H$
RSCAN0	Receive rule ID register 1	RSCAN0GAFLID1	<RSCAN0_base> + 0510_H
RSCAN0	Receive rule mask register 1	RSCAN0GAFLM1	<RSCAN0_base> + 0514_H
RSCAN0	Receive rule pointer 0 register 1	RSCAN0GAFLP01	<RSCAN0_base> + 0518_H

Table 22.13 Registers (6/19)

Module	Register	Symbol	Address
RSCAN0	Receive rule pointer 1 register 1	RSCAN0GAFLP11	<RSCAN0_base> + 051C$_H$
RSCAN0	Receive rule ID register 2	RSCAN0GAFLID2	<RSCAN0_base> + 0520$_H$
RSCAN0	Receive rule mask register 2	RSCAN0GAFLM2	<RSCAN0_base> + 0524$_H$
RSCAN0	Receive rule pointer 0 register 2	RSCAN0GAFLP02	<RSCAN0_base> + 0528$_H$
RSCAN0	Receive rule pointer 1 register 2	RSCAN0GAFLP12	<RSCAN0_base> + 052C$_H$
RSCAN0	Receive rule ID register 3	RSCAN0GAFLID3	<RSCAN0_base> + 0530$_H$
RSCAN0	Receive rule mask register 3	RSCAN0GAFLM3	<RSCAN0_base> + 0534$_H$
RSCAN0	Receive rule pointer 0 register 3	RSCAN0GAFLP03	<RSCAN0_base> + 0538$_H$
RSCAN0	Receive rule pointer 1 register 3	RSCAN0GAFLP13	<RSCAN0_base> + 053C$_H$
RSCAN0	Receive rule ID register 4	RSCAN0GAFLID4	<RSCAN0_base> + 0540$_H$
RSCAN0	Receive rule mask register 4	RSCAN0GAFLM4	<RSCAN0_base> + 0544$_H$
RSCAN0	Receive rule pointer 0 register 4	RSCAN0GAFLP04	<RSCAN0_base> + 0548$_H$
RSCAN0	Receive rule pointer 1 register 4	RSCAN0GAFLP14	<RSCAN0_base> + 054C$_H$
RSCAN0	Receive rule ID register 5	RSCAN0GAFLID5	<RSCAN0_base> + 0550$_H$
RSCAN0	Receive rule mask register 5	RSCAN0GAFLM5	<RSCAN0_base> + 0554$_H$
RSCAN0	Receive rule pointer 0 register 5	RSCAN0GAFLP05	<RSCAN0_base> + 0558$_H$
RSCAN0	Receive rule pointer 1 register 5	RSCAN0GAFLP15	<RSCAN0_base> + 055C$_H$
RSCAN0	Receive rule ID register 6	RSCAN0GAFLID6	<RSCAN0_base> + 0560$_H$
RSCAN0	Receive rule mask register 6	RSCAN0GAFLM6	<RSCAN0_base> + 0564$_H$
RSCAN0	Receive rule pointer 0 register 6	RSCAN0GAFLP06	<RSCAN0_base> + 0568$_H$
RSCAN0	Receive rule pointer 1 register 6	RSCAN0GAFLP16	<RSCAN0_base> + 056C$_H$
RSCAN0	Receive rule ID register 7	RSCAN0GAFLID7	<RSCAN0_base> + 0570$_H$
RSCAN0	Receive rule mask register 7	RSCAN0GAFLM7	<RSCAN0_base> + 0574$_H$
RSCAN0	Receive rule pointer 0 register 7	RSCAN0GAFLP07	<RSCAN0_base> + 0578$_H$
RSCAN0	Receive rule pointer 1 register 7	RSCAN0GAFLP17	<RSCAN0_base> + 057C$_H$
RSCAN0	Receive rule ID register 8	RSCAN0GAFLID8	<RSCAN0_base> + 0580$_H$
RSCAN0	Receive rule mask register 8	RSCAN0GAFLM8	<RSCAN0_base> + 0584$_H$
RSCAN0	Receive rule pointer 0 register 8	RSCAN0GAFLP08	<RSCAN0_base> + 0588$_H$
RSCAN0	Receive rule pointer 1 register 8	RSCAN0GAFLP18	<RSCAN0_base> + 058C$_H$
RSCAN0	Receive rule ID register 9	RSCAN0GAFLID9	<RSCAN0_base> + 0590$_H$
RSCAN0	Receive rule mask register 9	RSCAN0GAFLM9	<RSCAN0_base> + 0594$_H$
RSCAN0	Receive rule pointer 0 register 9	RSCAN0GAFLP09	<RSCAN0_base> + 0598$_H$
RSCAN0	Receive rule pointer 1 register 9	RSCAN0GAFLP19	<RSCAN0_base> + 059C$_H$
RSCAN0	Receive rule ID register 10	RSCAN0GAFLID10	<RSCAN0_base> + 05A0$_H$
RSCAN0	Receive rule mask register 10	RSCAN0GAFLM10	<RSCAN0_base> + 05A4$_H$
RSCAN0	Receive rule pointer 0 register 10	RSCAN0GAFLP010	<RSCAN0_base> + 05A8$_H$
RSCAN0	Receive rule pointer 1 register 10	RSCAN0GAFLP110	<RSCAN0_base> + 05AC$_H$
RSCAN0	Receive rule ID register 11	RSCAN0GAFLID11	<RSCAN0_base> + 05B0$_H$
RSCAN0	Receive rule mask register 11	RSCAN0GAFLM11	<RSCAN0_base> + 05B4$_H$
RSCAN0	Receive rule pointer 0 register 11	RSCAN0GAFLP011	<RSCAN0_base> + 05B8$_H$
RSCAN0	Receive rule pointer 1 register 11	RSCAN0GAFLP111	<RSCAN0_base> + 05BC$_H$
RSCAN0	Receive rule ID register 12	RSCAN0GAFLID12	<RSCAN0_base> + 05C0$_H$
RSCAN0	Receive rule mask register 12	RSCAN0GAFLM12	<RSCAN0_base> + 05C4$_H$
RSCAN0	Receive rule pointer 0 register 12	RSCAN0GAFLP012	<RSCAN0_base> + 05C8$_H$

Table 22.13 Registers (7/19)

Module	Register	Symbol	Address
RSCAN0	Receive rule pointer 1 register 12	RSCAN0GAFLP112	<RSCAN0_base> + $05CC_H$
RSCAN0	Receive rule ID register 13	RSCAN0GAFLID13	<RSCAN0_base> + $05D0_H$
RSCAN0	Receive rule mask register 13	RSCAN0GAFLM13	<RSCAN0_base> + $05D4_H$
RSCAN0	Receive rule pointer 0 register 13	RSCAN0GAFLP013	<RSCAN0_base> + $05D8_H$
RSCAN0	Receive rule pointer 1 register 13	RSCAN0GAFLP113	<RSCAN0_base> + $05DC_H$
RSCAN0	Receive rule ID register 14	RSCAN0GAFLID14	<RSCAN0_base> + $05E0_H$
RSCAN0	Receive rule mask register 14	RSCAN0GAFLM14	<RSCAN0_base> + $05E4_H$
RSCAN0	Receive rule pointer 0 register 14	RSCAN0GAFLP014	<RSCAN0_base> + $05E8_H$
RSCAN0	Receive rule pointer 1 register 14	RSCAN0GAFLP114	<RSCAN0_base> + $05EC_H$
RSCAN0	Receive rule ID register 15	RSCAN0GAFLID15	<RSCAN0_base> + $05F0_H$
RSCAN0	Receive rule mask register 15	RSCAN0GAFLM15	<RSCAN0_base> + $05F4_H$
RSCAN0	Receive rule pointer 0 register 15	RSCAN0GAFLP015	<RSCAN0_base> + $05F8_H$
RSCAN0	Receive rule pointer 1 register 15	RSCAN0GAFLP115	<RSCAN0_base> + $05FC_H$
RSCAN0	Receive buffer ID register 0	RSCAN0RMID0	<RSCAN0_base> + 0600_H
RSCAN0	Receive buffer pointer register 0	RSCAN0RMPTR0	<RSCAN0_base> + 0604_H
RSCAN0	Receive buffer data field 0 register 0	RSCAN0RMDF00	<RSCAN0_base> + 0608_H
RSCAN0	Receive buffer data field 1 register 0	RSCAN0RMDF10	<RSCAN0_base> + $060C_H$
RSCAN0	Receive buffer ID register 1	RSCAN0RMID1	<RSCAN0_base> + 0610_H
RSCAN0	Receive buffer pointer register 1	RSCAN0RMPTR1	<RSCAN0_base> + 0614_H
RSCAN0	Receive buffer data field 0 register 1	RSCAN0RMDF01	<RSCAN0_base> + 0618_H
RSCAN0	Receive buffer data field 1 register 1	RSCAN0RMDF11	<RSCAN0_base> + $061C_H$
RSCAN0	Receive buffer ID register 2	RSCAN0RMID2	<RSCAN0_base> + 0620_H
RSCAN0	Receive buffer pointer register 2	RSCAN0RMPTR2	<RSCAN0_base> + 0624_H
RSCAN0	Receive buffer data field 0 register 2	RSCAN0RMDF02	<RSCAN0_base> + 0628_H
RSCAN0	Receive buffer data field 1 register 2	RSCAN0RMDF12	<RSCAN0_base> + $062C_H$
RSCAN0	Receive buffer ID register 3	RSCAN0RMID3	<RSCAN0_base> + 0630_H
RSCAN0	Receive buffer pointer register 3	RSCAN0RMPTR3	<RSCAN0_base> + 0634_H
RSCAN0	Receive buffer data field 0 register 3	RSCAN0RMDF03	<RSCAN0_base> + 0638_H
RSCAN0	Receive buffer data field 1 register 3	RSCAN0RMDF13	<RSCAN0_base> + $063C_H$
RSCAN0	Receive buffer ID register 4	RSCAN0RMID4	<RSCAN0_base> + 0640_H
RSCAN0	Receive buffer pointer register 4	RSCAN0RMPTR4	<RSCAN0_base> + 0644_H
RSCAN0	Receive buffer data field 0 register 4	RSCAN0RMDF04	<RSCAN0_base> + 0648_H
RSCAN0	Receive buffer data field 1 register 4	RSCAN0RMDF14	<RSCAN0_base> + $064C_H$
RSCAN0	Receive buffer ID register 5	RSCAN0RMID5	<RSCAN0_base> + 0650_H
RSCAN0	Receive buffer pointer register 5	RSCAN0RMPTR5	<RSCAN0_base> + 0654_H
RSCAN0	Receive buffer data field 0 register 5	RSCAN0RMDF05	<RSCAN0_base> + 0658_H
RSCAN0	Receive buffer data field 1 register 5	RSCAN0RMDF15	<RSCAN0_base> + $065C_H$
RSCAN0	Receive buffer ID register 6	RSCAN0RMID6	<RSCAN0_base> + 0660_H
RSCAN0	Receive buffer pointer register 6	RSCAN0RMPTR6	<RSCAN0_base> + 0664_H
RSCAN0	Receive buffer data field 0 register 6	RSCAN0RMDF06	<RSCAN0_base> + 0668_H
RSCAN0	Receive buffer data field 1 register 6	RSCAN0RMDF16	<RSCAN0_base> + $066C_H$
RSCAN0	Receive buffer ID register 7	RSCAN0RMID7	<RSCAN0_base> + 0670_H
RSCAN0	Receive buffer pointer register 7	RSCAN0RMPTR7	<RSCAN0_base> + 0674_H
RSCAN0	Receive buffer data field 0 register 7	RSCAN0RMDF07	<RSCAN0_base> + 0678_H

Table 22.13 Registers (8/19)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 1 register 7	RSCAN0RMDF17	<RSCAN0_base> + $067C_H$
RSCAN0	Receive buffer ID register 8	RSCAN0RMID8	<RSCAN0_base> + 0680_H
RSCAN0	Receive buffer pointer register 8	RSCAN0RMPTR8	<RSCAN0_base> + 0684_H
RSCAN0	Receive buffer data field 0 register 8	RSCAN0RMDF08	<RSCAN0_base> + 0688_H
RSCAN0	Receive buffer data field 1 register 8	RSCAN0RMDF18	<RSCAN0_base> + $068C_H$
RSCAN0	Receive buffer ID register 9	RSCAN0RMID9	<RSCAN0_base> + 0690_H
RSCAN0	Receive buffer pointer register 9	RSCAN0RMPTR9	<RSCAN0_base> + 0694_H
RSCAN0	Receive buffer data field 0 register 9	RSCAN0RMDF09	<RSCAN0_base> + 0698_H
RSCAN0	Receive buffer data field 1 register 9	RSCAN0RMDF19	<RSCAN0_base> + $069C_H$
RSCAN0	Receive buffer ID register 10	RSCAN0RMID10	<RSCAN0_base> + $06A0_H$
RSCAN0	Receive buffer pointer register 10	RSCAN0RMPTR10	<RSCAN0_base> + $06A4_H$
RSCAN0	Receive buffer data field 0 register 10	RSCAN0RMDF010	<RSCAN0_base> + $06A8_H$
RSCAN0	Receive buffer data field 1 register 10	RSCAN0RMDF110	<RSCAN0_base> + $06AC_H$
RSCAN0	Receive buffer ID register 11	RSCAN0RMID11	<RSCAN0_base> + $06B0_H$
RSCAN0	Receive buffer pointer register 11	RSCAN0RMPTR11	<RSCAN0_base> + $06B4_H$
RSCAN0	Receive buffer data field 0 register 11	RSCAN0RMDF011	<RSCAN0_base> + $06B8_H$
RSCAN0	Receive buffer data field 1 register 11	RSCAN0RMDF111	<RSCAN0_base> + $06BC_H$
RSCAN0	Receive buffer ID register 12	RSCAN0RMID12	<RSCAN0_base> + $06C0_H$
RSCAN0	Receive buffer pointer register 12	RSCAN0RMPTR12	<RSCAN0_base> + $06C4_H$
RSCAN0	Receive buffer data field 0 register 12	RSCAN0RMDF012	<RSCAN0_base> + $06C8_H$
RSCAN0	Receive buffer data field 1 register 12	RSCAN0RMDF112	<RSCAN0_base> + $06CC_H$
RSCAN0	Receive buffer ID register 13	RSCAN0RMID13	<RSCAN0_base> + $06D0_H$
RSCAN0	Receive buffer pointer register 13	RSCAN0RMPTR13	<RSCAN0_base> + $06D4_H$
RSCAN0	Receive buffer data field 0 register 13	RSCAN0RMDF013	<RSCAN0_base> + $06D8_H$
RSCAN0	Receive buffer data field 1 register 13	RSCAN0RMDF113	<RSCAN0_base> + $06DC_H$
RSCAN0	Receive buffer ID register 14	RSCAN0RMID14	<RSCAN0_base> + $06E0_H$
RSCAN0	Receive buffer pointer register 14	RSCAN0RMPTR14	<RSCAN0_base> + $06E4_H$
RSCAN0	Receive buffer data field 0 register 14	RSCAN0RMDF014	<RSCAN0_base> + $06E8_H$
RSCAN0	Receive buffer data field 1 register 14	RSCAN0RMDF114	<RSCAN0_base> + $06EC_H$
RSCAN0	Receive buffer ID register 15	RSCAN0RMID15	<RSCAN0_base> + $06F0_H$
RSCAN0	Receive buffer pointer register 15	RSCAN0RMPTR15	<RSCAN0_base> + $06F4_H$
RSCAN0	Receive buffer data field 0 register 15	RSCAN0RMDF015	<RSCAN0_base> + $06F8_H$
RSCAN0	Receive buffer data field 1 register 15	RSCAN0RMDF115	<RSCAN0_base> + $06FC_H$
RSCAN0	Receive buffer ID register 16	RSCAN0RMID16	<RSCAN0_base> + 0700_H
RSCAN0	Receive buffer pointer register 16	RSCAN0RMPTR16	<RSCAN0_base> + 0704_H
RSCAN0	Receive buffer data field 0 register 16	RSCAN0RMDF016	<RSCAN0_base> + 0708_H
RSCAN0	Receive buffer data field 1 register16	RSCAN0RMDF116	<RSCAN0_base> + $070C_H$
RSCAN0	Receive buffer ID register 17	RSCAN0RMID17	<RSCAN0_base> + 0710_H
RSCAN0	Receive buffer pointer register 17	RSCAN0RMPTR17	<RSCAN0_base> + 0714_H
RSCAN0	Receive buffer data field 0 register 17	RSCAN0RMDF017	<RSCAN0_base> + 0718_H
RSCAN0	Receive buffer data field 1 register 17	RSCAN0RMDF117	<RSCAN0_base> + $071C_H$
RSCAN0	Receive buffer ID register 18	RSCAN0RMID18	<RSCAN0_base> + 0720_H
RSCAN0	Receive buffer pointer register 18	RSCAN0RMPTR18	<RSCAN0_base> + 0724_H
RSCAN0	Receive buffer data field 0 register 18	RSCAN0RMDF018	<RSCAN0_base> + 0728_H

Table 22.13 Registers (9/19)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 1 register 18	RSCAN0RMDF118	<RSCAN0_base> + 072C$_H$
RSCAN0	Receive buffer ID register 19	RSCAN0RMID19	<RSCAN0_base> + 0730$_H$
RSCAN0	Receive buffer pointer register 19	RSCAN0RMPTR19	<RSCAN0_base> + 0734$_H$
RSCAN0	Receive buffer data field 0 register 19	RSCAN0RMDF019	<RSCAN0_base> + 0738$_H$
RSCAN0	Receive buffer data field 1 register 19	RSCAN0RMDF119	<RSCAN0_base> + 073C$_H$
RSCAN0	Receive buffer ID register 20	RSCAN0RMID20	<RSCAN0_base> + 0740$_H$
RSCAN0	Receive buffer pointer register 20	RSCAN0RMPTR20	<RSCAN0_base> + 0744$_H$
RSCAN0	Receive buffer data field 0 register 20	RSCAN0RMDF020	<RSCAN0_base> + 0748$_H$
RSCAN0	Receive buffer data field 1 register 20	RSCAN0RMDF120	<RSCAN0_base> + 074C$_H$
RSCAN0	Receive buffer ID register 21	RSCAN0RMID21	<RSCAN0_base> + 0750$_H$
RSCAN0	Receive buffer pointer register 21	RSCAN0RMPTR21	<RSCAN0_base> + 0754$_H$
RSCAN0	Receive buffer data field 0 register 21	RSCAN0RMDF021	<RSCAN0_base> + 0758$_H$
RSCAN0	Receive buffer data field 1 register 21	RSCAN0RMDF121	<RSCAN0_base> + 075C$_H$
RSCAN0	Receive buffer ID register 22	RSCAN0RMID22	<RSCAN0_base> + 0760$_H$
RSCAN0	Receive buffer pointer register 22	RSCAN0RMPTR22	<RSCAN0_base> + 0764$_H$
RSCAN0	Receive buffer data field 0 register 22	RSCAN0RMDF022	<RSCAN0_base> + 0768$_H$
RSCAN0	Receive buffer data field 1 register 22	RSCAN0RMDF122	<RSCAN0_base> + 076C$_H$
RSCAN0	Receive buffer ID register 23	RSCAN0RMID23	<RSCAN0_base> + 0770$_H$
RSCAN0	Receive buffer pointer register 23	RSCAN0RMPTR23	<RSCAN0_base> + 0774$_H$
RSCAN0	Receive buffer data field 0 register 23	RSCAN0RMDF023	<RSCAN0_base> + 0778$_H$
RSCAN0	Receive buffer data field 1 register 23	RSCAN0RMDF123	<RSCAN0_base> + 077C$_H$
RSCAN0	Receive buffer ID register 24	RSCAN0RMID24	<RSCAN0_base> + 0780$_H$
RSCAN0	Receive buffer pointer register 24	RSCAN0RMPTR24	<RSCAN0_base> + 0784$_H$
RSCAN0	Receive buffer data field 0 register 24	RSCAN0RMDF024	<RSCAN0_base> + 0788$_H$
RSCAN0	Receive buffer data field 1 register 24	RSCAN0RMDF124	<RSCAN0_base> + 078C$_H$
RSCAN0	Receive buffer ID register 25	RSCAN0RMID25	<RSCAN0_base> + 0790$_H$
RSCAN0	Receive buffer pointer register 25	RSCAN0RMPTR25	<RSCAN0_base> + 0794$_H$
RSCAN0	Receive buffer data field 0 register 25	RSCAN0RMDF025	<RSCAN0_base> + 0798$_H$
RSCAN0	Receive buffer data field 1 register 25	RSCAN0RMDF125	<RSCAN0_base> + 079C$_H$
RSCAN0	Receive buffer ID register 26	RSCAN0RMID26	<RSCAN0_base> + 07A0$_H$
RSCAN0	Receive buffer pointer register 26	RSCAN0RMPTR26	<RSCAN0_base> + 07A4$_H$
RSCAN0	Receive buffer data field 0 register 26	RSCAN0RMDF026	<RSCAN0_base> + 07A8$_H$
RSCAN0	Receive buffer data field 1 register 26	RSCAN0RMDF126	<RSCAN0_base> + 07AC$_H$
RSCAN0	Receive buffer ID register 27	RSCAN0RMID27	<RSCAN0_base> + 07B0$_H$
RSCAN0	Receive buffer pointer register 27	RSCAN0RMPTR27	<RSCAN0_base> + 07B4$_H$
RSCAN0	Receive buffer data field 0 register 27	RSCAN0RMDF027	<RSCAN0_base> + 07B8$_H$
RSCAN0	Receive buffer data field 1 register 27	RSCAN0RMDF127	<RSCAN0_base> + 07BC$_H$
RSCAN0	Receive buffer ID register 28	RSCAN0RMID28	<RSCAN0_base> + 07C0$_H$
RSCAN0	Receive buffer pointer register 28	RSCAN0RMPTR28	<RSCAN0_base> + 07C4$_H$
RSCAN0	Receive buffer data field 0 register 28	RSCAN0RMDF028	<RSCAN0_base> + 07C8$_H$
RSCAN0	Receive buffer data field 1 register 28	RSCAN0RMDF128	<RSCAN0_base> + 07CC$_H$
RSCAN0	Receive buffer ID register 29	RSCAN0RMID29	<RSCAN0_base> + 07D0$_H$
RSCAN0	Receive buffer pointer register 29	RSCAN0RMPTR29	<RSCAN0_base> + 07D4$_H$
RSCAN0	Receive buffer data field 0 register 29	RSCAN0RMDF029	<RSCAN0_base> + 07D8$_H$

Table 22.13 Registers (10/19)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 1 register 29	RSCAN0RMDF129	<RSCAN0_base> + $07DC_H$
RSCAN0	Receive buffer ID register 30	RSCAN0RMID30	<RSCAN0_base> + $07E0_H$
RSCAN0	Receive buffer pointer register 30	RSCAN0RMPTR30	<RSCAN0_base> + $07E4_H$
RSCAN0	Receive buffer data field 0 register 30	RSCAN0RMDF030	<RSCAN0_base> + $07E8_H$
RSCAN0	Receive buffer data field 1 register 30	RSCAN0RMDF130	<RSCAN0_base> + $07EC_H$
RSCAN0	Receive buffer ID register 31	RSCAN0RMID31	<RSCAN0_base> + $07F0_H$
RSCAN0	Receive buffer pointer register 31	RSCAN0RMPTR31	<RSCAN0_base> + $07F4_H$
RSCAN0	Receive buffer data field 0 register 31	RSCAN0RMDF031	<RSCAN0_base> + $07F8_H$
RSCAN0	Receive buffer data field 1 register 31	RSCAN0RMDF131	<RSCAN0_base> + $07FC_H$
RSCAN0	Receive buffer ID register 32	RSCAN0RMID32	<RSCAN0_base> + 0800_H
RSCAN0	Receive buffer pointer register 32	RSCAN0RMPTR32	<RSCAN0_base> + 0804_H
RSCAN0	Receive buffer data field 0 register 32	RSCAN0RMDF032	<RSCAN0_base> + 0808_H
RSCAN0	Receive buffer data field 1 register 32	RSCAN0RMDF132	<RSCAN0_base> + $080C_H$
RSCAN0	Receive buffer ID register 33	RSCAN0RMID33	<RSCAN0_base> + 0810_H
RSCAN0	Receive buffer pointer register 33	RSCAN0RMPTR33	<RSCAN0_base> + 0814_H
RSCAN0	Receive buffer data field 0 register 33	RSCAN0RMDF033	<RSCAN0_base> + 0818_H
RSCAN0	Receive buffer data field 1 register 33	RSCAN0RMDF133	<RSCAN0_base> + $081C_H$
RSCAN0	Receive buffer ID register 34	RSCAN0RMID34	<RSCAN0_base> + 0820_H
RSCAN0	Receive buffer pointer register 34	RSCAN0RMPTR34	<RSCAN0_base> + 0824_H
RSCAN0	Receive buffer data field 0 register 34	RSCAN0RMDF034	<RSCAN0_base> + 0828_H
RSCAN0	Receive buffer data field 1 register 34	RSCAN0RMDF134	<RSCAN0_base> + $082C_H$
RSCAN0	Receive buffer ID register 35	RSCAN0RMID35	<RSCAN0_base> + 0830_H
RSCAN0	Receive buffer pointer register 35	RSCAN0RMPTR35	<RSCAN0_base> + 0834_H
RSCAN0	Receive buffer data field 0 register 35	RSCAN0RMDF035	<RSCAN0_base> + 0838_H
RSCAN0	Receive buffer data field 1 register 35	RSCAN0RMDF135	<RSCAN0_base> + $083C_H$
RSCAN0	Receive buffer ID register 36	RSCAN0RMID36	<RSCAN0_base> + 0840_H
RSCAN0	Receive buffer pointer register 36	RSCAN0RMPTR36	<RSCAN0_base> + 0844_H
RSCAN0	Receive buffer data field 0 register 36	RSCAN0RMDF036	<RSCAN0_base> + 0848_H
RSCAN0	Receive buffer data field 1 register 36	RSCAN0RMDF136	<RSCAN0_base> + $084C_H$
RSCAN0	Receive buffer ID register 37	RSCAN0RMID37	<RSCAN0_base> + 0850_H
RSCAN0	Receive buffer pointer register 37	RSCAN0RMPTR37	<RSCAN0_base> + 0854_H
RSCAN0	Receive buffer data field 0 register 37	RSCAN0RMDF037	<RSCAN0_base> + 0858_H
RSCAN0	Receive buffer data field 1 register 37	RSCAN0RMDF137	<RSCAN0_base> + $085C_H$
RSCAN0	Receive buffer ID register 38	RSCAN0RMID38	<RSCAN0_base> + 0860_H
RSCAN0	Receive buffer pointer register 38	RSCAN0RMPTR38	<RSCAN0_base> + 0864_H
RSCAN0	Receive buffer data field 0 register 38	RSCAN0RMDF038	<RSCAN0_base> + 0868_H
RSCAN0	Receive buffer data field 1 register 38	RSCAN0RMDF138	<RSCAN0_base> + $086C_H$
RSCAN0	Receive buffer ID register 39	RSCAN0RMID39	<RSCAN0_base> + 0870_H
RSCAN0	Receive buffer pointer register 39	RSCAN0RMPTR39	<RSCAN0_base> + 0874_H
RSCAN0	Receive buffer data field 0 register 39	RSCAN0RMDF039	<RSCAN0_base> + 0878_H
RSCAN0	Receive buffer data field 1 register 39	RSCAN0RMDF139	<RSCAN0_base> + $087C_H$
RSCAN0	Receive buffer ID register 40	RSCAN0RMID40	<RSCAN0_base> + 0880_H
RSCAN0	Receive buffer pointer register 40	RSCAN0RMPTR40	<RSCAN0_base> + 0884_H
RSCAN0	Receive buffer data field 0 register 40	RSCAN0RMDF040	<RSCAN0_base> + 0888_H

Table 22.13 Registers (11/19)

Module	Register	Symbol	Address
RSCAN0	Receive buffer data field 1 register 40	RSCAN0RMDF140	<RSCAN0_base> + $088C_H$
RSCAN0	Receive buffer ID register 41	RSCAN0RMID41	<RSCAN0_base> + 0890_H
RSCAN0	Receive buffer pointer register 41	RSCAN0RMPTR41	<RSCAN0_base> + 0894_H
RSCAN0	Receive buffer data field 0 register 41	RSCAN0RMDF041	<RSCAN0_base> + 0898_H
RSCAN0	Receive buffer data field 1 register 41	RSCAN0RMDF141	<RSCAN0_base> + $089C_H$
RSCAN0	Receive buffer ID register 42	RSCAN0RMID42	<RSCAN0_base> + $08A0_H$
RSCAN0	Receive buffer pointer register 42	RSCAN0RMPTR42	<RSCAN0_base> + $08A4_H$
RSCAN0	Receive buffer data field 0 register 42	RSCAN0RMDF042	<RSCAN0_base> + $08A8_H$
RSCAN0	Receive buffer data field 1 register 42	RSCAN0RMDF142	<RSCAN0_base> + $08AC_H$
RSCAN0	Receive buffer ID register 43	RSCAN0RMID43	<RSCAN0_base> + $08B0_H$
RSCAN0	Receive buffer pointer register 43	RSCAN0RMPTR43	<RSCAN0_base> + $08B4_H$
RSCAN0	Receive buffer data field 0 register 43	RSCAN0RMDF043	<RSCAN0_base> + $08B8_H$
RSCAN0	Receive buffer data field 1 register 43	RSCAN0RMDF143	<RSCAN0_base> + $08BC_H$
RSCAN0	Receive buffer ID register 44	RSCAN0RMID44	<RSCAN0_base> + $08C0_H$
RSCAN0	Receive buffer pointer register 44	RSCAN0RMPTR44	<RSCAN0_base> + $08C4_H$
RSCAN0	Receive buffer data field 0 register 44	RSCAN0RMDF044	<RSCAN0_base> + $08C8_H$
RSCAN0	Receive buffer data field 1 register 44	RSCAN0RMDF144	<RSCAN0_base> + $08CC_H$
RSCAN0	Receive buffer ID register 45	RSCAN0RMID45	<RSCAN0_base> + $08D0_H$
RSCAN0	Receive buffer pointer register 45	RSCAN0RMPTR45	<RSCAN0_base> + $08D4_H$
RSCAN0	Receive buffer data field 0 register 45	RSCAN0RMDF045	<RSCAN0_base> + $08D8_H$
RSCAN0	Receive buffer data field 1 register 45	RSCAN0RMDF145	<RSCAN0_base> + $08DC_H$
RSCAN0	Receive buffer ID register 46	RSCAN0RMID46	<RSCAN0_base> + $08E0_H$
RSCAN0	Receive buffer pointer register 46	RSCAN0RMPTR46	<RSCAN0_base> + $08E4_H$
RSCAN0	Receive buffer data field 0 register 46	RSCAN0RMDF046	<RSCAN0_base> + $08E8_H$
RSCAN0	Receive buffer data field 1 register 46	RSCAN0RMDF146	<RSCAN0_base> + $08EC_H$
RSCAN0	Receive buffer ID register 47	RSCAN0RMID47	<RSCAN0_base> + $08F0_H$
RSCAN0	Receive buffer pointer register 47	RSCAN0RMPTR47	<RSCAN0_base> + $08F4_H$
RSCAN0	Receive buffer data field 0 register 47	RSCAN0RMDF047	<RSCAN0_base> + $08F8_H$
RSCAN0	Receive buffer data field 1 register 47	RSCAN0RMDF147	<RSCAN0_base> + $08FC_H$
RSCAN0	Receive FIFO buffer access ID register 0	RSCAN0RFID0	<RSCAN0_base> + $0E00_H$
RSCAN0	Receive FIFO buffer access pointer register 0	RSCAN0RFPTR0	<RSCAN0_base> + $0E04_H$
RSCAN0	Receive FIFO buffer access data field 0 register 0	RSCAN0RFDF00	<RSCAN0_base> + $0E08_H$
RSCAN0	Receive FIFO buffer access data field 1 register 0	RSCAN0RFDF10	<RSCAN0_base> + $0E0C_H$
RSCAN0	Receive FIFO buffer access ID register 1	RSCAN0RFID1	<RSCAN0_base> + $0E10_H$
RSCAN0	Receive FIFO buffer access pointer register 1	RSCAN0RFPTR1	<RSCAN0_base> + $0E14_H$
RSCAN0	Receive FIFO buffer access data field 0 register 1	RSCAN0RFDF01	<RSCAN0_base> + $0E18_H$
RSCAN0	Receive FIFO buffer access data field 1 register 1	RSCAN0RFDF11	<RSCAN0_base> + $0E1C_H$
RSCAN0	Receive FIFO buffer access ID register 2	RSCAN0RFID2	<RSCAN0_base> + $0E20_H$
RSCAN0	Receive FIFO buffer access pointer register 2	RSCAN0RFPTR2	<RSCAN0_base> + $0E24_H$
RSCAN0	Receive FIFO buffer access data field 0 register 2	RSCAN0RFDF02	<RSCAN0_base> + $0E28_H$
RSCAN0	Receive FIFO buffer access data field 1 register 2	RSCAN0RFDF12	<RSCAN0_base> + $0E2C_H$
RSCAN0	Receive FIFO buffer access ID register 3	RSCAN0RFID3	<RSCAN0_base> + $0E30_H$
RSCAN0	Receive FIFO buffer access pointer register 3	RSCAN0RFPTR3	<RSCAN0_base> + $0E34_H$
RSCAN0	Receive FIFO buffer access data field 0 register 3	RSCAN0RFDF03	<RSCAN0_base> + $0E38_H$

Table 22.13 Registers (12/19)

Module	Register	Symbol	Address
RSCAN0	Receive FIFO buffer access data field 1 register 3	RSCAN0RFDF13	<RSCAN0_base> + $0E3C_H$
RSCAN0	Receive FIFO buffer access ID register 4	RSCAN0RFID4	<RSCAN0_base> + $0E40_H$
RSCAN0	Receive FIFO buffer access pointer register 4	RSCAN0RFPTR4	<RSCAN0_base> + $0E44_H$
RSCAN0	Receive FIFO buffer access data field 0 register 4	RSCAN0RFDF04	<RSCAN0_base> + $0E48_H$
RSCAN0	Receive FIFO buffer access data field 1 register 4	RSCAN0RFDF14	<RSCAN0_base> + $0E4C_H$
RSCAN0	Receive FIFO buffer access ID register 5	RSCAN0RFID5	<RSCAN0_base> + $0E50_H$
RSCAN0	Receive FIFO buffer access pointer register 5	RSCAN0RFPTR5	<RSCAN0_base> + $0E54_H$
RSCAN0	Receive FIFO buffer access data field 0 register 5	RSCAN0RFDF05	<RSCAN0_base> + $0E58_H$
RSCAN0	Receive FIFO buffer access data field 1 register 5	RSCAN0RFDF15	<RSCAN0_base> + $0E5C_H$
RSCAN0	Receive FIFO buffer access ID register 6	RSCAN0RFID6	<RSCAN0_base> + $0E60_H$
RSCAN0	Receive FIFO buffer access pointer register 6	RSCAN0RFPTR6	<RSCAN0_base> + $0E64_H$
RSCAN0	Receive FIFO buffer access data field 0 register 6	RSCAN0RFDF06	<RSCAN0_base> + $0E68_H$
RSCAN0	Receive FIFO buffer access data field 1 register 6	RSCAN0RFDF16	<RSCAN0_base> + $0E6C_H$
RSCAN0	Receive FIFO buffer access ID register 7	RSCAN0RFID7	<RSCAN0_base> + $0E70_H$
RSCAN0	Receive FIFO buffer access pointer register 7	RSCAN0RFPTR7	<RSCAN0_base> + $0E74_H$
RSCAN0	Receive FIFO buffer access data field 0 register 7	RSCAN0RFDF07	<RSCAN0_base> + $0E78_H$
RSCAN0	Receive FIFO buffer access data field 1 register 7	RSCAN0RFDF17	<RSCAN0_base> + $0E7C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 0	RSCAN0CFID0	<RSCAN0_base> + $0E80_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 0	RSCAN0CFPTR0	<RSCAN0_base> + $0E84_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 0	RSCAN0CFDF00	<RSCAN0_base> + $0E88_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 0	RSCAN0CFDF10	<RSCAN0_base> + $0E8C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 1	RSCAN0CFID1	<RSCAN0_base> + $0E90_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 1	RSCAN0CFPTR1	<RSCAN0_base> + $0E94_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 1	RSCAN0CFDF01	<RSCAN0_base> + $0E98_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 1	RSCAN0CFDF11	<RSCAN0_base> + $0E9C_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 2	RSCAN0CFID2	<RSCAN0_base> + $0EA0_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 2	RSCAN0CFPTR2	<RSCAN0_base> + $0EA4_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 2	RSCAN0CFDF02	<RSCAN0_base> + $0EA8_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 2	RSCAN0CFDF12	<RSCAN0_base> + $0EAC_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 3	RSCAN0CFID3	<RSCAN0_base> + $0EB0_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 3	RSCAN0CFPTR3	<RSCAN0_base> + $0EB4_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 3	RSCAN0CFDF03	<RSCAN0_base> + $0EB8_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 3	RSCAN0CFDF13	<RSCAN0_base> + $0EBC_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 4	RSCAN0CFID4	<RSCAN0_base> + $0EC0_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 4	RSCAN0CFPTR4	<RSCAN0_base> + $0EC4_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 4	RSCAN0CFDF04	<RSCAN0_base> + $0EC8_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 4	RSCAN0CFDF14	<RSCAN0_base> + $0ECC_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 5	RSCAN0CFID5	<RSCAN0_base> + $0ED0_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 5	RSCAN0CFPTR5	<RSCAN0_base> + $0ED4_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 5	RSCAN0CFDF05	<RSCAN0_base> + $0ED8_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 5	RSCAN0CFDF15	<RSCAN0_base> + $0EDC_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 6	RSCAN0CFID6	<RSCAN0_base> + $0EE0_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 6	RSCAN0CFPTR6	<RSCAN0_base> + $0EE4_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 6	RSCAN0CFDF06	<RSCAN0_base> + $0EE8_H$

Table 22.13 Registers (13/19)

Module	Register	Symbol	Address
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 6	RSCAN0CFDF16	<RSCAN0_base> + $0EEC_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 7	RSCAN0CFID7	<RSCAN0_base> + $0EF0_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 7	RSCAN0CFPTR7	<RSCAN0_base> + $0EF4_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 7	RSCAN0CFDF07	<RSCAN0_base> + $0EF8_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 7	RSCAN0CFDF17	<RSCAN0_base> + $0EFC_H$
RSCAN0	Transmit/receive FIFO buffer access ID register 8	RSCAN0CFID8	<RSCAN0_base> + $0F00_H$
RSCAN0	Transmit/receive FIFO buffer access pointer register 8	RSCAN0CFPTR8	<RSCAN0_base> + $0F04_H$
RSCAN0	Transmit/receive FIFO buffer access data field 0 register 8	RSCAN0CFDF08	<RSCAN0_base> + $0F08_H$
RSCAN0	Transmit/receive FIFO buffer access data field 1 register 8	RSCAN0CFDF18	<RSCAN0_base> + $0F0C_H$
RSCAN0	Transmit buffer ID register 0	RSCAN0TMID0	<RSCAN0_base> + 1000_H
RSCAN0	Transmit buffer pointer register 0	RSCAN0TMPTR0	<RSCAN0_base> + 1004_H
RSCAN0	Transmit buffer data field 0 register 0	RSCAN0TMDF00	<RSCAN0_base> + 1008_H
RSCAN0	Transmit buffer data field 1 register 0	RSCAN0TMDF10	<RSCAN0_base> + $100C_H$
RSCAN0	Transmit buffer ID register 1	RSCAN0TMID1	<RSCAN0_base> + 1010_H
RSCAN0	Transmit buffer pointer register 1	RSCAN0TMPTR1	<RSCAN0_base> + 1014_H
RSCAN0	Transmit buffer data field 0 register 1	RSCAN0TMDF01	<RSCAN0_base> + 1018_H
RSCAN0	Transmit buffer data field 1 register 1	RSCAN0TMDF11	<RSCAN0_base> + $101C_H$
RSCAN0	Transmit buffer ID register 2	RSCAN0TMID2	<RSCAN0_base> + 1020_H
RSCAN0	Transmit buffer pointer register 2	RSCAN0TMPTR2	<RSCAN0_base> + 1024_H
RSCAN0	Transmit buffer data field 0 register 2	RSCAN0TMDF02	<RSCAN0_base> + 1028_H
RSCAN0	Transmit buffer data field 1 register 2	RSCAN0TMDF12	<RSCAN0_base> + $102C_H$
RSCAN0	Transmit buffer ID register 3	RSCAN0TMID3	<RSCAN0_base> + 1030_H
RSCAN0	Transmit buffer pointer register 3	RSCAN0TMPTR3	<RSCAN0_base> + 1034_H
RSCAN0	Transmit buffer data field 0 register 3	RSCAN0TMDF03	<RSCAN0_base> + 1038_H
RSCAN0	Transmit buffer data field 1 register 3	RSCAN0TMDF13	<RSCAN0_base> + $103C_H$
RSCAN0	Transmit buffer ID register 4	RSCAN0TMID4	<RSCAN0_base> + 1040_H
RSCAN0	Transmit buffer pointer register 4	RSCAN0TMPTR4	<RSCAN0_base> + 1044_H
RSCAN0	Transmit buffer data field 0 register 4	RSCAN0TMDF04	<RSCAN0_base> + 1048_H
RSCAN0	Transmit buffer data field 1 register 4	RSCAN0TMDF14	<RSCAN0_base> + $104C_H$
RSCAN0	Transmit buffer ID register 5	RSCAN0TMID5	<RSCAN0_base> + 1050_H
RSCAN0	Transmit buffer pointer register 5	RSCAN0TMPTR5	<RSCAN0_base> + 1054_H
RSCAN0	Transmit buffer data field 0 register 5	RSCAN0TMDF05	<RSCAN0_base> + 1058_H
RSCAN0	Transmit buffer data field 1 register 5	RSCAN0TMDF15	<RSCAN0_base> + $105C_H$
RSCAN0	Transmit buffer ID register 6	RSCAN0TMID6	<RSCAN0_base> + 1060_H
RSCAN0	Transmit buffer pointer register 6	RSCAN0TMPTR6	<RSCAN0_base> + 1064_H
RSCAN0	Transmit buffer data field 0 register 6	RSCAN0TMDF06	<RSCAN0_base> + 1068_H
RSCAN0	Transmit buffer data field 1 register 6	RSCAN0TMDF16	<RSCAN0_base> + $106C_H$
RSCAN0	Transmit buffer ID register 7	RSCAN0TMID7	<RSCAN0_base> + 1070_H
RSCAN0	Transmit buffer pointer register 7	RSCAN0TMPTR7	<RSCAN0_base> + 1074_H
RSCAN0	Transmit buffer data field 0 register 7	RSCAN0TMDF07	<RSCAN0_base> + 1078_H
RSCAN0	Transmit buffer data field 1 register 7	RSCAN0TMDF17	<RSCAN0_base> + $107C_H$
RSCAN0	Transmit buffer ID register 8	RSCAN0TMID8	<RSCAN0_base> + 1080_H
RSCAN0	Transmit buffer pointer register 8	RSCAN0TMPTR8	<RSCAN0_base> + 1084_H
RSCAN0	Transmit buffer data field 0 register 8	RSCAN0TMDF08	<RSCAN0_base> + 1088_H

Table 22.13 Registers (14/19)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 1 register 8	RSCAN0TMDF18	<RSCAN0_base> + $108C_H$
RSCAN0	Transmit buffer ID register 9	RSCAN0TMID9	<RSCAN0_base> + 1090_H
RSCAN0	Transmit buffer pointer register 9	RSCAN0TMPTR9	<RSCAN0_base> + 1094_H
RSCAN0	Transmit buffer data field 0 register 9	RSCAN0TMDF09	<RSCAN0_base> + 1098_H
RSCAN0	Transmit buffer data field 1 register 9	RSCAN0TMDF19	<RSCAN0_base> + $109C_H$
RSCAN0	Transmit buffer ID register 10	RSCAN0TMID10	<RSCAN0_base> + $10A0_H$
RSCAN0	Transmit buffer pointer register 10	RSCAN0TMPTR10	<RSCAN0_base> + $10A4_H$
RSCAN0	Transmit buffer data field 0 register 10	RSCAN0TMDF010	<RSCAN0_base> + $10A8_H$
RSCAN0	Transmit buffer data field 1 register 10	RSCAN0TMDF110	<RSCAN0_base> + $10AC_H$
RSCAN0	Transmit buffer ID register 11	RSCAN0TMID11	<RSCAN0_base> + $10B0_H$
RSCAN0	Transmit buffer pointer register 11	RSCAN0TMPTR11	<RSCAN0_base> + $10B4_H$
RSCAN0	Transmit buffer data field 0 register 11	RSCAN0TMDF011	<RSCAN0_base> + $10B8_H$
RSCAN0	Transmit buffer data field 1 register 11	RSCAN0TMDF111	<RSCAN0_base> + $10BC_H$
RSCAN0	Transmit buffer ID register 12	RSCAN0TMID12	<RSCAN0_base> + $10C0_H$
RSCAN0	Transmit buffer pointer register 12	RSCAN0TMPTR12	<RSCAN0_base> + $10C4_H$
RSCAN0	Transmit buffer data field 0 register 12	RSCAN0TMDF012	<RSCAN0_base> + $10C8_H$
RSCAN0	Transmit buffer data field 1 register 12	RSCAN0TMDF112	<RSCAN0_base> + $10CC_H$
RSCAN0	Transmit buffer ID register 13	RSCAN0TMID13	<RSCAN0_base> + $10D0_H$
RSCAN0	Transmit buffer pointer register 13	RSCAN0TMPTR13	<RSCAN0_base> + $10D4_H$
RSCAN0	Transmit buffer data field 0 register 13	RSCAN0TMDF013	<RSCAN0_base> + $10D8_H$
RSCAN0	Transmit buffer data field 1 register 13	RSCAN0TMDF113	<RSCAN0_base> + $10DC_H$
RSCAN0	Transmit buffer ID register 14	RSCAN0TMID14	<RSCAN0_base> + $10E0_H$
RSCAN0	Transmit buffer pointer register 14	RSCAN0TMPTR14	<RSCAN0_base> + $10E4_H$
RSCAN0	Transmit buffer data field 0 register 14	RSCAN0TMDF014	<RSCAN0_base> + $10E8_H$
RSCAN0	Transmit buffer data field 1 register 14	RSCAN0TMDF114	<RSCAN0_base> + $10EC_H$
RSCAN0	Transmit buffer ID register 15	RSCAN0TMID15	<RSCAN0_base> + $10F0_H$
RSCAN0	Transmit buffer pointer register 15	RSCAN0TMPTR15	<RSCAN0_base> + $10F4_H$
RSCAN0	Transmit buffer data field 0 register 15	RSCAN0TMDF015	<RSCAN0_base> + $10F8_H$
RSCAN0	Transmit buffer data field 1 register 15	RSCAN0TMDF115	<RSCAN0_base> + $10FC_H$
RSCAN0	Transmit buffer ID register 16	RSCAN0TMID16	<RSCAN0_base> + 1100_H
RSCAN0	Transmit buffer pointer register 16	RSCAN0TMPTR16	<RSCAN0_base> + 1104_H
RSCAN0	Transmit buffer data field 0 register 16	RSCAN0TMDF016	<RSCAN0_base> + 1108_H
RSCAN0	Transmit buffer data field 1 register 16	RSCAN0TMDF116	<RSCAN0_base> + $110C_H$
RSCAN0	Transmit buffer ID register 17	RSCAN0TMID17	<RSCAN0_base> + 1110_H
RSCAN0	Transmit buffer pointer register 17	RSCAN0TMPTR17	<RSCAN0_base> + 1114_H
RSCAN0	Transmit buffer data field 0 register 17	RSCAN0TMDF017	<RSCAN0_base> + 1118_H
RSCAN0	Transmit buffer data field 1 register 17	RSCAN0TMDF117	<RSCAN0_base> + $111C_H$
RSCAN0	Transmit buffer ID register 18	RSCAN0TMID18	<RSCAN0_base> + 1120_H
RSCAN0	Transmit buffer pointer register 18	RSCAN0TMPTR18	<RSCAN0_base> + 1124_H
RSCAN0	Transmit buffer data field 0 register 18	RSCAN0TMDF018	<RSCAN0_base> + 1128_H
RSCAN0	Transmit buffer data field 1 register 18	RSCAN0TMDF118	<RSCAN0_base> + $112C_H$
RSCAN0	Transmit buffer ID register 19	RSCAN0TMID19	<RSCAN0_base> + 1130_H
RSCAN0	Transmit buffer pointer register 19	RSCAN0TMPTR19	<RSCAN0_base> + 1134_H
RSCAN0	Transmit buffer data field 0 register 19	RSCAN0TMDF019	<RSCAN0_base> + 1138_H

Table 22.13 Registers (15/19)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 1 register 19	RSCAN0TMDF119	<RSCAN0_base> + $113C_H$
RSCAN0	Transmit buffer ID register 20	RSCAN0TMID20	<RSCAN0_base> + 1140_H
RSCAN0	Transmit buffer pointer register 20	RSCAN0TMPTR20	<RSCAN0_base> + 1144_H
RSCAN0	Transmit buffer data field 0 register 20	RSCAN0TMDF020	<RSCAN0_base> + 1148_H
RSCAN0	Transmit buffer data field 1 register 20	RSCAN0TMDF120	<RSCAN0_base> + $114C_H$
RSCAN0	Transmit buffer ID register 21	RSCAN0TMID21	<RSCAN0_base> + 1150_H
RSCAN0	Transmit buffer pointer register 21	RSCAN0TMPTR21	<RSCAN0_base> + 1154_H
RSCAN0	Transmit buffer data field 0 register 21	RSCAN0TMDF021	<RSCAN0_base> + 1158_H
RSCAN0	Transmit buffer data field 1 register 21	RSCAN0TMDF121	<RSCAN0_base> + $115C_H$
RSCAN0	Transmit buffer ID register 22	RSCAN0TMID22	<RSCAN0_base> + 1160_H
RSCAN0	Transmit buffer pointer register 22	RSCAN0TMPTR22	<RSCAN0_base> + 1164_H
RSCAN0	Transmit buffer data field 0 register 22	RSCAN0TMDF022	<RSCAN0_base> + 1168_H
RSCAN0	Transmit buffer data field 1 register 22	RSCAN0TMDF122	<RSCAN0_base> + $116C_H$
RSCAN0	Transmit buffer ID register 23	RSCAN0TMID23	<RSCAN0_base> + 1170_H
RSCAN0	Transmit buffer pointer register 23	RSCAN0TMPTR23	<RSCAN0_base> + 1174_H
RSCAN0	Transmit buffer data field 0 register 23	RSCAN0TMDF023	<RSCAN0_base> + 1178_H
RSCAN0	Transmit buffer data field 1 register 23	RSCAN0TMDF123	<RSCAN0_base> + $117C_H$
RSCAN0	Transmit buffer ID register 24	RSCAN0TMID24	<RSCAN0_base> + 1180_H
RSCAN0	Transmit buffer pointer register 24	RSCAN0TMPTR24	<RSCAN0_base> + 1184_H
RSCAN0	Transmit buffer data field 0 register 24	RSCAN0TMDF024	<RSCAN0_base> + 1188_H
RSCAN0	Transmit buffer data field 1 register 24	RSCAN0TMDF124	<RSCAN0_base> + $118C_H$
RSCAN0	Transmit buffer ID register 25	RSCAN0TMID25	<RSCAN0_base> + 1190_H
RSCAN0	Transmit buffer pointer register 25	RSCAN0TMPTR25	<RSCAN0_base> + 1194_H
RSCAN0	Transmit buffer data field 0 register 25	RSCAN0TMDF025	<RSCAN0_base> + 1198_H
RSCAN0	Transmit buffer data field 1 register 25	RSCAN0TMDF125	<RSCAN0_base> + $119C_H$
RSCAN0	Transmit buffer ID register 26	RSCAN0TMID26	<RSCAN0_base> + $11A0_H$
RSCAN0	Transmit buffer pointer register 26	RSCAN0TMPTR26	<RSCAN0_base> + $11A4_H$
RSCAN0	Transmit buffer data field 0 register 26	RSCAN0TMDF026	<RSCAN0_base> + $11A8_H$
RSCAN0	Transmit buffer data field 1 register 26	RSCAN0TMDF126	<RSCAN0_base> + $11AC_H$
RSCAN0	Transmit buffer ID register 27	RSCAN0TMID27	<RSCAN0_base> + $11B0_H$
RSCAN0	Transmit buffer pointer register 27	RSCAN0TMPTR27	<RSCAN0_base> + $11B4_H$
RSCAN0	Transmit buffer data field 0 register 27	RSCAN0TMDF027	<RSCAN0_base> + $11B8_H$
RSCAN0	Transmit buffer data field 1 register 27	RSCAN0TMDF127	<RSCAN0_base> + $11BC_H$
RSCAN0	Transmit buffer ID register 28	RSCAN0TMID28	<RSCAN0_base> + $11C0_H$
RSCAN0	Transmit buffer pointer register 28	RSCAN0TMPTR28	<RSCAN0_base> + $11C4_H$
RSCAN0	Transmit buffer data field 0 register 28	RSCAN0TMDF028	<RSCAN0_base> + $11C8_H$
RSCAN0	Transmit buffer data field 1 register 28	RSCAN0TMDF128	<RSCAN0_base> + $11CC_H$
RSCAN0	Transmit buffer ID register 29	RSCAN0TMID29	<RSCAN0_base> + $11D0_H$
RSCAN0	Transmit buffer pointer register 29	RSCAN0TMPTR29	<RSCAN0_base> + $11D4_H$
RSCAN0	Transmit buffer data field 0 register 29	RSCAN0TMDF029	<RSCAN0_base> + $11D8_H$
RSCAN0	Transmit buffer data field 1 register 29	RSCAN0TMDF129	<RSCAN0_base> + $11DC_H$
RSCAN0	Transmit buffer ID register 30	RSCAN0TMID30	<RSCAN0_base> + $11E0_H$
RSCAN0	Transmit buffer pointer register 30	RSCAN0TMPTR30	<RSCAN0_base> + $11E4_H$
RSCAN0	Transmit buffer data field 0 register 30	RSCAN0TMDF030	<RSCAN0_base> + $11E8_H$

Table 22.13 Registers (16/19)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 1 register 30	RSCAN0TMDF130	<RSCAN0_base> + $11EC_H$
RSCAN0	Transmit buffer ID register 31	RSCAN0TMID31	<RSCAN0_base> + $11F0_H$
RSCAN0	Transmit buffer pointer register 31	RSCAN0TMPTR31	<RSCAN0_base> + $11F4_H$
RSCAN0	Transmit buffer data field 0 register 31	RSCAN0TMDF031	<RSCAN0_base> + $11F8_H$
RSCAN0	Transmit buffer data field 1 register 31	RSCAN0TMDF131	<RSCAN0_base> + $11FC_H$
RSCAN0	Transmit buffer ID register 32	RSCAN0TMID32	<RSCAN0_base> + 1200_H
RSCAN0	Transmit buffer pointer register 32	RSCAN0TMPTR32	<RSCAN0_base> + 1204_H
RSCAN0	Transmit buffer data field 0 register 32	RSCAN0TMDF032	<RSCAN0_base> + 1208_H
RSCAN0	Transmit buffer data field 1 register 32	RSCAN0TMDF132	<RSCAN0_base> + $120C_H$
RSCAN0	Transmit buffer ID register 33	RSCAN0TMID33	<RSCAN0_base> + 1210_H
RSCAN0	Transmit buffer pointer register 33	RSCAN0TMPTR33	<RSCAN0_base> + 1214_H
RSCAN0	Transmit buffer data field 0 register 33	RSCAN0TMDF033	<RSCAN0_base> + 1218_H
RSCAN0	Transmit buffer data field 1 register 33	RSCAN0TMDF133	<RSCAN0_base> + $121C_H$
RSCAN0	Transmit buffer ID register 34	RSCAN0TMID34	<RSCAN0_base> + 1220_H
RSCAN0	Transmit buffer pointer register 34	RSCAN0TMPTR34	<RSCAN0_base> + 1224_H
RSCAN0	Transmit buffer data field 0 register 34	RSCAN0TMDF034	<RSCAN0_base> + 1228_H
RSCAN0	Transmit buffer data field 1 register 34	RSCAN0TMDF134	<RSCAN0_base> + $122C_H$
RSCAN0	Transmit buffer ID register 35	RSCAN0TMID35	<RSCAN0_base> + 1230_H
RSCAN0	Transmit buffer pointer register 35	RSCAN0TMPTR35	<RSCAN0_base> + 1234_H
RSCAN0	Transmit buffer data field 0 register 35	RSCAN0TMDF035	<RSCAN0_base> + 1238_H
RSCAN0	Transmit buffer data field 1 register 35	RSCAN0TMDF135	<RSCAN0_base> + $123C_H$
RSCAN0	Transmit buffer ID register 36	RSCAN0TMID36	<RSCAN0_base> + 1240_H
RSCAN0	Transmit buffer pointer register 36	RSCAN0TMPTR36	<RSCAN0_base> + 1244_H
RSCAN0	Transmit buffer data field 0 register 36	RSCAN0TMDF036	<RSCAN0_base> + 1248_H
RSCAN0	Transmit buffer data field 1 register 36	RSCAN0TMDF136	<RSCAN0_base> + $124C_H$
RSCAN0	Transmit buffer ID register 37	RSCAN0TMID37	<RSCAN0_base> + 1250_H
RSCAN0	Transmit buffer pointer register 37	RSCAN0TMPTR37	<RSCAN0_base> + 1254_H
RSCAN0	Transmit buffer data field 0 register 37	RSCAN0TMDF037	<RSCAN0_base> + 1258_H
RSCAN0	Transmit buffer data field 1 register 37	RSCAN0TMDF137	<RSCAN0_base> + $125C_H$
RSCAN0	Transmit buffer ID register 38	RSCAN0TMID38	<RSCAN0_base> + 1260_H
RSCAN0	Transmit buffer pointer register 38	RSCAN0TMPTR38	<RSCAN0_base> + 1264_H
RSCAN0	Transmit buffer data field 0 register 38	RSCAN0TMDF038	<RSCAN0_base> + 1268_H
RSCAN0	Transmit buffer data field 1 register 38	RSCAN0TMDF138	<RSCAN0_base> + $126C_H$
RSCAN0	Transmit buffer ID register 39	RSCAN0TMID39	<RSCAN0_base> + 1270_H
RSCAN0	Transmit buffer pointer register 39	RSCAN0TMPTR39	<RSCAN0_base> + 1274_H
RSCAN0	Transmit buffer data field 0 register 39	RSCAN0TMDF039	<RSCAN0_base> + 1278_H
RSCAN0	Transmit buffer data field 1 register 39	RSCAN0TMDF139	<RSCAN0_base> + $127C_H$
RSCAN0	Transmit buffer ID register 40	RSCAN0TMID40	<RSCAN0_base> + 1280_H
RSCAN0	Transmit buffer pointer register 40	RSCAN0TMPTR40	<RSCAN0_base> + 1284_H
RSCAN0	Transmit buffer data field 0 register 40	RSCAN0TMDF040	<RSCAN0_base> + 1288_H
RSCAN0	Transmit buffer data field 1 register 40	RSCAN0TMDF140	<RSCAN0_base> + $128C_H$
RSCAN0	Transmit buffer ID register 41	RSCAN0TMID41	<RSCAN0_base> + 1290_H
RSCAN0	Transmit buffer pointer register 41	RSCAN0TMPTR41	<RSCAN0_base> + 1294_H
RSCAN0	Transmit buffer data field 0 register 41	RSCAN0TMDF041	<RSCAN0_base> + 1298_H

Table 22.13 Registers (17/19)

Module	Register	Symbol	Address
RSCAN0	Transmit buffer data field 1 register 41	RSCAN0TMDF141	<RSCAN0_base> + $129C_H$
RSCAN0	Transmit buffer ID register 42	RSCAN0TMID42	<RSCAN0_base> + $12A0_H$
RSCAN0	Transmit buffer pointer register 42	RSCAN0TMPTR42	<RSCAN0_base> + $12A4_H$
RSCAN0	Transmit buffer data field 0 register 42	RSCAN0TMDF042	<RSCAN0_base> + $12A8_H$
RSCAN0	Transmit buffer data field 1 register 42	RSCAN0TMDF142	<RSCAN0_base> + $12AC_H$
RSCAN0	Transmit buffer ID register 43	RSCAN0TMID43	<RSCAN0_base> + $12B0_H$
RSCAN0	Transmit buffer pointer register 43	RSCAN0TMPTR43	<RSCAN0_base> + $12B4_H$
RSCAN0	Transmit buffer data field 0 register 43	RSCAN0TMDF043	<RSCAN0_base> + $12B8_H$
RSCAN0	Transmit buffer data field 1 register 43	RSCAN0TMDF143	<RSCAN0_base> + $12BC_H$
RSCAN0	Transmit buffer ID register 44	RSCAN0TMID44	<RSCAN0_base> + $12C0_H$
RSCAN0	Transmit buffer pointer register 44	RSCAN0TMPTR44	<RSCAN0_base> + $12C4_H$
RSCAN0	Transmit buffer data field 0 register 44	RSCAN0TMDF044	<RSCAN0_base> + $12C8_H$
RSCAN0	Transmit buffer data field 1 register 44	RSCAN0TMDF144	<RSCAN0_base> + $12CC_H$
RSCAN0	Transmit buffer ID register 45	RSCAN0TMID45	<RSCAN0_base> + $12D0_H$
RSCAN0	Transmit buffer pointer register 45	RSCAN0TMPTR45	<RSCAN0_base> + $12D4_H$
RSCAN0	Transmit buffer data field 0 register 45	RSCAN0TMDF045	<RSCAN0_base> + $12D8_H$
RSCAN0	Transmit buffer data field 1 register 45	RSCAN0TMDF145	<RSCAN0_base> + $12DC_H$
RSCAN0	Transmit buffer ID register 46	RSCAN0TMID46	<RSCAN0_base> + $12E0_H$
RSCAN0	Transmit buffer pointer register 46	RSCAN0TMPTR46	<RSCAN0_base> + $12E4_H$
RSCAN0	Transmit buffer data field 0 register 46	RSCAN0TMDF046	<RSCAN0_base> + $12E8_H$
RSCAN0	Transmit buffer data field 1 register 46	RSCAN0TMDF146	<RSCAN0_base> + $12EC_H$
RSCAN0	Transmit buffer ID register 47	RSCAN0TMID47	<RSCAN0_base> + $12F0_H$
RSCAN0	Transmit buffer pointer register 47	RSCAN0TMPTR47	<RSCAN0_base> + $12F4_H$
RSCAN0	Transmit buffer data field 0 register 47	RSCAN0TMDF047	<RSCAN0_base> + $12F8_H$
RSCAN0	Transmit buffer data field 1 register 47	RSCAN0TMDF147	<RSCAN0_base> + $12FC_H$
RSCAN0	Transmit history access register 0	RSCAN0THLACC0	<RSCAN0_base> + 1800_H
RSCAN0	Transmit history access register 1	RSCAN0THLACC1	<RSCAN0_base> + 1804_H
RSCAN0	Transmit history access register 2	RSCAN0THLACC2	<RSCAN0_base> + 1808_H
RSCAN0	RAM test page access register 0	RSCAN0RPGACC0	<RSCAN0_base> + 1900_H
RSCAN0	RAM test page access register 1	RSCAN0RPGACC1	<RSCAN0_base> + 1904_H
RSCAN0	RAM test page access register 2	RSCAN0RPGACC2	<RSCAN0_base> + 1908_H
RSCAN0	RAM test page access register 3	RSCAN0RPGACC3	<RSCAN0_base> + $190C_H$
RSCAN0	RAM test page access register 4	RSCAN0RPGACC4	<RSCAN0_base> + 1910_H
RSCAN0	RAM test page access register 5	RSCAN0RPGACC5	<RSCAN0_base> + 1914_H
RSCAN0	RAM test page access register 6	RSCAN0RPGACC6	<RSCAN0_base> + 1918_H
RSCAN0	RAM test page access register 7	RSCAN0RPGACC7	<RSCAN0_base> + $191C_H$
RSCAN0	RAM test page access register 8	RSCAN0RPGACC8	<RSCAN0_base> + 1920_H
RSCAN0	RAM test page access register 9	RSCAN0RPGACC9	<RSCAN0_base> + 1924_H
RSCAN0	RAM test page access register 10	RSCAN0RPGACC10	<RSCAN0_base> + 1928_H
RSCAN0	RAM test page access register 11	RSCAN0RPGACC11	<RSCAN0_base> + $192C_H$
RSCAN0	RAM test page access register 12	RSCAN0RPGACC12	<RSCAN0_base> + 1930_H
RSCAN0	RAM test page access register 13	RSCAN0RPGACC13	<RSCAN0_base> + 1934_H
RSCAN0	RAM test page access register 14	RSCAN0RPGACC14	<RSCAN0_base> + 1938_H
RSCAN0	RAM test page access register 15	RSCAN0RPGACC15	<RSCAN0_base> + $193C_H$

Table 22.13 Registers (18/19)

Module	Register	Symbol	Address
RSCAN0	RAM test page access register 16	RSCAN0RPGACC16	<RSCAN0_base> + 1940_H
RSCAN0	RAM test page access register 17	RSCAN0RPGACC17	<RSCAN0_base> + 1944_H
RSCAN0	RAM test page access register 18	RSCAN0RPGACC18	<RSCAN0_base> + 1948_H
RSCAN0	RAM test page access register 19	RSCAN0RPGACC19	<RSCAN0_base> + $194C_H$
RSCAN0	RAM test page access register 20	RSCAN0RPGACC20	<RSCAN0_base> + 1950_H
RSCAN0	RAM test page access register 21	RSCAN0RPGACC21	<RSCAN0_base> + 1954_H
RSCAN0	RAM test page access register 22	RSCAN0RPGACC22	<RSCAN0_base> + 1958_H
RSCAN0	RAM test page access register 23	RSCAN0RPGACC23	<RSCAN0_base> + $195C_H$
RSCAN0	RAM test page access register 24	RSCAN0RPGACC24	<RSCAN0_base> + 1960_H
RSCAN0	RAM test page access register 25	RSCAN0RPGACC25	<RSCAN0_base> + 1964_H
RSCAN0	RAM test page access register 26	RSCAN0RPGACC26	<RSCAN0_base> + 1968_H
RSCAN0	RAM test page access register 27	RSCAN0RPGACC27	<RSCAN0_base> + $196C_H$
RSCAN0	RAM test page access register 28	RSCAN0RPGACC28	<RSCAN0_base> + 1970_H
RSCAN0	RAM test page access register 29	RSCAN0RPGACC29	<RSCAN0_base> + 1974_H
RSCAN0	RAM test page access register 30	RSCAN0RPGACC30	<RSCAN0_base> + 1978_H
RSCAN0	RAM test page access register 31	RSCAN0RPGACC31	<RSCAN0_base> + $197C_H$
RSCAN0	RAM test page access register 32	RSCAN0RPGACC32	<RSCAN0_base> + 1980_H
RSCAN0	RAM test page access register 33	RSCAN0RPGACC33	<RSCAN0_base> + 1984_H
RSCAN0	RAM test page access register 34	RSCAN0RPGACC34	<RSCAN0_base> + 1988_H
RSCAN0	RAM test page access register 35	RSCAN0RPGACC35	<RSCAN0_base> + $198C_H$
RSCAN0	RAM test page access register 36	RSCAN0RPGACC36	<RSCAN0_base> + 1990_H
RSCAN0	RAM test page access register 37	RSCAN0RPGACC37	<RSCAN0_base> + 1994_H
RSCAN0	RAM test page access register 38	RSCAN0RPGACC38	<RSCAN0_base> + 1998_H
RSCAN0	RAM test page access register 39	RSCAN0RPGACC39	<RSCAN0_base> + $199C_H$
RSCAN0	RAM test page access register 40	RSCAN0RPGACC40	<RSCAN0_base> + $19A0_H$
RSCAN0	RAM test page access register 41	RSCAN0RPGACC41	<RSCAN0_base> + $19A4_H$
RSCAN0	RAM test page access register 42	RSCAN0RPGACC42	<RSCAN0_base> + $19A8_H$
RSCAN0	RAM test page access register 43	RSCAN0RPGACC43	<RSCAN0_base> + $19AC_H$
RSCAN0	RAM test page access register 44	RSCAN0RPGACC44	<RSCAN0_base> + $19B0_H$
RSCAN0	RAM test page access register 45	RSCAN0RPGACC45	<RSCAN0_base> + $19B4_H$
RSCAN0	RAM test page access register 46	RSCAN0RPGACC46	<RSCAN0_base> + $19B8_H$
RSCAN0	RAM test page access register 47	RSCAN0RPGACC47	<RSCAN0_base> + $19BC_H$
RSCAN0	RAM test page access register 48	RSCAN0RPGACC48	<RSCAN0_base> + $19C0_H$
RSCAN0	RAM test page access register 49	RSCAN0RPGACC49	<RSCAN0_base> + $19C4_H$
RSCAN0	RAM test page access register 50	RSCAN0RPGACC50	<RSCAN0_base> + $19C8_H$
RSCAN0	RAM test page access register 51	RSCAN0RPGACC51	<RSCAN0_base> + $19CC_H$
RSCAN0	RAM test page access register 52	RSCAN0RPGACC52	<RSCAN0_base> + $19D0_H$
RSCAN0	RAM test page access register 53	RSCAN0RPGACC53	<RSCAN0_base> + $19D4_H$
RSCAN0	RAM test page access register 54	RSCAN0RPGACC54	<RSCAN0_base> + $19D8_H$
RSCAN0	RAM test page access register 55	RSCAN0RPGACC55	<RSCAN0_base> + $19DC_H$
RSCAN0	RAM test page access register 56	RSCAN0RPGACC56	<RSCAN0_base> + $19E0_H$
RSCAN0	RAM test page access register 57	RSCAN0RPGACC57	<RSCAN0_base> + $19E4_H$
RSCAN0	RAM test page access register 58	RSCAN0RPGACC58	<RSCAN0_base> + $19E8_H$
RSCAN0	RAM test page access register 59	RSCAN0RPGACC59	<RSCAN0_base> + $19EC_H$

Table 22.13 Registers (19/19)

Module	Register	Symbol	Address
RSCAN0	RAM test page access register 60	RSCAN0RPGACC60	<RSCAN0_base> + $19F0_H$
RSCAN0	RAM test page access register 61	RSCAN0RPGACC61	<RSCAN0_base> + $19F4_H$
RSCAN0	RAM test page access register 62	RSCAN0RPGACC62	<RSCAN0_base> + $19F8_H$
RSCAN0	RAM test page access register 63	RSCAN0RPGACC63	<RSCAN0_base> + $19FC_H$

Table 22.14 Transmit Buffer p Allocated to Each Channel

	CANm
Transmit buffer p	Transmit buffer 16 × m + 0
	Transmit buffer 16 × m + 1
	Transmit buffer 16 × m + 2
	Transmit buffer 16 × m + 3
	Transmit buffer 16 × m + 4
	Transmit buffer 16 × m + 5
	Transmit buffer 16 × m + 6
	Transmit buffer 16 × m + 7
	Transmit buffer 16 × m + 8
	Transmit buffer 16 × m + 9
	Transmit buffer 16 × m + 10
	Transmit buffer 16 × m + 11
	Transmit buffer 16 × m + 12
	Transmit buffer 16 × m + 13
	Transmit buffer 16 × m + 14
	Transmit buffer 16 × m + 15

Table 22.15 Transmit/Receive FIFO Buffer k Allocated to Each Channel

	CANm
Transmit/receive FIFO buffer k	Transmit/receive FIFO buffer 3 × m + 0
	Transmit/receive FIFO buffer 3 × m + 1
	Transmit/receive FIFO buffer 3 × m + 2

Table 22.16 Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer by the Setting of Bits CFTML[3:0]

Setting of Bits CFTML[3:0]	Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer
0000_B	Transmit buffer 16 × m + 0
0001_B	Transmit buffer 16 × m + 1
0010_B	Transmit buffer 16 × m + 2
0011_B	Transmit buffer 16 × m + 3
0100_B	Transmit buffer 16 × m + 4
0101_B	Transmit buffer 16 × m + 5
0110_B	Transmit buffer 16 × m + 6
0111_B	Transmit buffer 16 × m + 7
1000_B	Transmit buffer 16 × m + 8
1001_B	Transmit buffer 16 × m + 9
1010_B	Transmit buffer 16 × m + 10
1011_B	Transmit buffer 16 × m + 11
1100_B	Transmit buffer 16 × m + 12
1101_B	Transmit buffer 16 × m + 13
1110_B	Transmit buffer 16 × m + 14
1111_B	Transmit buffer 16 × m + 15

Table 22.17 Transmit Buffer p Allocated to the Transmit Queue of Each Channel

Setting of Bits TXQDC[3:0]	Transmit Buffer p Allocated to the Transmit Queue
0000_B	Setting prohibited
0001_B	Setting prohibited
0010_B	Transmit buffer 16 × m + 15 to 16 × m + 13
0011_B	Transmit buffer 16 × m + 15 to 16 × m + 12
0100_B	Transmit buffer 16 × m + 15 to 16 × m + 11
0101_B	Transmit buffer 16 × m + 15 to 16 × m + 10
0110_B	Transmit buffer 16 × m + 15 to 16 × m + 9
0111_B	Transmit buffer 16 × m + 15 to 16 × m + 8
1000_B	Transmit buffer 16 × m + 15 to 16 × m + 7
1001_B	Transmit buffer 16 × m + 15 to 16 × m + 6
1010_B	Transmit buffer 16 × m + 15 to 16 × m + 5
1011_B	Transmit buffer 16 × m + 15 to 16 × m + 4
1100_B	Transmit buffer 16 × m + 15 to 16 × m + 3
1101_B	Transmit buffer 16 × m + 15 to 16 × m + 2
1110_B	Transmit buffer 16 × m + 15 to 16 × m + 1
1111_B	Transmit buffer 16 × m + 15 to 16 × m + 0

22.3.2 RSCAN0CmCFG — Channel Configuration Register (m = 0 to 2)

Access: RSCAN0CmCFG register can be read/written in 32-bit units
RSCAN0CmCFGL, RSCAN0CmCFGH registers can be read/written in 16-bit units
RSCAN0CmCFGLL, RSCAN0CmCFGLH, RSCAN0CmCFGHL, RSCAN0CmCFGHH registers can be read/written in 8-bit units

Address: RSCAN0CmCFG: <RSCAN0_base> + 0000_H + (10_H × m)

RSCAN0CmCFGL: <RSCAN0_base> + 0000_H + (10_H × m),
RSCAN0CmCFGH: <RSCAN0_base> + 0002_H + (10_H × m)

RSCAN0CmCFGLL: <RSCAN0_base> + 0000_H + (10_H × m),
RSCAN0CmCFGLH: <RSCAN0_base> + 0001_H + (10_H × m),
RSCAN0CmCFGHL: <RSCAN0_base> + 0002_H + (10_H × m),
RSCAN0CmCFGHH: <RSCAN0_base> + 0003_H + (10_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SJW[1:0]		—	TSEG2[2:0]			TSEG1[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	BRP[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.18 RSCAN0CmCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
25, 24	SJW[1:0]	Resynchronization Jump Width Control b25 b24 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq
23	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
22 to 20	TSEG2[2:0]	Time Segment 2 Control b22 b21 b20 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq

Table 22.18 RSCAN0CmCFG Register Contents (2/2)

Bit Position	Bit Name	Function
19 to 16	TSEG1[3:0]	Time Segment 1 Control b19 b18 b17 b16 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq
15 to 10	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
9 to 0	BRP[9:0]	Prescaler Division Ratio Set When these bits are set to P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

Modify the RSCAN0CmCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode before shifting to channel communication mode or channel halt mode. For a description of the bit timing parameters and settings, see Section 22.10.1, Initial Settings.

SJW[1:0] Bits

These bits are used to specify a Tq value for the resynchronization jump width. Allowed values are 1 Tq to 4 Tq, inclusive.
Set a value less than or equal to the value of the TSEG2[2:0] bits.

TSEG2[2:0] Bits

These bits are used to specify a Tq value for the length of phase segment 2 (PHASE_SEG2).
Allowed values are 2 Tq to 8 Tq, inclusive.
Set a value smaller than the value of the TSEG1[3:0] bits.

TSEG1[3:0] Bits

These bits are used to specify a Tq value for the total length of the propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1).
Allowed values are 4 Tq to 16 Tq, inclusive.

BRP[9:0] Bits

The CANmTq clock (fCANTQm) is calculated by dividing the CAN clock (fCAN) by the baud rate prescaler, ((BRP[9:0]) + 1). One clock cycle of the CANmTq clock is 1 Time Quantum (Tq).

22.3.3 RSCAN0CmCTR — Channel Control Register (m = 0 to 2)

Access: RSCAN0CmCTR register can be read/written in 32-bit units
RSCAN0CmCTRL, RSCAN0CmCTRH registers can be read/written in 16-bit units
RSCAN0CmCTRLL, RSCAN0CmCTRLH, RSCAN0CmCTRHL, RSCAN0CmCTRHH registers can be read/written in 8-bit units

Address: RSCAN0CmCTR: <RSCAN0_base> + 0004_H + (10_H × m)

RSCAN0CmCTRL: <RSCAN0_base> + 0004_H + (10_H × m),
RSCAN0CmCTRH: <RSCAN0_base> + 0006_H + (10_H × m)

RSCAN0CmCTRLL: <RSCAN0_base> + 0004_H + (10_H × m),
RSCAN0CmCTRLH: <RSCAN0_base> + 0005_H + (10_H × m),
RSCAN0CmCTRHL: <RSCAN0_base> + 0006_H + (10_H × m),
RSCAN0CmCTRHH: <RSCAN0_base> + 0007_H + (10_H × m)

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	—	—	—	TAIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 22.19 RSCAN0CmCTR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
26, 25	CTMS[1:0]	Communication Test Mode Select b26 b25 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback mode) 1 1: Self-test mode 1 (internal loopback mode)
24	CTME	Communication Test Mode Enable 0: Communication test mode is disabled. 1: Communication test mode is enabled.
23	ERRD	Error Display Mode Select 0: Error flags are displayed only for the first error information after bits 14 to 8 in RSCAN0CmERFL are all cleared. 1: Error flags for all error information are displayed.
22, 21	BOM[1:0]	Bus Off Recovery Mode Select b22 b21 0 0: ISO11898-1 compliant 0 1: Entry to channel halt mode automatically at bus-off entry 1 0: Entry to channel halt mode automatically at bus-off end 1 1: Entry to channel halt mode (in bus-off state) by program request
20 to 17	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
16	TAIE	Transmit Abort Interrupt Enable 0: Transmit abort interrupt is disabled. 1: Transmit abort interrupt is enabled.
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.

Table 22.19 RSCAN0CmCTR Register Contents (2/2)

Bit Position	Bit Name	Function
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.
13	OLIE	Overload Frame Transmit Interrupt Enable 0: Overload frame transmit interrupt is disabled. 1: Overload frame transmit interrupt is enabled.
12	BORIE	Bus Off Recovery Interrupt Enable 0: Bus off recovery interrupt is disabled. 1: Bus off recovery interrupt is enabled.
11	BOEIE	Bus Off Entry Interrupt Enable 0: Bus off entry interrupt is disabled. 1: Bus off entry interrupt is enabled.
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.
7 to 4	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
3	RTBO	Forcible Return from Bus-off When this bit is set to 1, forcible return from the bus off state is made. This bit is always read as 0.
2	CSLPR	Channel Stop Mode 0: Other than channel stop mode 1: Channel stop mode
1, 0	CHMDC[1:0]	Mode Select b1 b0 0 0: Channel communication mode 0 1: Channel reset mode 1 0: Channel halt mode 1 1: Setting prohibited

CTMS[1:0] Bits

These bits are used to select a communication test mode. Modify these bits in channel halt mode only. These bits are set to 0 in channel reset mode.

CTME Bit

Setting this bit to 1 enables communication test mode. Modify these bits in channel halt mode. This bit is set to 0 in channel reset mode.

ERRD Bit

This bit is used to control the display mode of bits 14 to 8 in the RSCAN0CmERFL register.
When this bit is clear to 0, if any error is detected while the flags of bits 14 to 8 in the RSCANnCmERFL register are all 0, only the flags of the first error event are set to 1. If two or more errors occur in the first error event, all the flags of the detected errors are set to 1.

When this bit is set to 1, all the flags of errors that have occurred are set to 1 regardless of the error occurrence order.

Modify this bit only in channel reset mode or channel halt mode.

BOM[1:0] Bits

These bits are used to select the bus off recovery mode of the RSCAN module.

When the BOM[1:0] bits are set to 00_B, return from the bus off state to the error active state is compliant with the CAN specifications. That is, the RSCAN module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CHMDC[1:0] bits are set to 10_B (channel halt mode) before recessive bits are detected 128 times, the RSCAN module does not transition to channel halt mode until recessive bits are detected 128 times.

When the RSCAN module reaches the bus off state when the BOM[1:0] bits are set to 01_B, the CHMDC[1:0] bits in the RSCAN0CmCTR register (m = 0 to 2) are set to 10_B and the RSCAN module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits in the RSCAN0CmSTS register are cleared to 00_H.

When the RSCAN module reaches the bus off state when the BOM[1:0] bits are set to 10_B, the CHMDC[1:0] bits are set to 10_B and the RSCAN module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H.

When the BOM[1:0] bits are set to 11_B and the CHMDC[1:0] bits are set to 10_B while the RSCAN module is in the bus off state, the RSCAN module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H. However, if 11 consecutive recessive bits are detected 128 times and the RSCAN module has recovered to the error active state from the bus off state before the CHMDC[1:0] bits are set to 10_B, a bus off recovery interrupt request is generated.

If a program writes to the CHMDC[1:0] bit at the same time as the RSCAN module transition to channel halt mode (at bus off entry when the BOM[1:0] bits are 01_B or at bus off end when the BOM[1:0] bits are 10_B), the program's writing takes precedence. Modify the BOM[1:0] bits only in channel reset mode.

TAIE Bit

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

ALIE Bit

When the ALF flag in the RSCAN0CmERFL register is set to 1 with the ALIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BLIE Bit

When the BLF flag in the RSCAN0CmERFL register is set to 1 with the BLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

OLIE Bit

When the OVLF flag in the RSCAN0CmERFL register is set to 1 with the OLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BORIE Bit

When the BORF flag in the RSCAN0CmERFL register is set to 1 with the BORIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BOEIE Bit

When the BOEF flag in the RSCAN0CmERFL register is set to 1 with the BOEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EPIE Bit

When the EPF flag in the RSCAN0CmERFL register is set to 1 with the EPIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EWIE Bit

When the EWF flag in the RSCAN0CmERFL register is set to 1 with the EWIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BEIE Bit

When the BEF flag in the RSCAN0CmERFL register is set to 1 with the BEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

RTBO Bit

Setting this bit to 1 in the bus off state forcibly changes the state from the bus off state to the error active state. This bit is automatically cleared to 0. Setting this bit to 1 clears the TEC[7:0] and REC[7:0] bits in the RSCAN0CmSTS register to 00_H and also clears the BOSTS flag in the RSCAN0CmSTS register to 0 (not in bus off state). The other registers remain unchanged. No bus off recovery interrupt request is generated upon return from the bus off state in this case. Use this bit only when the BOM[1:0] bits in the RSCAN0CmCTR register are 00_B (ISO11898-1 compliant).

A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the RSCAN module transitions to the error active state. Set this bit to 1 in channel communication mode.

CSLPR Bit

Setting this bit to 1 places the channel into channel stop mode.
Clearing this bit to 0 makes the channel exit channel stop mode.
This bit should not be modified in channel communication mode or channel halt mode.

CHMDC[1:0] Bits

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see Section 22.5.2, Channel Modes. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to 11_B. When the CAN module has automatically transitioned to channel halt mode based on the setting of the BOM[1:0] bits, the CHMDC[1:0] bits automatically become 10_B.

22.3.4 RSCAN0CmSTS — Channel Status Register (m = 0 to 2)

Access: RSCAN0CmSTS register can be read only in 32-bit units
RSCAN0CmSTSL, RSCAN0CmSTSH registers can be read only in 16-bit units
RSCAN0CmSTSLL, RSCAN0CmSTSHL, RSCAN0CmSTSHH registers can be read only in 8-bit units

Address: RSCAN0CmSTS: <RSCAN0_base> + 0008_H + (10_H × m)

RSCAN0CmSTSL: <RSCAN0_base> + 0008_H + (10_H × m),
RSCAN0CmSTSH: <RSCAN0_base> + $000A_H$ + (10_H × m)

RSCAN0CmSTSLL: <RSCAN0_base> + 0008_H + (10_H × m),
RSCAN0CmSTSHL: <RSCAN0_base> + $000A_H$ + (10_H × m),
RSCAN0CmSTSHH: <RSCAN0_base> + $000B_H$ + (10_H × m)

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEC[7:0]								REC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.20 RSCAN0CmSTS Register Contents

Bit Position	Bit Name	Function
31 to 24	TEC[7:0]	The transmit error counter (TEC) can be read.
23 to 16	REC[7:0]	The receive error counter (REC) can be read.
15 to 8	Reserved	These bits are read as the value after reset.
7	COMSTS	Communication Status Flag 0: Communication is not ready. 1: Communication is ready.
6	RECSTS	Receive Status Flag 0: Bus idle, in transmission or bus off state 1: In reception
5	TRMSTS	Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus off state
4	BOSTS	Bus Off Status Flag 0: Not in bus off state 1: In bus off state
3	EPSTS	Error Passive Status Flag 0: Not in error passive state 1: In error passive state
2	CSLPSTS	Channel Stop Status Flag 0: Not in channel stop mode 1: In channel stop mode
1	CHLTSTS	Channel Halt Status Flag 0: Not in channel halt mode 1: In channel halt mode
0	CRSTSTS	Channel Reset Status Flag 0: Not in channel reset mode 1: In channel reset mode

TEC[7:0] Bits

These bits contain the transmit error counter value. For transmit error counter increment/decrement conditions, see the CAN specification (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

REC[7:0] Bits

These bits contain the receive error counter value. For receive error counter increment/decrement conditions, see the CAN specifications (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

COMSTS Flag

This bit indicates that communication is ready.

This flag becomes 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag is cleared to 0 in channel reset mode or channel halt mode.

RECSTS Flag

This flag is set to 1 when reception has started, and is cleared to 0 when the bus has become idle or transmission has started.

TRMSTS Flag

This flag is set to 1 when transmission has started, and is cleared to 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

BOSTS Flag

This flag is set to 1 when the bus off state (TEC[7:0] > 255) is entered. It is cleared to 0 when the CAN module has exited the bus off state.

EPSTS Flag

This flag is set to 1 when the RSCAN module has entered the error passive state ((128 ≤ TEC[7:0] ≤ 255) or (128 ≤ REC[7:0])), It is cleared to 0 when the RSCAN module has exited the error passive state or has entered channel reset mode.

CSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel stop mode, and is cleared to 0 when the CAN module has returned from channel stop mode.

CHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel halt mode, and is cleared to 0 when the CAN module has returned from channel halt mode.

CRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel reset mode, and is cleared to 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 when the CAN module transitions from channel reset mode to channel stop mode.

22.3.5 RSCAN0CmERFL — Channel Error Flag Register (m = 0 to 2)

Access: RSCAN0CmERFL register can be read/written in 32-bit units
RSCAN0CmERFLL register can be read/written in 16-bit units
RSCANnCmERFLH register is a read-only register that can be read in 16-bit units.
RSCANnCmERFLLL, RSCANnCmERFLLH registers can be read or written in 8-bit units.
RSCANnCmERFLHL, RSCANnCmERFLHH registers are the read-only registers that can be read in 8-bit units.

Address: RSCAN0CmERFL: <RSCAN0_base> + $000C_H$ + (10_H × m)

RSCAN0CmERFLL: <RSCAN0_base> + $000C_H$ + (10_H × m),
RSCAN0CmERFLH: <RSCAN0_base> + $000E_H$ + (10_H × m)

RSCAN0CmERFLLL: <RSCAN0_base> + $000C_H$ + (10_H × m),
RSCAN0CmERFLLH: <RSCAN0_base> + $000D_H$ + (10_H × m),
RSCAN0CmERFLHL: <RSCAN0_base> + $000E_H$ + (10_H × m),
RSCAN0CmERFLHH: <RSCAN0_base> + $000F_H$ + (10_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CRCREG[14:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 22.21 RSCAN0CmERFL Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
30 to 16	CRCREG[14:0]	CRC Calculation Data A CRC value calculated based on the transmit message or receive message is indicated.
15	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
14	ADERR	ACK Delimiter Error Flag 0: No ACK delimiter error is detected. 1: ACK delimiter error is detected.
13	B0ERR	Dominant Bit Error Flag 0: No dominant bit error is detected. 1: Dominant bit error is detected.
12	B1ERR	Recessive Bit Error Flag 0: No recessive bit error is detected. 1: Recessive bit error is detected.
11	CERR	CRC Error Flag 0: No CRC error is detected. 1: CRC error is detected.
10	AERR	ACK Error Flag 0: No ACK error is detected. 1: ACK error is detected.

Table 22.21 RSCAN0CmERFL Register Contents (2/2)

Bit Position	Bit Name	Function
9	FERR	Form Error Flag 0: No form error is detected. 1: Form error is detected.
8	SERR	Stuff Error Flag 0: No stuff error is detected. 1: Stuff error is detected.
7	ALF	Arbitration-lost Flag 0: No arbitration-lost is detected. 1: Arbitration-lost is detected.
6	BLF	Bus Lock Flag 0: No channel bus lock is detected. 1: Channel bus lock is detected.
5	OVLF	Overload Flag 0: No overload is detected. 1: Overload is detected.
4	BORF	Bus Off Recovery Flag 0: No bus off recovery is detected. 1: Bus off recovery is detected.
3	BOEF	Bus Off Entry Flag 0: No bus off entry is detected. 1: Bus off entry is detected.
2	EPF	Error Passive Flag 0: No error passive is detected. 1: Error passive is detected.
1	EWF	Error Warning Flag 0: No error warning is detected. 1: Error warning is detected.
0	BEF	Bus Error Flag 0: No channel bus error is detected. 1: Channel bus error is detected.

See the CAN specification (ISO11898-1) for a description of error occurrence conditions. To clear each flag of this register, the program must write a 0 to the corresponding bit. These flags cannot be set to 1 by the program. If any of these flags is set to 1 at the same time that the program writes 0 to the flag, the flag is still set to 1. The channel reset mode transition clears all of these flags to 0.

If the ERRD bit in the RSCAN0CmCTR register is set to 0 (ie, only the flags for the first error event are displayed) and an error related to bits 14 to 8 of RSCAN0CmERFL is detected, the flag bits are only set to 1 by the error event if bits 14 to 8 were all 0 at the time when the error occurred.

CRCREG[14:0] Flag

When the CTME bit in the RSCAN0CmCTR register is set to 1 (communication test mode is enabled), the CRC value calculated based on the transmit or receive message can be read. When the CTME bit is set to 0 (communication test mode is disabled), these bits are always read as 0.
This bit is always 0 in channel reset mode.

ADERR Flag

This flag is set to 1 when a form error has been detected in the ACK delimiter during transmission.

B0ERR Flag

This flag is set to 1 when a recessive bit has been detected though a dominant bit was transmitted.

B1ERR Flag

This flag is set to 1 when a dominant bit has been detected though a recessive bit was transmitted.

CERR Flag

This flag is set to 1 when a CRC error has been detected.

AERR Flag

This flag is set to 1 when an ACK error has been detected.

FERR Flag

This flag is set to 1 when a form error has been detected.

SERR Flag

This flag is set to 1 when a stuff error has been detected.

ALF Flag

This flag is set to 1 when an arbitration-lost has been detected.

BLF Flag

This flag is set to 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, detection of a dominant lock is restarted when either of the following conditions is met.

- A recessive bit is detected after the BLF bit has been cleared from 1 to 0.
- The CAN module transitions to channel reset mode and returns to channel communication mode after the BLF bit has been cleared from 1 to 0.

OVLF Flag

This flag is set to 1 when the overload frame transmit condition has been detected when performing reception or transmission.

BORF Flag

This flag is set to 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus off state. However, this flag is not set to 1 if the CAN module returns from the bus off state in any of the following ways before 11 consecutive recessive bits are detected 128 times.

- The CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 01_B (channel reset mode).
- The RTBO bit in the RSCAN0CmCTR register is set to 1 (forcible return from the bus off state is made).
- The BOM[1:0] bits in the RSCAN0CmCTR register are set to 01_B (transition to channel halt mode at bus off entry).
- The CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 10_B (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the BOM[1:0] bits set to 11_B (transition to channel halt mode upon a request from the program during bus off).

BOEF Flag

This flag is set to 1 when the bus off state is entered (TEC[7:0] value > 255). This flag is also set to 1 if the bus off state is entered when the BOM[1:0] bits in the RSCAN0CmCTR register (m = 0 to 2) set to 01_B (transition to channel halt mode at bus off entry).

EPF Flag

This flag becomes 1 when the error passive state is reached ((128 ≤ TEC[7:0] ≤ 255) or (128 ≤ REC[7:0])).
This flag becomes 1 only when the REC[7:0] or TEC[7:0] value first exceeds 127. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 127, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 127 or less and then the REC[7:0] or TEC[7:0] value exceeds 127 again.

EWF Flag

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 95. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 95, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 95 or less and then the REC[7:0] or TEC[7:0] value exceeds 95 again.

BEF Flag

This flag is set to 1 when any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags in the RSCAN0CmERFL register is set to 1.

NOTE

To clear the flag of this register to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

22.3.6 RSCAN0GCFG — Global Configuration Register

Access: RSCAN0GCFG register can be read/written in 32-bit units
RSCAN0GCFGL, RSCAN0GCFGH registers can be read/written in 16-bit units
RSCAN0GCFGLL, RSCAN0GCFGLH, RSCAN0GCFGHL, RSCAN0GCFGHH registers can be read/written in 8-bit units

Address: RSCAN0GCFG: <RSCAN0_base> + 0084_H

RSCAN0GCFGL: <RSCAN0_base> + 0084_H,
RSCAN0GCFGH: <RSCAN0_base> + 0086_H

RSCAN0GCFGLL: <RSCAN0_base> + 0084_H,
RSCAN0GCFGLH: <RSCAN0_base> + 0085_H,
RSCAN0GCFGHL: <RSCAN0_base> + 0086_H,
RSCAN0GCFGHH: <RSCAN0_base> + 0087_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITRCP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSBTCS[2:0]			TSSS	TSP[3:0]				—	—	—	DCS	MME	DRE	DCE	TPRI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 22.22 RSCAN0GCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	ITRCP[15:0]	Interval Timer Prescaler Set When these bits are set to M, the pclk is divided by M. Setting 0000_H is prohibited when the interval timer is in use.
15 to 13	TSBTCS[2:0]	Timestamp Clock Source Select b15 b14 b13 0 0 0: Channel 0 bit time clock 0 0 1: Channel 1 bit time clock 0 1 0: Channel 2 bit time clock 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited
12	TSSS	Timestamp Source Select 0: pclk/2*1 1: Bit time clock

Table 22.22 RSCAN0GCFG Register Contents (2/2)

Bit Position	Bit Name	Function
11 to 8	TSP[3:0]	Timestamp Clock Source Division b11 b10 b9 b8 0 0 0 0: Not divided 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 256 1 0 0 1: Divided by 512 1 0 1 0: Divided by 1024 1 0 1 1: Divided by 2048 1 1 0 0: Divided by 4096 1 1 0 1: Divided by 8192 1 1 1 0: Divided by 16384 1 1 1 1: Divided by 32768
7 to 5	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
4	DCS	CAN Clock Source Select*2 0: clkc 1: clk_xincan
3	MME	Mirror Function Enable 0: Mirror function is disabled. 1: Mirror function is enabled.
2	DRE	DLC Replacement Enable 0: DLC replacement is disabled. 1: DLC replacement is enabled.
1	DCE	DLC Check Enable 0: DLC check is disabled. 1: DLC check is enabled.
0	TPRI	Transmit Priority Select 0: ID priority 1: Transmit buffer number priority

Note 1. When specifying pclk/2 as the timestamp counter count source, set bits TSBTCS[2:0] to 000_B.

Note 2. For the CAN clock frequency settings, see Table 22.8, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/D1L/D1M.

Modify the RSCAN0GCFG register only in global reset mode.

ITRCP[15:0] Bits

These bits are used to set a clock source division value of the interval timer for FIFO buffers. See Section 22.7.3.1, Interval Transmission Function.

TSBTCS[2:0] Bits

When the TSSS bit is 1, these bits are used to select the channel of the bit time clock that will be the clock source of the timestamp counter.

TSSS Bit

This bit is used to select a clock source of the timestamp counter.

TSP[3:0] Bits

A clock obtained by dividing the clock source selected with the TSBTCS[2:0] bits and TSSS bit according to the TSP[3:0] bits is used as the timestamp counter count source.

DCS Bit

When this bit is set to 0, clkc is used as the clock source of the CAN clock (fCAN).

When this bit is set to 1, clk_xincan is used as the clock source of the CAN clock (fCAN).

For the CAN clock frequency settings, see Table 22.8, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/D1L/D1M.

MME Bit

Setting this bit to 1 makes the mirror function available.

DRE Bit

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of 00_H is stored in each data byte beyond the DLC value of the receive rule.

The DLC replacement function is only available when the DCE bit is set to 1 (DLC check is enabled).

DCE Bit

Setting this bit to 1 makes the DLC check function available. When disabling the DLC check function, set the GAFLDLC[3:0] bits in the RSCAN0GAFLP0j register to 0000_B before clearing the DCE bit in the RSCAN0GCFG register to 0.

TPRI Bit

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO11898-1 specifications). When this bit is set to 1, transmit buffer number priority is selected and the transmit buffer with the smallest number of those has the highest priority.

While the transmit queue is in use, this bit should be set to 0.

22.3.7 RSCAN0GCTR — Global Control Register

Access: RSCAN0GCTR register can be read/written in 32-bit units
RSCAN0GCTRL, RSCAN0GCTRH registers can be read/written in 16-bit units
RSCAN0GCTRLL, RSCAN0GCTRLH, RSCAN0GCTRHL registers can be read/written in 8-bit units

Address: RSCAN0GCTR: <RSCAN0_base> + 0088_H

RSCAN0GCTRL: <RSCAN0_base> + 0088_H,
RSCAN0GCTRH: <RSCAN0_base> + $008A_H$

RSCAN0GCTRLL: <RSCAN0_base> + 0088_H,
RSCAN0GCTRLH: <RSCAN0_base> + 0089_H,
RSCAN0GCTRHL: <RSCAN0_base> + $008A_H$

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 22.23 RSCAN0GCTR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
16	TSRST	Timestamp Counter Reset Setting the TSRST bit to 1 resets the timestamp counter. This bit is always read as 0.
15 to 11	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
10	THLEIE	Transmit History Buffer Overflow Interrupt Enable 0: Transmit history buffer overflow interrupt is disabled. 1: Transmit history buffer overflow interrupt is enabled.
9	MEIE	FIFO Message Lost Interrupt Enable 0: FIFO message lost interrupt is disabled. 1: FIFO message lost interrupt is enabled.
8	DEIE	DLC Error Interrupt Enable 0: DLC error interrupt is disabled. 1: DLC error interrupt is enabled.
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	GSLPR	Global Stop Mode 0: Other than global stop mode 1: Global stop mode
1, 0	GMDC[1:0]	Global Mode Select b1 b0 0 0: Global operating mode 0 1: Global reset mode 1 0: Global test mode 1 1: Setting prohibited

TSRST Bit

This bit is used to reset the timestamp counter. When this bit is set to 1, the RSCAN0GTSC register is cleared to 0000_H.

THLEIE Bit

When the THLEIE bit is set to 1 and the THLES flag in the RSCAN0GERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

MEIE Bit

When the MEIE bit is set to 1 and the MES flag in the RSCAN0GERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

DEIE Bit

When the DEIE bit is set to 1 and the DEF flag in the RSCAN0GERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

GSLPR Bit

Setting this bit to 1 places the RSCAN module into global stop mode.
Clearing this bit to 0 makes the RSCAN module leave from global stop mode.
This bit should not be modified in global operating mode or global test mode.

GMDC[1:0] Bits

These bits are used to select the mode of entire RSCAN module (global operating mode, global reset mode, or global test mode). For details, see Section 22.5.1, Global Modes. Setting the GSLPR bit to 1 when in global reset mode places the RSCAN module into global stop mode.

22.3.8 RSCAN0GSTS — Global Status Register

Access: RSCAN0GSTS register can be read only in 32-bit units
RSCAN0GSTSL register is a read-only register that can be read only in 16-bit units
RSCAN0GSTSLL register is a read-only register that can be read only in 8-bit units

Address: RSCAN0GSTS: <RSCAN0_base> + $008C_H$

RSCAN0GSTSL: <RSCAN0_base> + $008C_H$

RSCAN0GSTSLL: <RSCAN0_base> + $008C_H$

Value after reset: $0000\ 000D_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	GRAMINIT	GSLPSTS	GHLTSTS	GRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.24 RSCAN0GSTS Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	GRAMINIT	CAN RAM Initialization Status Flag 0: CAN RAM initialization is completed. 1: CAN RAM initialization is ongoing.
2	GSLPSTS	Global Stop Status Flag 0: Not in global stop mode 1: In global stop mode
1	GHLTSTS	Global Test Status Flag 0: Not in global test mode 1: In global test mode
0	GRSTSTS	Global Reset Status Flag 0: Not in global reset mode 1: In global reset mode

GRAMINIT Flag

This flag indicates the initialization status of the CAN RAM.

This flag is set to 1 after the MCU has been reset, and is cleared to 0 when CAN RAM initialisation is completed.

GSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to global stop mode, and is cleared to 0 when the CAN module has returned from global stop mode.

GHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global test mode, and is cleared to 0 when the CAN module has exited global test mode.

GRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global reset mode, and is cleared to 0 when the CAN module has exited global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

22.3.9 RSCAN0GERFL — Global Error Flag Register

Access: RSCAN0GERFL register can be read/written in 32-bit units
RSCAN0GERFLL register can be read/written in 16-bit units
RSCAN0GERFLLL register can be read/written in 8-bit units

Address: RSCAN0GERFL: <RSCAN0_base> + 0090_H

RSCAN0GERFLL: <RSCAN0_base> + 0090_H

RSCAN0GERFLLL: <RSCAN0_base> + 0090_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	THLES	MES	DEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 22.25 RSCAN0GERFL Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing these bits, write the value after reset.
13 to 8	Reserved	When read, the undefined value is returned. When writing these bits, write the value after reset.
7, 6	Reserved	When read, the value after reset is returned. When writing these bits, write the value after reset.
5	Reserved	When read, the undefined value is returned. When writing these bits, write the value after reset.
4, 3	Reserved	When read, the value after reset is returned. When writing these bits, write the value after reset.
2	THLES	Transmit History Buffer Overflow Status Flag 0: No transmit history buffer overflow has occurred. 1: A transmit history buffer overflow has occurred.
1	MES	FIFO Message Lost Status Flag 0: No FIFO message lost error has occurred. 1: A FIFO message lost error has occurred.
0	DEF	DLC Error Flag 0: No DLC error has occurred. 1: A DLC error has occurred.

All flags in the RSCAN0GERFL register are cleared to 0 in global reset mode.

THLES Flag

The THLES flag is set to 1 when any one of the THLELT flags in the RSCAN0THLSTSm register (m = 0 to 2) is set to 1.

This flag is cleared to 0 when the THLELT flags of all channels are set to 0.

MES Flag

The MES flag is set to 1 when any one of the RFMLT flags in the RSCAN0RFSTSx register (x = 0 to 7) or the CFMLT flags in the RSCAN0CFSTSk register (k = 0 to 8) is set to 1.

This flag is cleared to 0 when all RFMLT flags and CFMLT flags are set to 0.

DEF Flag

The DEF flag is set to 1 when an error has been detected during the DLC check. The program can clear this flag by writing 0 to this bit.

NOTE

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

22.3.10 RSCAN0GTINTSTS0 — Global TX Interrupt Status Register 0

Access: RSCAN0GTINTSTS0 register can be read only in 32-bit units
RSCAN0GTINTSTS0L, RSCAN0GTINTSTS0H registers can be read only in 16-bit units
RSCAN0GTINTSTS0LL, RSCAN0GTINTSTS0LH, RSCAN0GTINTSTS0HL registers can be read only in 8-bit units

Address: RSCAN0GTINTSTS0: <RSCAN0_base> + 0460_H

RSCAN0GTINTSTS0L: <RSCAN0_base> + 0460_H,
RSCAN0GTINTSTS0H: <RSCAN0_base> + 0462_H

RSCAN0GTINTSTS0LL: <RSCAN0_base> + 0460_H,
RSCAN0GTINTSTS0LH: <RSCAN0_base> + 0461_H,
RSCAN0GTINTSTS0HL: <RSCAN0_base> + 0462_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	THIF2	CFTIF2	TQIF2	TAIF2	TSIF2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R*1	R*1	R*1	R*1	R*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	—	—	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R*1	R*1	R*1	R*1	R*1	R	R	R	R*1	R*1	R*1	R*1	R*1

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 22.26 RSCAN0GTINTSTS0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 21	Reserved	These bits are read as the value after reset.
20	THIF2	Channel 2 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
19	CFTIF2	Channel 2 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
18	TQIF2	Channel 2 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
17	TAIF2	Channel 2 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
16	TSIF2	Channel 2 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
15 to 13	Reserved	These bits are read as the value after reset.
12	THIF1	Channel 1 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
11	CFTIF1	Channel 1 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
10	TQIF1	Channel 1 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.

Table 22.26 RSCAN0GTINTSTS0 Register Contents (2/2)

Bit Position	Bit Name	Function
9	TAIF1	Channel 1 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
8	TSIF1	Channel 1 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
7 to 5	Reserved	These bits are read as the value after reset.
4	THIF0	Channel 0 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF0	Channel 0 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF0	Channel 0 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF0	Channel 0 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
0	TSIF0	Channel 0 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

TSIFm Bits

The TSIFm bit is set to 1 when the TMIEp bit in the RSCAN0TMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to 10_B (transmit completed without abort request) or 11_B (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to 00_B under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIE bit to 0 also clears this flag to 0.

TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RSCAN0CmCTR register is 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RSCAN0TMSTSp register are set to 01_B (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to 00_B after the transmit abort is completed.

TQIFm Bits

When the TXQIE bit in the RSCAN0TXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQIF bit in the RSCAN0TXQSTSm register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQIF bit (transmit queue interrupt request) in the RSCAN0TXQSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the TXQIE bit is cleared to 0.

CFTIFm Bits

When the CFTXIE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RSCAN0CFSTSk register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

THIFm Bits

When the THLIE bit in the RSCAN0THLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RSCAN0THLSTSm register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RSCAN0THLSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the THLIE bit is cleared to 0.

22.3.11 RSCAN0GTSC — Global Timestamp Counter Register

Access: RSCAN0GTSC register is a read-only register that can be read in 32-bit units.
RSCAN0GTSCL register is a read-only register that can be read only in 16-bit units.

Address: RSCAN0GTSC: <RSCAN0_base> + 0094_H

RSCAN0GTSCL: <RSCAN0_base> + 0094_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.27 RSCAN0GTSC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset.
15 to 0	TS[15:0]	Timestamp Value The timestamp counter value can be read. Counter Value: 0000_H to $FFFF_H$

TS[15:0] Bits

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. When the SOF is detected, the TS[15:0] value is captured and later stored in the receive buffer or the FIFO buffer. The timestamp counter is initialized in global reset mode.

The timestamp counter starts and stops counting differently, depending on the count source.

- When the TSSS bit in the RSCAN0GCFG register is 0 (pclk):
  The timestamp counter starts counting when the RSCAN module has transitioned to global operating mode.
  This counter stops counting when the RSCAN module has transitioned to global stop mode or global test mode.
- When the TSSS bit is 1 (CANm bit time clock):
  The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.
  This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

22.3.12 RSCAN0GAFLECTR — Receive Rule Entry Control Register

Access: RSCAN0GAFLECTR register can be read or written in 32-bit units
RSCAN0GAFLECTRL register can be read/written in 16-bit units
RSCAN0GAFLECTRLL, RSCAN0GAFLECTRLH registers can be read or written in 8-bit units

Address: RSCAN0GAFLECTR: <RSCAN0_base> + 0098_H

RSCAN0GAFLECTRL: <RSCAN0_base> + 0098_H

RSCAN0GAFLECTRLL: <RSCAN0_base> + 0098_H,
RSCAN0GAFLECTRLH: <RSCAN0_base> + 0099_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	AFLDAE	—	—	—	AFLPN[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 22.28 RSCAN0GAFLECTR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
8	AFLDAE	Receive Rule Table Write Enable 0: Receive rule table write is disabled. 1: Receive rule table write is enabled.
7 to 5	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
4 to 0	AFLPN[4:0]	Receive Rule Table Page Number Configuration A page number can be selected from a range of page 0 (00000_B) to page 11 (01011_B).

AFLDAE Bit

Setting this bit to 0 disables the write to the receive rule table. After writes to the receive rule table are completed, set this bit to 0 to disable the write to the table. The receive rule table can be read regardless of the value of this bit.

Set the AFLDAE bit to 1 only in global reset mode.

AFLPN[4:0] Bits

These bits are used to set the page number of the receive rule table. Sixteen receive rules can be set per page.

Set these bits to a value within the range of 00000_B to 01011_B.

22.3.13 RSCAN0GAFLCFG0 — Receive Rule Configuration Register 0

Access: RSCAN0GAFLCFG0 register can be read/written in 32-bit units
RSCAN0GAFLCFG0L, RSCAN0GAFLCFG0H registers can be read/written in 16-bit units
RSCAN0GAFLCFG0LH, RSCAN0GAFLCFG0HL, RSCAN0GAFLCFG0HH registers can be read/written in 8-bit units

Address: RSCAN0GAFLCFG0: <RSCAN0_base> + $009C_H$

RSCAN0GAFLCFG0L: <RSCAN0_base> + $009C_H$,
RSCAN0GAFLCFG0H: <RSCAN0_base> + $009E_H$

RSCAN0GAFLCFG0LH: <RSCAN0_base> + $009D_H$,
RSCAN0GAFLCFG0HL: <RSCAN0_base> + $009E_H$,
RSCAN0GAFLCFG0HH: <RSCAN0_base> + $009F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC0[7:0]								RNC1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RNC2[7:0]								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.29 RSCAN0GAFLCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC0[7:0]	Number of Rules for Channel 0 Set the number of receive rules exclusively used for channel 0.
23 to 16	RNC1[7:0]	Number of Rules for Channel 1 Set the number of receive rules exclusively used for channel 1.
15 to 8	RNC2[7:0]	Number of Rules for Channel 2 Set the number of receive rules exclusively used for channel 2.
7 to 0	Reserved	These bits are read as the value after reset. The write value should be the value after reset.

Modify the RSCAN0GAFLCFG0 register only in global reset mode.

Up to 64 × (number of channels) rules can be registered in the receive rule table for the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total of the number of rules allocated to each channel is not larger than the number of rules that can be registered for the entire unit.

RNC0[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 0 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC1[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 1 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC2[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 2 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

22.3.14 RSCAN0GAFLIDj — Receive Rule ID Register (j = 0 to 15)

Access: RSCAN0GAFLIDj register can be read/written in 32-bit units
RSCAN0GAFLIDjL, RSCAN0GAFLIDjH registers can be read/written in 16-bit units
RSCAN0GAFLIDjLL, RSCAN0GAFLIDjLH, RSCAN0GAFLIDjHL, RSCAN0GAFLIDjHH registers can be read/written in 8-bit units

Address: RSCAN0GAFLIDj: <RSCAN0_base> + 0500_H + (10_H × j)

RSCAN0GAFLIDjL: <RSCAN0_base> + 0500_H + (10_H × j),
RSCAN0GAFLIDjH: <RSCAN0_base> + 0502_H + (10_H × j)

RSCAN0GAFLIDjLL: <RSCAN0_base> + 0500_H + (10_H × j),
RSCAN0GAFLIDjLH: <RSCAN0_base> + 0501_H + (10_H × j),
RSCAN0GAFLIDjHL: <RSCAN0_base> + 0502_H + (10_H × j),
RSCAN0GAFLIDjHH: <RSCAN0_base> + 0503_H + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.30 RSCAN0GAFLIDj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDE	IDE Select 0: Standard ID 1: Extended ID
30	GAFLRTR	RTR Select 0: Data frame 1: Remote frame
29	GAFLLB	Receive Rule Target Message Select 0: When a message transmitted from another CAN node is received 1: When the own transmitted message is received
28 to 0	GAFLID[28:0]	ID Set the ID of the receive rule. For the standard ID, set the ID in bits b10 to b0 and set bits b28 to b11 to 0.

Modify the RSCAN0GAFLIDj register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDE Bit

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

GAFLRTR Bit

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

GAFLLB Bit

When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.

When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when the CAN node is receiving its own transmitted messages.

GAFLID[28:0] Bits

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID of the received message during the acceptance filter processing.

22.3.15 RSCAN0GAFLMj — Receive Rule Mask Register (j = 0 to 15)

Access: RSCAN0GAFLMj register can be read/written in 32-bit units
RSCAN0GAFLMjL, RSCAN0GAFLMjH registers can be read/written in 16-bit units
RSCAN0GAFLMjLL, RSCAN0GAFLMjLH, RSCAN0GAFLMjHL, RSCAN0GAFLMjHH registers can be read/written in 8-bit units

Address: RSCAN0GAFLMj: <RSCAN0_base> + 0504_H + (10_H × j)

RSCAN0GAFLMjL: <RSCAN0_base> + 0504_H + (10_H × j),
RSCAN0GAFLMjH: <RSCAN0_base> + 0506_H + (10_H × j)

RSCAN0GAFLMjLL: <RSCAN0_base> + 0504_H + (10_H × j),
RSCAN0GAFLMjLH: <RSCAN0_base> + 0505_H + (10_H × j),
RSCAN0GAFLMjHL: <RSCAN0_base> + 0506_H + (10_H × j),
RSCAN0GAFLMjHH: <RSCAN0_base> + 0507_H + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDEM	GAFLRTRM	—	GAFLIDM[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLIDM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.31 RSCAN0GAFLMj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDEM	IDE Mask 0: The IDE bit is not compared. 1: The IDE bit is compared.
30	GAFLRTRM	RTR Mask 0: The RTR bit is not compared. 1: The RTR bit is compared
29	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
28 to 0	GAFLIDM[28:0]	ID Mask 0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.

Modify the RSCAN0GAFLMj register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDEM Bit

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDE bit in the RSCAN0GAFLIDj register.

When this bit is cleared to 0, the IDs of all the receive messages and the specified IDs are regarded as matched. To set the GAFLIDEM bit to 0, set the GAFLIDM[28:0] bits to all 0 at the same time.

GAFLRTRM Bit

This bit is used to mask the RTR bit of the receive rule.

GAFLIDM[28:0] Bits

These bits are used to mask the corresponding ID bit of the receive rule.

22.3.16 RSCAN0GAFLP0j — Receive Rule Pointer 0 Register (j = 0 to 15)

Access: RSCAN0GAFLP0j register can be read/written in 32-bit units
RSCAN0GAFLP0jL, RSCAN0GAFLP0jH registers can be read/written in 16-bit units
RSCAN0GAFLP0jLH, RSCAN0GAFLP0jHL, RSCAN0GAFLP0jHH registers can be read/written in 8-bit units

Address: RSCAN0GAFLP0j: <RSCAN0_base> + 0508_H + (10_H × j)

RSCAN0GAFLP0jL: <RSCAN0_base> + 0508_H + (10_H × j),
RSCAN0GAFLP0jH: <RSCAN0_base> + $050A_H$ + (10_H × j)

RSCAN0GAFLP0jLH: <RSCAN0_base> + 0509_H + (10_H × j),
RSCAN0GAFLP0jHL: <RSCAN0_base> + $050A_H$ + (10_H × j),
RSCAN0GAFLP0jHH: <RSCAN0_base> + $050B_H$ + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLDLC[3:0]				GAFLPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLRMV	GAFLRMDP[6:0]							—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 22.32 RSCAN0GAFLP0j Register Contents

Bit Position	Bit Name	Function
31 to 28	GAFLDLC[3:0]	Receive Rule DLC b31 b30 b29 b28 0 0 0 0: DLC check is disabled. 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	GAFLPTR[11:0]	Receive Rule Label Set the 12-bit label information.
15	GAFLRMV	Receive Buffer Enable 0: No receive buffer is used. 1: A receive buffer is used.
14 to 8	GAFLRMDP[6:0]	Receive Buffer Number Select Set the receive buffer number to store receive messages.
7 to 0	Reserved	These bits are read as the value after reset. The write value should be the value after reset.

Modify the RSCAN0GAFLP0j register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLDLC[3:0] Bits

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000_B disables the DLC check function allowing messages with any data length to pass the DLC check.

GAFLPTR[11:0] Bits

These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

GAFLRMV Bit

When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

GAFLRMDP[6:0] Bits

These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the NRXMB[7:0] bits in the RSCAN0RMNB register.

22.3.17 RSCAN0GAFLP1j — Receive Rule Pointer 1 Register (j = 0 to 15)

Access: RSCAN0GAFLP1j register can be read/written in 32-bit units
RSCAN0GAFLP1jL, RSCAN0GAFLP1jH registers can be read/written in 16-bit units
RSCAN0GAFLP1jLL, RSCAN0GAFLP1jLH, RSCAN0GAFLP1jHL, RSCAN0GAFLP1jHH registers can be read/written in 8-bit units

Address: RSCAN0GAFLP1j: <RSCAN0_base> + $050C_H$ + (10_H × j)

RSCAN0GAFLP1jL: <RSCAN0_base> + $050C_H$ + (10_H × j),
RSCAN0GAFLP1jH: <RSCAN0_base> + $050E_H$ + (10_H × j)

RSCAN0GAFLP1jLL: <RSCAN0_base> + $050C_H$ + (10_H × j),
RSCAN0GAFLP1jLH: <RSCAN0_base> + $050D_H$ + (10_H × j),
RSCAN0GAFLP1jHL: <RSCAN0_base> + $050E_H$ + (10_H × j),
RSCAN0GAFLP1jHH: <RSCAN0_base> + $050F_H$ + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GAFLFDP[25:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLFDP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.33 RSCAN0GAFLP1j Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
25 to 8	GAFLFDP[25:8]	Transmit/Receive FIFO Buffer k Select (Bit position –8 = target transmit/receive FIFO buffer number k) 0: Transmit/receive FIFO buffer is not selected. 1: Transmit/receive FIFO buffer is selected.
7 to 0	GAFLFDP[7:0]	Receive FIFO Buffer x Select (Bit position = target receive FIFO buffer number x) 0: Receive FIFO buffer is not selected. 1: Receive FIFO buffer is selected.

Modify the RSCAN0GAFLP1j register when the AFLDAE bit in the RSCAN0GAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLFDP [25:0] Bits

These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to eight FIFO buffers are selectable. However, when the GAFLRMV bit in the RSCAN0GAFLP0j register is set to 1 (a message is stored in the receive buffer), up to seven FIFO buffers can be selected. Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFM[1:0] bits in the RSCAN0CFCCk register are set to 00_B (receive mode) or 10_B (gateway mode) are selectable.

22.3.18 RSCAN0RMNB — Receive Buffer Number Register

Access: RSCAN0RMNB register can be read or written in 32-bit units
RSCAN0RMNBL register can be read or written in 16-bit units
RSCAN0RMNBLL register can be read or written in 8-bit units

Address: RSCAN0RMNB: <RSCAN0_base> + $00A4_H$

RSCAN0RMNBL: <RSCAN0_base> + $00A4_H$

RSCAN0RMNBLL: <RSCAN0_base> + $00A4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	NRXMB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.34 RSCAN0RMNB Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
7 to 0	NRXMB[7:0]	Receive Buffer Number Configuration Set the number of receive buffers. Set a value of 0 to 48.

Modify the RSCAN0RMNB register only in global reset mode.

NRXMB[7:0] Bits

These bits are used to set the total number of receive buffers of the RSCAN module. The maximum value is 16 × (number of channels).

Setting these bits all to 0 makes receive buffers unavailable.

22.3.19 RSCAN0RMNDy — Receive Buffer New Data Register (y = 0 to 1)

Access: RSCAN0RMNDy register can be read/written in 32-bit units
RSCAN0RMNDyL, RSCAN0RMNDyH registers can be read/written in 16-bit units
RSCAN0RMNDyLL, RSCAN0RMNDyLH, RSCAN0RMNDyHL, RSCAN0RMNDyHH registers can be read/written in 8-bit units

Address: RSCAN0RMNDy: <RSCAN0_base> + $00A8_H$ + (04_H × y)

RSCAN0RMNDyL: <RSCAN0_base> + $00A8_H$ + (04_H × y),
RSCAN0RMNDyH: <RSCAN0_base> + $00AA_H$ + (04_H × y)

RSCAN0RMNDyLL: <RSCAN0_base> + $00A8_H$ + (04_H × y),
RSCAN0RMNDyLH: <RSCAN0_base> + $00A9_H$ + (04_H × y),
RSCAN0RMNDyHL: <RSCAN0_base> + $00AA_H$ + (04_H × y),
RSCAN0RMNDyHH: <RSCAN0_base> + $00AB_H$ + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMNSq (q = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMNSq (q = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.35 RSCAN0RMNDy Register Contents

Bit Position	Bit Name	Function
31 to 16	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 31 to y × 32 + 16) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.
15 to 0	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 15 to y × 32 + 0) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.

Write 0 to the RSCAN0RMNDy register in global operating mode or global test mode.

RMNSq Flags (q = 0 to 47)

Each RMNS flag is set to 1 when the processing for storing a message in the corresponding receive buffer starts.

To clear a flag to 0, the program must write 0 to the flag. Use a store instruction to write “0” to the flag and “1” to other flags. These bits cannot be set to 0 while a message is being stored. It takes ten clock cycles of pclk to store a message.

These flags are cleared to 0 in global reset mode.

22.3.20 RSCAN0RMIDq — Receive Buffer ID Register (q = 0 to 47)

Access: RSCAN0RMIDq register can be read only in 32-bit units
RSCAN0RMIDqL, RSCAN0RMIDqH registers can be read only in 16-bit units
RSCAN0RMIDqLL, RSCAN0RMIDqLH, RSCAN0RMIDqHL, RSCAN0RMIDqHH registers can be read only in 8-bit units

Address: RSCAN0RMIDq: <RSCAN0_base> + 0600_H + (10_H × q)

RSCAN0RMIDqL: <RSCAN0_base> + 0600_H + (10_H × q),
RSCAN0RMIDqH: <RSCAN0_base> + 0602_H + (10_H × q)

RSCAN0RMIDqLL: <RSCAN0_base> + 0600_H + (10_H × q),
RSCAN0RMIDqLH: <RSCAN0_base> + 0601_H + (10_H × q),
RSCAN0RMIDqHL: <RSCAN0_base> + 0602_H + (10_H × q),
RSCAN0RMIDqHH: <RSCAN0_base> + 0603_H + (10_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMIDE	RMRTR	—	RMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.36 RSCAN0RMIDq Register Contents

Bit Position	Bit Name	Function
31	RMIDE	Receive Buffer IDE 0: Standard ID 1: Extended ID
30	RMRTR	Receive Buffer RTR 0: Data frame 1: Remote frame
29	Reserved	These bits are read as the value after reset.
28 to 0	RMID[28:0]	Receive Buffer ID Data These bits contain the standard ID or extended ID of the received message. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RMIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

RMRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer.

RMID[28:0] Bits

These bits contain the ID of the message stored in the receive buffer.

22.3.21 RSCAN0RMPTRq — Receive Buffer Pointer Register (q = 0 to 47)

Access: RSCAN0RMPTRq register can be read only in 32-bit units
RSCAN0RMPTRqL, RSCAN0RMPTRqH registers can be read only in 16-bit units
RSCAN0RMPTRqLL, RSCAN0RMPTRqLH, RSCAN0RMPTRqHL, RSCAN0RMPTRqHH registers can be read only in 8-bit units

Address: RSCAN0RMPTRq: <RSCAN0_base> + 0604_H + (10_H × q)

RSCAN0RMPTRqL: <RSCAN0_base> + 0604_H + (10_H × q),
RSCAN0RMPTRqH: <RSCAN0_base> + 0606_H + (10_H × q)

RSCAN0RMPTRqLL: <RSCAN0_base> + 0604_H + (10_H × q),
RSCAN0RMPTRqLH: <RSCAN0_base> + 0605_H + (10_H × q),
RSCAN0RMPTRqHL: <RSCAN0_base> + 0606_H + (10_H × q),
RSCAN0RMPTRqHH: <RSCAN0_base> + 0607_H + (10_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDLC[3:0]				RMPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.37 RSCAN0RMPTRq Register Contents

Bit Position	Bit Name	Function
31 to 28	RMDLC[3:0]	Receive Buffer DLC Data b31 b30 b29 b28 0 0 0 0: No data byte 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	RMPTR[11:0]	Receive Buffer Label Data Label information of the received message.
15 to 0	RMTS[15:0]	Receive Buffer Timestamp Data Timestamp value of the received message.

RMDLC[3:0] Bits

These bits indicate the data length of the message stored in the receive buffer.

RMPTR[11:0] Bits

These bits indicate the label information of the message stored in the receive buffer.

RMTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the receive buffer.

22.3.22 RSCAN0RMDF0q — Receive Buffer Data Field 0 Register (q = 0 to 47)

Access: RSCAN0RMDF0q register can be read only in 32-bit units
RSCAN0RMDF0qL, RSCAN0RMDF0qH registers can be read only in 16-bit units
RSCAN0RMDF0qLL, RSCAN0RMDF0qLH, RSCAN0RMDF0qHL, RSCAN0RMDF0qHH registers can be read only in 8-bit units

Address: RSCAN0RMDF0q: <RSCAN0_base> + 0608_H + (10_H × q)

RSCAN0RMDF0qL: <RSCAN0_base> + 0608_H + (10_H × q),
RSCAN0RMDF0qH: <RSCAN0_base> + $060A_H$ + (10_H × q)

RSCAN0RMDF0qLL: <RSCAN0_base> + 0608_H + (10_H × q),
RSCAN0RMDF0qLH: <RSCAN0_base> + 0609_H + (10_H × q),
RSCAN0RMDF0qHL: <RSCAN0_base> + $060A_H$ + (10_H × q),
RSCAN0RMDF0qHH: <RSCAN0_base> + $060B_H$ + (10_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB3[7:0]								RMDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB1[7:0]								RMDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.38 RSCAN0RMDF0q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB3[7:0]	Receive Buffer Data Byte 3
23 to 16	RMDB2[7:0]	Receive Buffer Data Byte 2
15 to 8	RMDB1[7:0]	Receive Buffer Data Byte 1
7 to 0	RMDB0[7:0]	Receive Buffer Data Byte 0 Data for a message stored in the receive buffer can be read.

When the RMDLC[3:0] value in the RSCAN0RMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

22.3.23 RSCAN0RMDF1q — Receive Buffer Data Field 1 Register (q = 0 to 47)

Access: RSCAN0RMDF1q register can be read only in 32-bit units
RSCAN0RMDF1qL, RSCAN0RMDF1qH register can be read only in 16-bit units
RSCAN0RMDF1qLL, RSCAN0RMDF1qLH, RSCAN0RMDF1qHL, RSCAN0RMDF1qHH registers can be read only in 8-bit units

Address: RSCAN0RMDF1q: <RSCAN0_base> + $060C_H$ + (10_H × q)

RSCAN0RMDF1qL: <RSCAN0_base> + $060C_H$ + (10_H × q),
RSCAN0RMDF1qH: <RSCAN0_base> + $060E_H$ + (10_H × q)

RSCAN0RMDF1qLL: <RSCAN0_base> + $060C_H$ + (10_H × q),
RSCAN0RMDF1qLH: <RSCAN0_base> + $060D_H$ + (10_H × q),
RSCAN0RMDF1qHL: <RSCAN0_base> + $060E_H$ + (10_H × q),
RSCAN0RMDF1qHH: <RSCAN0_base> + $060F_H$ + (10_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB7[7:0]								RMDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB5[7:0]								RMDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.39 RSCAN0RMDF1q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB7[7:0]	Receive Buffer Data Byte 7
23 to 16	RMDB6[7:0]	Receive Buffer Data Byte 6
15 to 8	RMDB5[7:0]	Receive Buffer Data Byte 5
7 to 0	RMDB4[7:0]	Receive Buffer Data Byte 4 Data for a message stored in the receive buffer can be read.

When the RMDLC[3:0] value in the RSCAN0RMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

22.3.24 RSCAN0RFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7)

Access: RSCAN0RFCCx register can be read/written in 32-bit units
RSCAN0RFCCxL register can be read/written in 16-bit units
RSCAN0RFCCxLL, RSCAN0RFCCxLH registers can be read/written in 8-bit units

Address: RSCAN0RFCCx: <RSCAN0_base> + $00B8_H$ + (04_H × x)

RSCAN0RFCCxL: <RSCAN0_base> + $00B8_H$ + (04_H × x)

RSCAN0RFCCxLL: <RSCAN0_base> + $00B8_H$ + (04_H × x),
RSCAN0RFCCxLH: <RSCAN0_base> + $00B9_H$ + (04_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	—	—	—	—	—	RFIE	RFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 22.40 RSCAN0RFCCx Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
15 to 13	RFIGCV[2:0]	Receive FIFO Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.
12	RFIM	Receive FIFO Interrupt Source Select 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met. 1: An interrupt occurs each time a message has been received.
11	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
10 to 8	RFDC[2:0]	Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7 to 2	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
1	RFIE	Receive FIFO Interrupt Enable 0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled.

Table 22.40 RSCAN0RFCCx Register Contents (2/2)

Bit Position	Bit Name	Function
0	RFE	Receive FIFO Buffer Enable 0: No receive FIFO buffer is used. 1: Receive FIFO buffers are used.

RFIGCV[2:0] Bits

These bits are used to specify the number of received messages for generating a receive FIFO interrupt request when the RFIM bit is set to 0 with a fraction for the total number of buffers (the setting of RFDC[2:0]).

When the RFDC[2:0] bits are set to 001_B (4 messages), set the RFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B. Modify these bits only in global reset mode.

RFIM Bit

This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

RFDC[2:0] Bits

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. When these bits are set to 000_B, no receive FIFO buffer should be used. Modify these bits only in global reset mode.

RFIE Bit

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit set to 0 (no receive FIFO buffer is used).

RFE Bit

Setting the RFE bit to 1 makes receive FIFO buffers available. Clearing this bit to 0 sets the RFEMP flag in the RSCAN0RFSTSx register to 1 (buffer empty). Modify this bit in global operating mode or global test mode. Set this bit to 1 with another instruction after the settings to all bits in the RSCANnRFCCx register have been done. This bit is cleared to 0 in global reset mode.

22.3.25 RSCAN0RFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7)

Access: RSCAN0RFSTSx register can be read/written in 32-bit units
RSCAN0RFSTSxL register can be read/written in 16-bit units
RSCAN0RFSTSxLL register can be read/written in 8-bit units
RSCAN0RFSTSxLH register is a read-only register that can be read in 8-bit units

Address: RSCAN0RFSTSx: <RSCAN0_base> + $00D8_H$ + (04_H × x)

RSCAN0RFSTSxL: <RSCAN0_base> + $00D8_H$ + (04_H × x)

RSCAN0RFSTSxLL: <RSCAN0_base> + $00D8_H$ + (04_H × x),
RSCAN0RFSTSxLH: <RSCAN0_base> + $00D9_H$ + (04_H × x)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMC[7:0]								—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 22.41 RSCAN0RFSTSx Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
15 to 8	RFMC[7:0]	Receive FIFO Unread Message Counter The number of unread messages stored in the receive FIFO buffer is displayed.
7 to 4	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
3	RFIF	Receive FIFO Interrupt Request Flag 0: No receive FIFO interrupt request is present. 1: A receive FIFO interrupt request is present.
2	RFMLT	Receive FIFO Message Lost Flag 0: No receive FIFO message is lost. 1: A receive FIFO message is lost.
1	RFFLL	Receive FIFO Buffer Full Status Flag 0: The receive FIFO buffer is not full. 1: The receive FIFO buffer is full.
0	RFEMP	Receive FIFO Buffer Empty Status Flag 0: The receive FIFO buffer contains unread message. 1: The receive FIFO buffer contains no unread message (buffer empty).

RFMC[7:0] Flag

This flag indicates the number of unread messages in the receive FIFO buffer. This flag becomes 00_H when the RFE bit in the RSCAN0RFCCx register is set to 0.
This flag is 00_H in global reset mode.

RFIF Flag

This flag is set to 1 when the receive FIFO interrupt request generation conditions set by the RFIGCV[2:0] bits and the RFIM bit in the RSCAN0RFCCx register are met. This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit in global operating mode or global test mode.
To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFMLT Flag

This flag is set to 1 when an attempt is made to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.

This flag is cleared to 0 in global reset mode or by writing 0 to this flag.

Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFFLL Flag

This flag is set to 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFDC[2:0] bits in the RSCAN0RFCCx register.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFDC[2:0] bits, this flag is cleared to 0. This flag is also cleared to 0 when the RFE bit in the RSCAN0RFCCx register is set to 0 (no receive FIFO buffer is used) or in global reset mode.

RFEMP Flag

This flag is set to 1 when all messages in the receive FIFO buffer have been read. This flag is also set to 1 when the RFE bit in the RSCAN0RFCCx register is 0 or in global reset mode.

This flag is cleared to 0 when even a single received message has been stored in the receive FIFO buffer.

NOTE

To clear the RFMLT or RFIF flag to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

22.3.26 RSCAN0RFPCTRx — Receive FIFO Buffer Pointer Control Register (x = 0 to 7)

Access: RSCAN0RFPCTRx register can only be written in 32-bit units
RSCAN0RFPCTRxL register is a write-only register that can only be written in 16-bit units
RSCAN0RFPCTRxLL register is a write-only register that can only be written in 8-bit units

Address: RSCAN0RFPCTRx: <RSCAN0_base> + $00F8_H$ + (04_H × x)

RSCAN0RFPCTRxL: <RSCAN0_base> + $00F8_H$ + (04_H × x)

RSCAN0RFPCTRxLL: <RSCAN0_base> + $00F8_H$ + (04_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 22.42 RSCAN0RFPCTRx Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be the value after reset.
7 to 0	RFPC[7:0]	Receive FIFO Pointer Control When these bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer.

RFPC[7:0] Bits

When the RFPC[7:0] bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFMC[7:0] (receive FIFO unread message counter) value in the RSCAN0RFSTSx register is decremented by 1. Read the RSCAN0RFIDx, RSCAN0RFPTRx, RSCAN0RFDF0x, and RSCAN0RFDF1x registers to read messages in the receive FIFO buffer, and then write FF_H to the RFPC[7:0] bits.

When writing FF_H to these bits, make sure that the RFE bit in the RSCAN0RFCCx register is set to 1 (receive FIFO buffers are used) and the RFEMP flag in the RSCAN0RFSTSx register is 0 (the receive FIFO buffer contains unread messages).

22.3.27 RSCAN0RFIDx — Receive FIFO Buffer Access ID Register (x = 0 to 7)

Access: RSCAN0RFIDx register can be read only in 32-bit units
RSCAN0RFIDxL, RSCAN0RFIDxH registers can be read only in 16-bit units
RSCAN0RFIDxLL, RSCAN0RFIDxLH, RSCAN0RFIDxHL, RSCAN0RFIDxHH registers can be read only in 8-bit units

Address: RSCAN0RFIDx: <RSCAN0_base> + $0E00_H$ + (10_H × x)

RSCAN0RFIDxL: <RSCAN0_base> + $0E00_H$ + (10_H × x),
RSCAN0RFIDxH: <RSCAN0_base> + $0E02_H$ + (10_H × x)

RSCAN0RFIDxLL: <RSCAN0_base> + $0E00_H$ + (10_H × x),
RSCAN0RFIDxLH: <RSCAN0_base> + $0E01_H$ + (10_H × x),
RSCAN0RFIDxHL: <RSCAN0_base> + $0E02_H$ + (10_H × x),
RSCAN0RFIDxHH: <RSCAN0_base> + $0E03_H$ + (10_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFIDE	RFRTR	—	RFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.43 RSCAN0RFIDx Register Contents

Bit Position	Bit Name	Function
31	RFIDE	Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	RFRTR	Receive FIFO Buffer RTR 0: Data frame 1: Remote frame
29	Reserved	These bits are read as the value after reset.
28 to 0	RFID[28:0]	Receive FIFO Buffer ID Data The standard ID or extended ID of received message can be read. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

RFRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer.

RFID[28:0] Bits

These bits indicate the ID of the message stored in the receive FIFO buffer.

22.3.28 RSCAN0RFPTRx — Receive FIFO Buffer Access Pointer Register (x = 0 to 7)

Access: RSCAN0RFPTRx register can be read only in 32-bit units
RSCAN0RFPTRxL, RSCAN0RFPTRxH registers can be read only in 16-bit units
RSCAN0RFPTRxLL, RSCAN0RFPTRxLH, RSCAN0RFPTRxHL, RSCAN0RFPTRxHH registers can be read only in 8-bit units

Address: RSCAN0RFPTRx: <RSCAN0_base> + $0E04_H$ + (10_H × x)

RSCAN0RFPTRxL: <RSCAN0_base> + $0E04_H$ + (10_H × x),
RSCAN0RFPTRxH: <RSCAN0_base> + $0E06_H$ + (10_H × x)

RSCAN0RFPTRxLL: <RSCAN0_base> + $0E04_H$ + (10_H × x),
RSCAN0RFPTRxLH: <RSCAN0_base> + $0E05_H$ + (10_H × x),
RSCAN0RFPTRxHL: <RSCAN0_base> + $0E06_H$ + (10_H × x),
RSCAN0RFPTRxHH: <RSCAN0_base> + $0E07_H$ + (10_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDLC[3:0]				RFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.44 RSCAN0RFPTRx Register Contents

Bit Position	Bit Name	Function
31 to 28	RFDLC[3:0]	Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	RFPTR[11:0]	Receive FIFO Buffer Label Data Label information of the received message can be read.
15 to 0	RFTS[15:0]	Receive FIFO Buffer Timestamp Data Timestamp value of the received message can be read.

RFDLC[3:0] Bits

These bits contain the data length of the message stored in the receive FIFO buffer.

RFPTR[11:0] Bits

These bits contain the label information of the message stored in the receive FIFO buffer.

RFTS[15:0] Bits

These bits contain the timestamp value of the message stored in the receive FIFO buffer.

22.3.29 RSCAN0RFDF0x — Receive FIFO Buffer Access Data Field 0 Register (x = 0 to 7)

Access: RSCAN0RFDF0x register can be read-only in 32-bit units
RSCAN0RFDF0xL, RSCAN0RFDF0xH registers can be read only in 16-bit units
RSCAN0RFDF0xLL, RSCAN0RFDF0xLH, RSCAN0RFDF0xHL, RSCAN0RFDF0xHH registers can be read only in 8-bit units

Address: RSCAN0RFDF0x: <RSCAN0_base> + $0E08_H$ + (10_H × x)

RSCAN0RFDF0xL: <RSCAN0_base> + $0E08_H$ + (10_H × x),
RSCAN0RFDF0xH: <RSCAN0_base> + $0E0A_H$ + (10_H × x)

RSCAN0RFDF0xLL: <RSCAN0_base> + $0E08_H$ + (10_H × x),
RSCAN0RFDF0xLH: <RSCAN0_base> + $0E09_H$ + (10_H × x),
RSCAN0RFDF0xHL: <RSCAN0_base> + $0E0A_H$ + (10_H × x),
RSCAN0RFDF0xHH: <RSCAN0_base> + $0E0B_H$ + (10_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB3[7:0]								RFDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB1[7:0]								RFDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.45 RSCAN0RFDF0x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB3[7:0]	Receive FIFO Buffer Data Byte 3
23 to 16	RFDB2[7:0]	Receive FIFO Buffer Data Byte 2
15 to 8	RFDB1[7:0]	Receive FIFO Buffer Data Byte 1
7 to 0	RFDB0[7:0]	Receive FIFO Buffer Data Byte 0 Data for a message stored in the receive FIFO buffer can be read.

When the RFDLC[3:0] value in the RSCAN0RFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

22.3.30 RSCAN0RFDF1x — Receive FIFO Buffer Access Data Field 1 Register (x = 0 to 7)

Access: RSCAN0RFDF1x register can be read only in 32-bit units
RSCAN0RFDF1xL, RSCAN0RFDF1xH registers can be read only in 16-bit units
RSCAN0RFDF1xLL, RSCAN0RFDF1xLH, RSCAN0RFDF1xHL, RSCAN0RFDF1xHH registers can be read only in 8-bit units

Address: RSCAN0RFDF1x: <RSCAN0_base> + $0E0C_H$ + (10_H × x)

RSCAN0RFDF1xL: <RSCAN0_base> + $0E0C_H$ + (10_H × x),
RSCAN0RFDF1xH: <RSCAN0_base> + $0E0E_H$ + (10_H × x)

RSCAN0RFDF1xLL: <RSCAN0_base> + $0E0C_H$ + (10_H × x),
RSCAN0RFDF1xLH: <RSCAN0_base> + $0E0D_H$ + (10_H × x),
RSCAN0RFDF1xHL: <RSCAN0_base> + $0E0E_H$ + (10_H × x),
RSCAN0RFDF1xHH: <RSCAN0_base> + $0E0F_H$ + (10_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB7[7:0]								RFDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB5[7:0]								RFDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.46 RSCAN0RFDF1x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB7[7:0]	Receive FIFO Buffer Data Byte 7
23 to 16	RFDB6[7:0]	Receive FIFO Buffer Data Byte 6
15 to 8	RFDB5[7:0]	Receive FIFO Buffer Data Byte 5
7 to 0	RFDB4[7:0]	Receive FIFO Buffer Data Byte 4 Data for a message stored in the receive FIFO buffer can be read.

When the RFDLC[3:0] value in the RSCAN0RFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

22.3.31 RSCAN0CFCCk — Transmit/receive FIFO Buffer Configuration and Control Register k (k = 0 to 8)

Access: RSCAN0CFCCk register can be read/written in 32-bit units
RSCAN0CFCCkL, RSCAN0CFCCkH registers can be read/written in 16-bit units
RSCAN0CFCCkLL, RSCAN0CFCCkLH, RSCAN0CFCCkHL, RSCAN0CFCCkHH registers can be read/written in 8-bit units

Address: RSCAN0CFCCk: <RSCAN0_base> + 0118_H + (04_H × k)

RSCAN0CFCCkL: <RSCAN0_base> + 0118_H + (04_H × k),
RSCAN0CFCCkH: <RSCAN0_base> + $011A_H$ + (04_H × k)

RSCAN0CFCCkLL: <RSCAN0_base> + 0118_H + (04_H × k),
RSCAN0CFCCkLH: <RSCAN0_base> + 0119_H + (04_H × k),
RSCAN0CFCCkHL: <RSCAN0_base> + $011A_H$ + (04_H × k),
RSCAN0CFCCkHH: <RSCAN0_base> + $011B_H$ + (04_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFITT[7:0]								CFTML[3:0]				CFITR	CFITSS	CFM[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFIGCV[2:0]			CFIM	—	CFDC[2:0]			—	—	—	—	—	CFTXIE	CFRXIE	CFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 22.47 RSCAN0CFCCk Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	CFITT[7:0]	Set a message transmission interval. Set Value: 00_H to FF_H
23 to 20	CFTML[3:0]	Transmit Buffer Link Configuration Set the transmit buffer number to be linked to the transmit/receive FIFO buffer.
19	CFITR	Transmit/Receive FIFO Interval Timer Resolution 0: Clock dividing pclk/2 by (ITRCP [15:0] bits) 1: Clock dividing pclk/2 by (ITRCP [15:0] bits × 10)
18	CFITSS	Transmit/Receive FIFO Interval Timer Clock Source Select 0: Interval timer clock source selected by the CFITR bit 1: Interval timer clock source is the bit time clock for the channel to which the FIFO is linked.
17, 16	CFM[1:0]	Transmit/Receive FIFO Mode Select b17 b16 0 0: Receive mode 0 1: Transmit mode 1 0: Gateway mode 1 1: Setting prohibited
15 to 13	CFIGCV[2:0]	Transmit/Receive FIFO Receive Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.

Table 22.47 RSCAN0CFCCk Register Contents (2/2)

Bit Position	Bit Name	Function
12	CFIM	Transmit/Receive FIFO Interrupt Source Select 0: • Receive mode/gateway mode When the number of received messages has met the condition set by the CFIGCV[2:0] bits, a FIFO receive interrupt request is generated. • Transmit mode/gateway mode When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated. 1: • Receive mode/gateway mode A FIFO receive interrupt request is generated each time a message has been received. • Transmit mode/gateway mode A FIFO transmit interrupt request is generated each time a message has been transmitted.
11	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
10 to 8	CFDC[2:0]	Transmit/Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7 to 3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2	CFTXIE	Transmit/Receive FIFO Transmit Interrupt Enable 0: Transmit/receive FIFO transmit interrupt is disabled. 1: Transmit/receive FIFO transmit interrupt is enabled.
1	CFRXIE	Transmit/Receive FIFO Receive Interrupt Enable 0: Transmit/receive FIFO receive interrupt is disabled. 1: Transmit/receive FIFO receive interrupt is enabled.
0	CFE	Transmit/Receive FIFO Buffer Enable 0: No transmit/receive FIFO buffer is used. 1: Transmit/receive FIFO buffers are used.

CFITT[7:0] Bits

These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode).

Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITT[7:0] bits.

CFTML[3:0] Bits

These bits are used to set the number of transmit buffer on the channel which will be linked to transmit/receive FIFO buffer k when the CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode). There are three transmit/receive FIFO buffers per channel, so channel number m of FIFO buffer k is calculated as m = k/3 (integer division). The actual assigned transmit buffer number p linked to FIFO buffer k will be ((16 × m) + CFTML[3:0]).

See **Table 22.15** and **Table 22.16**, as for the relationship between transmit/receive FIFO buffer k and transmit buffer p.

Setting the CFDC[2:0] bits to 001_B or more enables the setting of the CFTML[3:0] bits.

Do not link to any transmit buffer which is already allocated to a transmit queue on the same channel or to another transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFITR Bit

This bit is enabled when the CFITSS bit is 0.

When this bit is 0, the interval timer clock source is the pclk/2 clock divided by the value of the ITRCP[15:0] bits in the RSCAN0GCFG register.

When this bit is 1, the interval timer clock source is the pclk/2 clock divided by (the value of the ITRCP[15:0] bits in the RSCAN0GCFG register × 10).

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFITSS Bit

When this bit is 0, the clock selected by the CFITR bit is the count source of the interval timer.

When this bit is 1, the bit time clock of the channel to which the FIFO is linked is the count source of the interval timer.

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFM[1:0] Bits

These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

CFIGCV[2:0] Bits

These bits are used to specify the number of received messages for generating a transmit/receive FIFO receive interrupt request when the CFM[1:0] bits are set to 00_B (receive mode) or 10_B (gateway mode) and the CFIM bit is set to 0 with a fraction for the total number of buffers (the setting of CFDC[2:0]).

When the CFDC[2:0] bits are set to 001_B (4 messages), set the CFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B.

Modify these bits only in global reset mode.

CFIM Bit

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

CFDC[2:0] Bits

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer. When these bits are set to 000_B, do not use a transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFTXIE Bit

When this bit is set to 1 and the CFTXIF flag in the RSCAN0CFSTSk register is set to 1, a transmit/receive FIFO transmit interrupt request is generated.

Modify this bit with the CFE bit set to 0 (no transmit/receive FIFO buffer is used).

CFRXIE Bit

When this bit is set to 1 and the CFRXIF flag in the RSCAN0CFSTSk register is set to 1, a transmit/receive FIFO receive interrupt request is generated.

Modify this bit with the CFE bit set to 0.

CFE Bit

Setting this bit to 1 makes transmit/receive FIFO buffers available.

When this bit is set to 0 in transmit mode or gateway mode, if a message in the transmit/receive FIFO buffer is being transmitted or will be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission of that message, or upon detection of a CAN bus error, or arbitration-lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

This bit is cleared to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode or gateway mode: Channel reset mode

Modify this bit in the following mode.

- Receive mode: Global operating mode or global test mode
- Transmit mode or gateway mode: Channel communication mode or channel halt mode

After all other bits in the RSCANnCFCCk register have been set, set this bit to 1 by using other instructions.

22.3.32 RSCAN0CFSTSk — Transmit/receive FIFO Buffer Status Register (k = 0 to 8)

Access: RSCAN0CFSTSk register can be read/written in 32-bit units
RSCAN0CFSTSkL register can be read/written in 16-bit units
RSCAN0CFSTSkLL register can be read/written in 8-bit units
RSCAN0CFSTSkLH register is a read-only register that can be read in 8-bit units.

Address: RSCAN0CFSTSk: <RSCAN0_base> + 0178_H + (04_H × k)

RSCAN0CFSTSkL: <RSCAN0_base> + 0178_H + (04_H × k)

RSCAN0CFSTSkLL: <RSCAN0_base> + 0178_H + (04_H × k),
RSCAN0CFSTSkLH: <RSCAN0_base> + 0179_H + (04_H × k)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFMC[7:0]								—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 22.48 RSCAN0CFSTSk Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
15 to 8	CFMC[7:0]	Transmit/Receive FIFO Message Counter The number of messages stored in the transmit/receive FIFO buffer.
7 to 5	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
4	CFTXIF	Transmit/Receive FIFO Transmit Interrupt Request Flag 0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present.
3	CFRXIF	Transmit/Receive FIFO Receive Interrupt Request Flag 0: No transmit/receive FIFO receive interrupt request is present. 1: A transmit/receive FIFO receive interrupt request is present.
2	CFMLT	Transmit/Receive FIFO Message Lost Flag 0: No transmit/receive FIFO message is lost. 1: A transmit/receive FIFO message is lost.
1	CFFLL	Transmit/Receive FIFO Buffer Full Status Flag 0: The transmit/receive FIFO buffer is not full. 1: The transmit/receive FIFO buffer is full.
0	CFEMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: The transmit/receive FIFO buffer contains messages. 1: The transmit/receive FIFO buffer contains no message (buffer empty).

CFMC[7:0] Bits

The CFMC[7:0] bits indicate the following values that depend on the setting of the CFM[1:0] bits in the RSCAN0CFCCk register.

- When CFM[1:0] value is 01_B (transmit mode): Number of untransmitted messages in the buffer

- When CFM[1:0] value is 00_B (receive mode): Number of unread received messages in the buffer
- When CFM[1:0] value is 10_B (gateway mode): Number of untransmitted received messages in the buffer

These bits are cleared to 0 when any of the following conditions is met.

- When CFM[1:0] value is 00_B: In global reset mode
- When CFM[1:0] value is 01_B or 10_B: In channel reset mode
- When the CFE bit in the RSCANnCFCCk register is cleared to 0.

CFTXIF Flag

The CFTXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 01_B or 10_B, and the factor selected by the CFIM bit in the RSCAN0CFCCk register occurs

The CFTXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFTXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFRXIF Flag

The CFRXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B, and the factor selected by the CFIM bit in the RSCAN0CFCCk register occurs

The CFRXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFRXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFMLT Flag

The CFMLT flag is set to 1 when any of the following conditions is met.

- When an attempt is made to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

The CFMLT flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFMLT flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFFLL Flag

The CFFLL flag is set to 1 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFDC[2:0] bits in the RSCAN0CFCCk register.

The CFFLL flag is cleared to 0 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFDC[2:0] bits.
- When the CFE bit in the RSCAN0CFCCk register is 0 (no transmit/receive FIFO buffer is used): When not in the transmit abort
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

CFEMP Flag

The CFEMP flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B: All messages have been read, or in global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: All messages have been transmitted, or in channel reset mode
- When the CFE bit is 0 (no transmit/receive FIFO buffer is used): Not in the transmit abort

The CFEMP flag is cleared to 0 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B: At least one received message has been stored in the transmit/receive FIFO buffer.
- When the CFM[1:0] bits are set to 01_B: A value of FF_H has been written to the RSCAN0CFPCTRk register after data was written to the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers.

NOTE

To clear CFTXIF, CFRXIF, or CFMLT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

22.3.33 RSCAN0CFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register (k = 0 to 8)

Access: RSCAN0CFPCTRk register can only be written in 32-bit units
RSCAN0CFPCTRkL register is a write-only register that can only be written in 16-bit units
RSCAN0CFPCTRkLL register is a write-only register that can only be written in 8-bit units

Address: RSCAN0CFPCTRk: <RSCAN0_base> + $01D8_H$ + (04_H × k)

RSCAN0CFPCTRkL: <RSCAN0_base> + $01D8_H$ + (04_H × k)

RSCAN0CFPCTRkLL: <RSCAN0_base> + $01D8_H$ + (04_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 22.49 RSCAN0CFPCTRk Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be the value after reset.
7 to 0	CFPC[7:0]	Transmit/Receive FIFO Pointer Control • Receive mode: Writing FF_H to these bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. • Transmit mode: Writing FF_H to these bits moves the write pointer to the next stage of the transmit/receive FIFO buffer. • Gateway mode: Setting prohibited

CFPC[7:0] Bits

- Receive mode (CFM[1:0] value in the RSCAN0CFCCk register is 00_B):
  Writing FF_H to the CFPC[7:0] bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value (transmit/receive FIFO message counter) in the RSCAN0CFSTSk register is decremented by 1. Read the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers to read messages from the transmit/receive FIFO buffer, and then write FF_H to the CFPC[7:0] bits.
  When writing FF_H to these bits, make sure that the CFE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO buffers are used) and the CFEMP flag in the RSCAN0CFSTSk register is 0 (the transmit/receive FIFO buffer contains messages).
- Transmit mode (CFM[1:0] value in the RSCAN0CFCCk register is 01_B):
  Writing FF_H to the CFPC[7:0] bits stores the data written to the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers in the transmit/receive FIFO buffer and moves the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value is incremented by 1. Write transmit messages to the

RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers before writing FF_H to the CFPC[7:0] bits.
When writing FF_H to these bits, make sure that the CFE bit in the RSCAN0CFCCk register is set to 1 and the CFFLL flag in the RSCAN0CFSTSk register is 0 (the transmit/receive FIFO buffer is not full).

- Gateway mode (CFM[1:0] value in the RSCAN0CFCCk register is 10_B):
  Setting prohibited

22.3.34 RSCAN0CFIDk — Transmit/receive FIFO Buffer Access ID Register (k = 0 to 8)

Access: RSCAN0CFIDk register can be read/written in 32-bit units
RSCAN0CFIDkL, RSCAN0CFIDkH registers can be read/written in 16-bit units
RSCAN0CFIDkLL, RSCAN0CFIDkLH, RSCAN0CFIDkHL, RSCAN0CFIDkHH registers can be read/written in 8-bit units

Address: RSCAN0CFIDk: <RSCAN0_base> + $0E80_H$ + (10_H × k)

RSCAN0CFIDkL: <RSCAN0_base> + $0E80_H$ + (10_H × k),
RSCAN0CFIDkH: <RSCAN0_base> + $0E82_H$ + (10_H × k)

RSCAN0CFIDkLL: <RSCAN0_base> + $0E80_H$ + (10_H × k),
RSCAN0CFIDkLH: <RSCAN0_base> + $0E81_H$ + (10_H × k),
RSCAN0CFIDkHL: <RSCAN0_base> + $0E82_H$ + (10_H × k),
RSCAN0CFIDkHH: <RSCAN0_base> + $0E83_H$ + (10_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFIDE	CFRTR	THLEN	CFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.50 RSCAN0CFIDk Register Contents

Bit Position	Bit Name	Function
31	CFIDE	Transmit/Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	CFRTR	Transmit/Receive FIFO Buffer RTR 0: Data frame 1: Remote frame
29	THLEN	Transmit History Data Store Enable This bit is valid only when the CFM[1:0] value is 01_B (transmit mode). 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	CFID[28:0]	Transmit/Receive FIFO Buffer ID Data • When CFM[1:0] value is 01_B (transmit mode): Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11. • When CFM[1:0] value is 00_B (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, read bits 10 to 0. Bits 28 to 11 are read as 0.

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). This RSCAN0CFIDk register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, these bits are used to set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

CFRTR Bit

This bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, this bit is used to set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmit messages is stored in the transmit history buffer after transmission is completed.

This bit is enabled when the CFM[1:0] value is 01_B (transmit mode).

CFID[28:0] Bits

These bits contain the ID of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

When the CFM[1:0] value is 01_B, this bit is used to set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

22.3.35 RSCAN0CFPTRk — Transmit/receive FIFO Buffer Access Pointer Register (k = 0 to 8)

Access: RSCAN0CFPTRk register can be read/written in 32-bit units
RSCAN0CFPTRkL, RSCAN0CFPTRkH registers can be read/written in 16-bit units
RSCAN0CFPTRkLL, RSCAN0CFPTRkLH, RSCAN0CFPTRkHL, RSCAN0CFPTRkHH registers can be read/written in 8-bit units

Address: RSCAN0CFPTRk: <RSCAN0_base> + $0E84_H$ + (10_H × k)

RSCAN0CFPTRkL: <RSCAN0_base> + $0E84_H$ + (10_H × k),
RSCAN0CFPTRkH: <RSCAN0_base> + $0E86_H$ + (10_H × k)

RSCAN0CFPTRkLL: <RSCAN0_base> + $0E84_H$ + (10_H × k),
RSCAN0CFPTRkLH: <RSCAN0_base> + $0E85_H$ + (10_H × k),
RSCAN0CFPTRkHL: <RSCAN0_base> + $0E86_H$ + (10_H × k),
RSCAN0CFPTRkHH: <RSCAN0_base> + $0E87_H$ + (10_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDLC[3:0]				CFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.51 RSCAN0CFPTRk Register Contents

Bit Position	Bit Name	Function
31 to 28	CFDLC[3:0]	Transmit/Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	CFPTR[11:0]	Transmit/Receive FIFO Buffer Label Data • When CFM[1:0] value is 01_B (transmit mode): Set the label information to be stored in the transmit history buffer. Only bits CFPTR[7:0] are valid. • When CFM[1:0] value is 00_B (receive mode): The label information of the received message can be read.
15 to 0	CFTS[15:0]	Transmit/Receive FIFO Buffer Timestamp Data These bits are valid only when the CFM[1:0] value is 00_B (receive mode). The timestamp value of the received message can be read.

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFDLC[3:0] Bits

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, these bits are used to set the data length of the message to be transmitted from the transmit/receive FIFO buffer. If the data length is set to 9 bytes or more, the actual transmit data defaults to 8 bytes.

CFPTR[11:0] Bits

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, the CFPTR[7:0] value is stored in the transmit history buffer when message transmission has been completed.

CFTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.

These bits are valid when the CFM[1:0] value is 00_B.

22.3.36 RSCAN0CFDF0k — Transmit/receive FIFO Buffer Access Data Field 0 Register (k = 0 to 8)

Access: RSCAN0CFDF0k register can be read/written in 32-bit units
RSCAN0CFDF0kL, RSCAN0CFDF0kH registers can be read/written in 16-bit units
RSCAN0CFDF0kLL, RSCAN0CFDF0kLH, RSCAN0CFDF0kHL, RSCAN0CFDF0kHH registers can be read/written in 8-bit units

Address: RSCAN0CFDF0k: <RSCAN0_base> + $0E88_H$ + (10_H × k)

RSCAN0CFDF0kL: <RSCAN0_base> + $0E88_H$ + (10_H × k),
RSCAN0CFDF0kH: <RSCAN0_base> + $0E8A_H$ + (10_H × k)

RSCAN0CFDF0kLL: <RSCAN0_base> + $0E88_H$ + (10_H × k),
RSCAN0CFDF0kLH: <RSCAN0_base> + $0E89_H$ + (10_H × k),
RSCAN0CFDF0kHL: <RSCAN0_base> + $0E8A_H$ + (10_H × k),
RSCAN0CFDF0kHH: <RSCAN0_base> + $0E8B_H$ + (10_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB3[7:0]								CFDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB1[7:0]								CFDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.52 RSCAN0CFDF0k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB3[7:0]	Transmit/Receive FIFO Buffer Data Byte 3
23 to 16	CFDB2[7:0]	Transmit/Receive FIFO Buffer Data Byte 2
15 to8	CFDB1[7:0]	Transmit/Receive FIFO Buffer Data Byte 1
7 to 0	CFDB0[7:0]	Transmit/Receive FIFO Buffer Data Byte 0 • When CFM[1:0] value is 01_B (transmit mode): Set the transmit/receive FIFO buffer data. • When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01_B (transmit mode).

This register is readable only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] value in the RSCAN0CFPTRk register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

22.3.37 RSCAN0CFDF1k — Transmit/receive FIFO Buffer Access Data Field 1 Register (k = 0 to 8)

Access: RSCAN0CFDF1k register can be read/written in 32-bit units
RSCAN0CFDF1kL, RSCAN0CFDF1kH registers can be read/written in 16-bit units
RSCAN0CFDF1kLL, RSCAN0CFDF1kLH, RSCAN0CFDF1kHL, RSCAN0CFDF1kHH registers can be read/written in 8-bit units

Address: RSCAN0CFDF1k: <RSCAN0_base> + $0E8C_H$ + (10_H × k)

RSCAN0CFDF1kL: <RSCAN0_base> + $0E8C_H$ + (10_H × k),
RSCAN0CFDF1kH: <RSCAN0_base> + $0E8E_H$ + (10_H × k)

RSCAN0CFDF1kLL: <RSCAN0_base> + $0E8C_H$ + (10_H × k),
RSCAN0CFDF1kLH: <RSCAN0_base> + $0E8D_H$ + (10_H × k),
RSCAN0CFDF1kHL: <RSCAN0_base> + $0E8E_H$ + (10_H × k),
RSCAN0CFDF1kHH: <RSCAN0_base> + $0E8F_H$ + (10_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB7[7:0]								CFDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB5[7:0]								CFDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.53 RSCAN0CFDF1k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB7[7:0]	Transmit/Receive FIFO Buffer Data Byte 7
23 to 16	CFDB6[7:0]	Transmit/Receive FIFO Buffer Data Byte 6
15 to 8	CFDB5[7:0]	Transmit/Receive FIFO Buffer Data Byte 5
7 to 0	CFDB4[7:0]	Transmit/Receive FIFO Buffer Data Byte 4 • When CFM[1:0] value is 01_B (transmit mode): Set the transmit/receive FIFO buffer data. • When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.

This register is writable only when the CFM[1:0] value in the RSCAN0CFCCk register is 01_B (transmit mode).

This register is readable only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] value in the RSCAN0CFPTRk register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

22.3.38 RSCAN0FESTS — FIFO Empty Status Register

Access: RSCAN0FESTSregisters can be read only in 32-bit units
RSCAN0FESTSL, RSCAN0FESTSH registers can be read only in 16-bit units
RSCAN0FESTSLL, RSCAN0FESTSLH, RSCAN0FESTSHL registers can be read only in 8-bit units

Address: RSCAN0FESTS: <RSCAN0_base> + 0238_H

RSCAN0FESTSL: <RSCAN0_base> + 0238_H,
RSCAN0FESTSH: <RSCAN0_base> + $023A_H$

RSCAN0FESTSLL: <RSCAN0_base> + 0238_H,
RSCAN0FESTSLH: <RSCAN0_base> + 0239_H,
RSCAN0FESTSHL: <RSCAN0_base> + $023A_H$

Value after reset: $03FF\ FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF8EMP
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7EMP	CF6EMP	CF5EMP	CF4EMP	CF3EMP	CF2EMP	CF1EMP	CF0EMP	RF7EMP	RF6EMP	RF5EMP	RF4EMP	RF3EMP	RF2EMP	RF1EMP	RF0EMP
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.54 RSCAN0FESTS Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	These bits are read as the value after reset.
16	CF8EMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: Transmit/receive FIFO buffer k contains a message. 1: Transmit/receive FIFO buffer k contains no message. (k = 0 to 8)
15	CF7EMP	
14	CF6EMP	
13	CF5EMP	
12	CF4EMP	
11	CF3EMP	
10	CF2EMP	
9	CF1EMP	
8	CF0EMP	
7	RF7EMP	Receive FIFO Buffer Empty Status Flag 0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message (buffer empty). (x = 0 to 7)
6	RF6EMP	
5	RF5EMP	
4	RF4EMP	
3	RF3EMP	
2	RF2EMP	
1	RF1EMP	
0	RF0EMP	

The RSCAN0FESTS register is set to $03FF\ FFFF_H$ in global reset mode.

CFkEMP Flag (k = 0 to 8)

The CFkEMP flag is set to 1 when the CFEMP flag in the RSCAN0CFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)). When the CFEMP flag is cleared to 0 (the transmit/receive FIFO buffer contains messages), the CFkEMP flag is cleared to 0.

RFxEMP Flag (x = 0 to 7)

The RFxEMP flag is set to 1 when the RFEMP flag in the RSCAN0RFSTSx register is set to 1 (the receive FIFO buffer contains no unread message). When the RFEMP flag is cleared to 0 (the receive FIFO buffer contains unread messages), the RFxEMP flag is cleared to 0.

22.3.39 RSCAN0FFSTS — FIFO Full Status Register

Access: RSCAN0FFSTS register can be read only in 32-bit units
RSCAN0FFSTSL, RSCAN0FFSTSH registers can be read only in 16-bit units
RSCAN0FFSTSLL, RSCAN0FFSTSLH, RSCAN0FFSTSHL registers can be read only in 8-bit units

Address: RSCAN0FFSTS: <RSCAN0_base> + $023C_H$

RSCAN0FFSTSL: <RSCAN0_base> + $023C_H$,
RSCAN0FFSTSH: <RSCAN0_base> + $023E_H$

RSCAN0FFSTSLL: <RSCAN0_base> + $023C_H$,
RSCAN0FFSTSLH: <RSCAN0_base> + $023D_H$,
RSCAN0FFSTSHL: <RSCAN0_base> + $023E_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF8FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7FLL	CF6FLL	CF5FLL	CF4FLL	CF3FLL	CF2FLL	CF1FLL	CF0FLL	RF7FLL	RF6FLL	RF5FLL	RF4FLL	RF3FLL	RF2FLL	RF1FLL	RF0FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.55 RSCAN0FFSTS Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	These bits are read as the value after reset.
16	CF8FLL	Transmit/Receive FIFO Buffer Full Status Flag 0: Transmit/receive buffer k is not full. 1: Transmit/receive buffer k is full. (k = 0 to 8)
15	CF7FLL	
14	CF6FLL	
13	CF5FLL	
12	CF4FLL	
11	CF3FLL	
10	CF2FLL	
9	CF1FLL	
8	CF0FLL	
7	RF7FLL	Receive FIFO Buffer Full Status Flag 0: Receive FIFO buffer x is not full. 1: Receive FIFO buffer x is full. (x = 0 to 7)
6	RF6FLL	
5	RF5FLL	
4	RF4FLL	
3	RF3FLL	
2	RF2FLL	
1	RF1FLL	
0	RF0FLL	

The RSCAN0FFSTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkFLL Flag (k = 0 to 8)

The CFkFLL flag is set to 1 when the CFFLL flag in the RSCAN0CFSTSk register is set to 1 (the transmit/receive FIFO buffer is full). When the CFFLL flag is cleared to 0 (the transmit/receive FIFO buffer is not full), the CFkFLL flag is cleared to 0.

RFxFLL Flag (x = 0 to 7)

The RFxFLL flag is set to 1 when the RFFLL flag in the RSCAN0RFSTSx register is set to 1 (the receive FIFO buffer is full). When the RFFLL flag is cleared to 0 (the receive FIFO buffer is not full), the RFxFLL flag is cleared to 0.

22.3.40 RSCAN0FMSTS — FIFO Message Lost Status Register

Access: RSCAN0FMSTS register can be read only in 32-bit units
RSCAN0FMSTSL, RSCAN0FMSTSH registers can be read only in 16-bit units
RSCAN0FMSTSLL, RSCAN0FMSTSLH, RSCAN0FMSTSHL registers can be read only in 8-bit units

Address: RSCAN0FMSTS: <RSCAN0_base> + 0240_H

RSCAN0FMSTSL: <RSCAN0_base> + 0240_H,
RSCAN0FMSTSH: <RSCAN0_base> + 0242_H

RSCAN0FMSTSLL: <RSCAN0_base> + 0240_H,
RSCAN0FMSTSLH: <RSCAN0_base> + 0241_H,
RSCAN0FMSTSHL: <RSCAN0_base> + 0242_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF8MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7MLT	CF6MLT	CF5MLT	CF4MLT	CF3MLT	CF2MLT	CF1MLT	CF0MLT	RF7MLT	RF6MLT	RF5MLT	RF4MLT	RF3MLT	RF2MLT	RF1MLT	RF0MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.56 RSCAN0FMSTS Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	These bits are read as the value after reset.
16	CF8MLT	Transmit/Receive FIFO Buffer Message Lost Status Flag 0: No transmit/receive FIFO buffer k message is lost. 1: A transmit/receive FIFO buffer k message is lost. (k = 0 to 8)
15	CF7MLT	
14	CF6MLT	
13	CF5MLT	
12	CF4MLT	
11	CF3MLT	
10	CF2MLT	
9	CF1MLT	
8	CF0MLT	
7	RF7MLT	Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost. (x = 0 to 7)
6	RF6MLT	
5	RF5MLT	
4	RF4MLT	
3	RF3MLT	
2	RF2MLT	
1	RF1MLT	
0	RF0MLT	

The RSCAN0FMSTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkMLT Flag (k = 0 to 8)

The CFkMLT flag is set to 1 when the CFMLT flag in the RSCAN0CFSTSk register is set to 1 (a transmit/receive FIFO message is lost). When the CFMLT flag is cleared to 0, the CFkMLT flag is cleared to 0.

RFxMLT Flag (x = 0 to 7)

The RFxMLT flag is set to 1 when the RFMLT flag in the RSCAN0RFSTSx register is set to 1 (a receive FIFO message is lost). When the RFMLT flag is cleared to 0, the RFxMLT flag is cleared to 0.

22.3.41 RSCAN0RFISTS — Receive FIFO Buffer Interrupt Flag Status Register

Access: RSCAN0RFISTS register can be read only in 32-bit units
RSCAN0RFISTSL register is a read-only register that can be read only in 16-bit units
RSCAN0RFISTSLL register is a read-only register that can be read only in 8-bit units

Address: RSCAN0RFISTS: <RSCAN0_base> + 0244_H

RSCAN0RFISTSL: <RSCAN0_base> + 0244_H

RSCAN0RFISTSLL: <RSCAN0_base> + 0244_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RF7IF	RF6IF	RF5IF	RF4IF	RF3IF	RF2IF	RF1IF	RF0IF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.57 RSCAN0RFISTS Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset.
7	RF7IF	Receive FIFO Buffer Interrupt Request Status Flag 0: No receive FIFO buffer x interrupt request is present. 1: A receive FIFO buffer x interrupt request is present. (x = 0 to 7)
6	RF6IF	
5	RF5IF	
4	RF4IF	
3	RF3IF	
2	RF2IF	
1	RF1IF	
0	RF0IF	

The RSCAN0RFISTS register is cleared to $0000\ 0000_H$ in global reset mode.

RFxIF Flag (x = 0 to 7)

The RFxIF flag is set to 1 when the RFIF flag in the RSCAN0RFSTSx register is set to 1 (a receive FIFO interrupt request is present). When the RFIF flag is cleared to 0, the RFxIF flag is cleared to 0.

22.3.42 RSCAN0CFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register

Access: RSCAN0CFRISTS register can be read only in 32-bit units
RSCAN0CFRISTSL register can be read only in 16-bit units
RSCAN0CFRISTSLL, RSCAN0CFRISTSLH registers can be read only in 8-bit units

Address: RSCAN0CFRISTS: <RSCAN0_base> + 0248_H

RSCAN0CFRISTSL: <RSCAN0_base> + 0248_H

RSCAN0CFRISTSLL: <RSCAN0_base> + 0248_H,
RSCAN0CFRISTSLH: <RSCAN0_base> + 0249_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CF8RXIF	CF7RXIF	CF6RXIF	CF5RXIF	CF4RXIF	CF3RXIF	CF2RXIF	CF1RXIF	CF0RXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.58 RSCAN0CFRISTS Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are read as the value after reset.
8	CF8RXIF	Transmit/Receive FIFO Buffer Receive Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k receive interrupt request is present. 1: A transmit/receive FIFO buffer k receive interrupt request is present. (k = 0 to 8)
7	CF7RXIF	
6	CF6RXIF	
5	CF5RXIF	
4	CF4RXIF	
3	CF3RXIF	
2	CF2RXIF	
1	CF1RXIF	
0	CF0RXIF	

The RSCAN0CFRISTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkRXIF Flag (k = 0 to 8)

The CFkRXIF flag is set to 1 when the CFRXIF flag in the RSCAN0CFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). When the CFRXIF flag is cleared to 0, the CFkRXIF flag is cleared to 0.

22.3.43 RSCAN0CFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register

Access: RSCAN0CFTISTS register can be read only in 32-bit units
RSCAN0CFTISTSL register can be read only in 16-bit units
RSCAN0CFTISTSLL, RSCAN0CFTISTSLH registers can be the read-only in 8-bit units

Address: RSCAN0CFTISTS: <RSCAN0_base> + $024C_H$

RSCAN0CFTISTSL: <RSCAN0_base> + $024C_H$

RSCAN0CFTISTSLL: <RSCAN0_base> + $024C_H$,
RSCAN0CFTISTSLH: <RSCAN0_base> + $024D_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CF8TXIF	CF7TXIF	CF6TXIF	CF5TXIF	CF4TXIF	CF3TXIF	CF2TXIF	CF1TXIF	CF0TXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.59 RSCAN0CFTISTS Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are read as the value after reset.
8	CF8TXIF	Transmit/Receive FIFO Buffer Transmit Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k transmit interrupt request is present. 1: A transmit/receive FIFO buffer k transmit interrupt request is present. (k = 0 to 8)
7	CF7TXIF	
6	CF6TXIF	
5	CF5TXIF	
4	CF4TXIF	
3	CF3TXIF	
2	CF2TXIF	
1	CF1TXIF	
0	CF0TXIF	

The RSCAN0CFTISTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkTXIF Flag (k = 0 to 8)

The CFkTXIF flag is set to 1 when the CFTXIF flag in the RSCAN0CFSTSk register is set to 1 (a transmit/receive FIFO transmit interrupt request is present). When the CFTXIF flag is cleared to 0, the CFkTXIF flag is cleared to 0.

22.3.44 RSCAN0TMCp — Transmit Buffer Control Register (p = 0 to 47)

Access: RSCAN0TMCp register can be read/written in 8-bit units

Address: RSCAN0TMCp: <RSCAN0_base> + 0250_H + (01_H × p)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W*1	R/W*1

Note 1. The only effective value for writing to this bit is 1, which sets the bit. Otherwise writing to the bit results in retention of its state.

Table 22.60 RSCAN0TMCp Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2	TMOM	One-Shot Transmission Enable 0: One-shot transmission is disabled. 1: One-shot transmission is enabled.
1	TMTAR	Transmit Abort Request 0: Transmit abort is not requested. 1: Transmit abort is requested.
0	TMTR	Transmit Request 0: Transmission is not requested. 1: Transmission is requested.

When the RSCAN0TMCp register meets any of the following conditions, set it to 00_H.

- The RSCAN0TMCp register corresponds to the transmit buffer number selected by the CFTML[3:0] bits in the RSCAN0CFCCk register (p = m × 16 + the value of CFTML[3:0] bits).
- The RSCAN0TMCp register corresponds to the transmit buffer allocated to the transmit queue by the TXQDC[3:0] bits in the RSCAN0TXQCCm (m = 0 to 2) register (p = (m × 16 + 15) to (m × 16 + 15 – the value of TXQDC[3:0] bits)).

Bits in the RSCAN0TMCp register are all cleared to 0 in channel reset mode. Modify the RSCAN0TMCp register in channel communication mode or channel halt mode.

TMOM Bit

Setting this bit to 1 enables one-shot transmission. When transmission fails, retransmission defined in the CAN protocol is not performed.

Modify the TMOM bit when the TMTRM flag in the RSCAN0TMSTSp register is set to 0. Set the TMOM bit to 1 together with the TMTR bit.

TMTAR Bit

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or one that will be transmitted next cannot be aborted.

The TMTAR bit can be set to 1 when TMTR bit is 1.

The TMTAR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration-lost has been detected.

If this bit becomes 0 at the same time as the program writes 1 to this bit, this bit becomes 0.

TMTR Bit

Setting this bit to 1 transmits the message stored in the transmit buffer.

The TMTR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed after the TMTAR bit was set to 1.
- An error or arbitration-lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the value of TMTRF[1:0] in the RSCAN0TMSTSp register is 00_B.

22.3.45 RSCAN0TMSTSp — Transmit Buffer Status Register (p = 0 to 47)

Access: RSCAN0TMSTSp register can be read/written in 8-bit units

Address: RSCAN0TMSTSp: <RSCAN0_base> + $02D0_H$ + (01_H × p)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R

Table 22.61 RSCAN0TMSTSp Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
4	TMTARM	Transmit Buffer Transmit Abort Request Status Flag 0: No transmit abort request is present. 1: A transmit abort request is present.
3	TMTRM	Transmit Buffer Transmit Request Status Flag 0: No transmit request is present. 1: A transmit request is present.
2, 1	TMTRF[1:0]	Transmit Buffer Transmit Result Status Flag b2 b1 0 0: Transmission is in progress or no transmit request is present. 0 1: Transmit abort has been completed. 1 0: Transmission has been completed (without transmit abort request). 1 1: Transmission has been completed (with transmit abort request).
0	TMTSTS	Transmit Buffer Transmit Status Flag 0: Transmission is not in progress. 1: Transmission is in progress.

The RSCAN0TMSTSp register is cleared to all 0 in channel reset mode.

TMTARM Flag

The TMTARM flag is set to 1 when the TMTAR bit in the RSCAN0TMCp register is set to 1.

The TMTARM flag is set to 0 when the TMTAR bit in the RSCAN0TMCp register is set to 0.

TMTRM Flag

The TMTRM flag is set to 1 when the TMTR bit in the RSCAN0TMCp register is set to 1.

The TMTRM flag is set to 0 when the TMTR bit in the RSCAN0TMCp register is set to 0.

TMTRF[1:0] Flag

This flag indicates the result of transmission from the transmit buffer.

00_B: Transmission is in progress or no transmit request is present.

01_B: Transmission from the transmit buffer was aborted.

10_B: Transmission has been completed with the TMTAR bit in the RSCAN0TMCp register set to 0 (transmit abort is not requested).

11_B: Transmission has been completed with the TMTAR bit in the RSCAN0TMCp register set to 1 (transmit abort is requested).

Write 00_B to the TMTRF[1:0] flag in channel communication mode or channel halt mode. Do not write any value other than 00_B to this flag.

TMTSTS Flag

This flag is set to 1 when transmission from the transmit buffer starts, and is cleared to 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

22.3.46 RSCAN0TMTRSTSy — Transmit Buffer Transmit Request Status Register (y = 0, 1)

Access: RSCAN0TMTRSTSy register can be read only in 32-bit units
RSCAN0TMTRSTSyL, RSCAN0TMTRSTSyH registers can be read only in 16-bit units
RSCAN0TMTRSTSyLL, RSCAN0TMTRSTSyLH, RSCAN0TMTRSTSyHL, RSCAN0TMTRSTSyHH registers can be read only in 8-bit units

Address: RSCAN0TMTRSTSy: <RSCAN0_base> + 0350_H + (04_H × y)

RSCAN0TMTRSTSyL: <RSCAN0_base> + 0350_H + (04_H × y),
RSCAN0TMTRSTSyH: <RSCAN0_base> + 0352_H + (04_H × y)

RSCAN0TMTRSTSyLL: <RSCAN0_base> + 0350_H + (04_H × y),
RSCAN0TMTRSTSyLH: <RSCAN0_base> + 0351_H + (04_H × y),
RSCAN0TMTRSTSyHL: <RSCAN0_base> + 0352_H + (04_H × y),
RSCAN0TMTRSTSyHH: <RSCAN0_base> + 0353_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTRSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTRSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.62 RSCAN0TMTRSTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmit request is present. 1: A transmit request is present.
15 to 0	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit request is present. 1: A transmit request is present.

TMTRSTSp Flags (p = 0 to 47)

These flags indicate the status of the TMTR bit in the RSCAN0TMCp register.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag is set to 1.

The corresponding TMTRSTSp flag is cleared to 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

Table 22.63 shows the bit assignment.

Table 22.63 TMTRSTSp Bit Assignment (1/2)

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15

Table 22.63 TMTRSTSp Bit Assignment (2/2)

Bit	Channel	Transmit Buffer Number
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

22.3.47 RSCAN0TMTARSTSy — Transmit Buffer Transmit Abort Request Status Register (y = 0, 1)

Access: RSCAN0TMTARSTSy register can be read only in 32-bit units
RSCAN0TMTARSTSyL, RSCAN0TMTARSTSyH registers can be read only in 16-bit units
RSCAN0TMTARSTSyLL, RSCAN0TMTARSTSyLH, RSCAN0TMTARSTSyHL, RSCAN0TMTARSTSyHH registers can be read only in 8-bit units

Address: RSCAN0TMTARSTSy: <RSCAN0_base> + 0360_H + (04_H × y)

RSCAN0TMTARSTSyL: <RSCAN0_base> + 0360_H + (04_H × y),
RSCAN0TMTARSTSyH: <RSCAN0_base> + 0362_H + (04_H × y)

RSCAN0TMTARSTSyLL: <RSCAN0_base> + 0360_H + (04_H × y),
RSCAN0TMTARSTSyLH: <RSCAN0_base> + 0361_H + (04_H × y),
RSCAN0TMTARSTSyHL: <RSCAN0_base> + 0362_H + (04_H × y),
RSCAN0TMTARSTSyHH: <RSCAN0_base> + 0363_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTARSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTARSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.64 RSCAN0TMTARSTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmit abort request is present. 1: A transmit abort request is present.
15 to 0	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit abort request is present. 1: A transmit abort request is present.

TMTARSTSp Flags (p = 0 to 47)

These flags indicate the status of the TMTAR bit in the RSCAN0TMCp register.

When the TMTAR bit is set to 1 (transmit abort is requested), the corresponding TMTARSTSp flag is set to 1.

The corresponding TMTARSTSp flag is cleared to 0 when the TMTAR bit is set to 0 (transmit abort is not requested) or in channel reset mode.

Table 22.65 shows the bit assignment.

Table 22.65 TMTARSTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
.	.	.
.	.	.

Table 22.65 TMTARSTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

22.3.48 RSCAN0TMTCSTSy — Transmit Buffer Transmit Complete Status Register (y = 0, 1)

Access: RSCAN0TMTCSTSy register can be read only in 32-bit units
RSCAN0TMTCSTSyL, RSCAN0TMTCSTSyH registers can be read only in 16-bit units
RSCAN0TMTCSTSyLL, RSCAN0TMTCSTSyLH, RSCAN0TMTCSTSyHL, RSCAN0TMTCSTSyHH registers can be read only in 8-bit units

Address: RSCAN0TMTCSTSy: <RSCAN0_base> + 0370_H + (04_H × y)

RSCAN0TMTCSTSyL: <RSCAN0_base> + 0370_H + (04_H × y),
RSCAN0TMTCSTSyH: <RSCAN0_base> + 0372_H + (04_H × y)

RSCAN0TMTCSTSyLL: <RSCAN0_base> + 0370_H + (04_H × y),
RSCAN0TMTCSTSyLH: <RSCAN0_base> + 0371_H + (04_H × y),
RSCAN0TMTCSTSyHL: <RSCAN0_base> + 0372_H + (04_H × y),
RSCAN0TMTCSTSyHH: <RSCAN0_base> + 0273_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTCSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTCSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.66 RSCAN0TMTCSTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission has not been completed. 1: Transmission has been completed.
15 to 0	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission has not been completed. 1: Transmission has been completed.

TMTCSTSp Flags (p = 0 to 47)

When the TMTRF[1:0] flag in the RSCAN0TMSTSp register is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)), the corresponding TMTCSTSp flag is set to 1.

A TMTCSTSp flag is cleared to 0 when the corresponding TMTRF[1:0] flag is set to 00_B or in channel reset mode.

Table 22.67 shows the bit assignment.

Table 22.67 TMTCSTSp Bit Assignment (1/2)

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15

Table 22.67 TMTCSTSp Bit Assignment (2/2)

Bit	Channel	Transmit Buffer Number
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

22.3.49 RSCAN0TMTASTSy — Transmit Buffer Transmit Abort Status Register (y = 0, 1)

Access: RSCAN0TMTASTSy register can be read only in 32-bit units
RSCAN0TMTASTSyL, RSCAN0TMTASTSyH registers can be read only in 16-bit units
RSCAN0TMTASTSyLL, RSCAN0TMTASTSyLH, RSCAN0TMTASTSyHL, RSCAN0TMTASTSyHH registers can be read only in 8-bit units

Address: RSCAN0TMTASTSy: <RSCAN0_base> + 0380_H + (04_H × y)

RSCAN0TMTASTSyL: <RSCAN0_base> + 0380_H + (04_H × y),
RSCAN0TMTASTSyH: <RSCAN0_base> + 0382_H + (04_H × y)

RSCAN0TMTASTSyLL: <RSCAN0_base> + 0380_H + (04_H × y),
RSCAN0TMTASTSyLH: <RSCAN0_base> + 0381_H + (04_H × y),
RSCAN0TMTASTSyHL: <RSCAN0_base> + 0382_H + (04_H × y),
RSCAN0TMTASTSyHH: <RSCAN0_base> + 0383_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTASTSp (p = y × 32 + 31 to y × 32 × 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTASTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.68 RSCAN0TMTASTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission is not aborted 1: Transmission is aborted
15 to 0	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission is not aborted. 1: Transmission is aborted.

TMTASTSp Flags (p = 0 to 47)

When the TMTRF[1:0] flag in the RSCAN0TMSTSp register is set to 01_B (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.

A TMTASTSp flag is cleared to 0 when the corresponding TMTRF[1:0] flag is set to 00_B or in channel reset mode.

Table 22.69 shows the bit assignment.

Table 22.69 TMTASTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0

Table 22.69 TMTASTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

22.3.50 RSCAN0TMIECy — Transmit Buffer Interrupt Enable Configuration Register (y = 0, 1)

Access: RSCAN0TMIECy register can be read/written in 32-bit units
RSCAN0TMIECyL, RSCAN0TMIECyH registers can be read/written in 16-bit units
RSCAN0TMIECyLL, RSCAN0TMIECyLH, RSCAN0TMIECyHL, RSCAN0TMIECyHH registers can be read/written in 8-bit units

Address: RSCAN0TMIECy: <RSCAN0_base> + 0390_H + (04_H × y)

RSCAN0TMIECyL: <RSCAN0_base> + 0390_H + (04_H × y),
RSCAN0TMIECyH: <RSCAN0_base> + 0392_H + (04_H × y)

RSCAN0TMIECyLL: <RSCAN0_base> + 0390_H + (04_H × y),
RSCAN0TMIECyLH: <RSCAN0_base> + 0391_H + (04_H × y),
RSCAN0TMIECyHL: <RSCAN0_base> + 0392_H + (04_H × y),
RSCAN0TMIECyHH: <RSCAN0_base> + 0393_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIEp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMIEp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.70 RSCAN0TMIECy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMIEp	Transmit Buffer Interrupt Enable p (p = y × 32 + 31 to y × 32 + 16) 0: Transmit buffer interrupt is disabled 1: Transmit buffer interrupt is enabled
15 to 0	TMIEp	Transmit Buffer Interrupt Enable p (p = y × 32 + 15 to y × 32 + 0) 0: Transmit buffer interrupt is disabled. 1: Transmit buffer interrupt is enabled.

TMIEp Bits (p = 0 to 47)

When any of these bits is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify these bits when the TMTRM flag in the corresponding RSCAN0TMSTSp register is 0 (no transmit request is present).

Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers or transmit buffers allocated to the transmit queue.

Table 22.71 shows the bit assignment.

Table 22.71 TMIEp Bit Assignment (1/2)

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
.	.	.
.	.	.

Table 22.71 TMIEp Bit Assignment (2/2)

Bit	Channel	Transmit Buffer Number
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

22.3.51 RSCAN0TMIDp — Transmit Buffer ID Register (p = 0 to 47)

Access: RSCAN0TMIDp register can be read/written in 32-bit units
RSCAN0TMIDpL, RSCAN0TMIDpH registers can be read/written in 16-bit units
RSCAN0TMIDpLL, RSCAN0TMIDpLH, RSCAN0TMIDpHL, RSCAN0TMIDpHH registers can be read/written in 8-bit units

Address: RSCAN0TMIDp: <RSCAN0_base> + 1000_H + (10_H × p)

RSCAN0TMIDpL: <RSCAN0_base> + 1000_H + (10_H × p),
RSCAN0TMIDpH: <RSCAN0_base> + 1002_H + (10_H × p)

RSCAN0TMIDpLL: <RSCAN0_base> + 1000_H + (10_H × p),
RSCAN0TMIDpLH: <RSCAN0_base> + 1001_H + (10_H × p),
RSCAN0TMIDpHL: <RSCAN0_base> + 1002_H + (10_H × p),
RSCAN0TMIDpHH: <RSCAN0_base> + 1003_H + (10_H × p)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIDE	TMRTR	THLEN	TMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.72 RSCAN0TMIDp Register Contents

Bit Position	Bit Name	Function
31	TMIDE	Transmit Buffer IDE 0: Standard ID 1: Extended ID
30	TMRTR	Transmit Buffer RTR 0: Data frame 1: Remote frame
29	THLEN	Transmit History Data Store Enable 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	TMID[28:0]	Transmit Buffer ID Data Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write data to this register. If this register is allocated to the transmit queue, only write data to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMIDE Bit

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

TMRTR Bit

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

THLEN Bit

With this bit set to 1, the transmit history data of the message transmitted (label information and the number and type) are stored in the transmit history buffer after transmission is completed.

TMID[28:0] Bits

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

22.3.52 RSCAN0TMPTRp — Transmit Buffer Pointer Register (p= 0 to 47)

Access: RSCAN0TMPTRp register can be read/written in 32-bit units
RSCAN0TMPTRpH register can be read/written in 16-bit units
RSCAN0TMPTRpHL, RSCAN0TMPTRpHH registers can be read/written in 8-bit units

Address: RSCAN0TMPTRp: <RSCAN0_base> + 1004_H + (10_H × p)

RSCAN0TMPTRpH: <RSCAN0_base> + 1006_H + (10_H × p)

RSCAN0TMPTRpHL: <RSCAN0_base> + 1006_H + (10_H × p),
RSCAN0TMPTRpHH: <RSCAN0_base> + 1007_H + (10_H × p)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDLC[3:0]				—	—	—	—	TMPTR[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.73 RSCAN0TMPTRp Register Contents

Bit Position	Bit Name	Function
31 to 28	TMDLC[3:0]	Transmit Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 24	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
23 to 16	TMPTR[7:0]	Transmit Buffer Label Data Set the label information to be stored in the transmit history buffer.
15 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMDLC[3:0] Bits

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMRTR bit in the RSCAN0TMIDp register is set to 0 (data frame). If the data length is set to 9 bytes or more, the transmit data is 8 bytes long.

When the TMRTR bit is set to 1 (remote frame), set the data length of messages to be requested.

TMPTR[7:0] Bits

When message transmission has been completed, the TMPTR[7:0] value is stored in the transmit history buffer.

22.3.53 RSCAN0TMDF0p — Transmit Buffer Data Field 0 Register (p = 0 to 47)

Access: RSCAN0TMDF0p register can be read/written in 32-bit units
RSCAN0TMDF0pL, RSCAN0TMDF0pH registers can be read/written in 16-bit units
RSCAN0TMDF0pLL, RSCAN0TMDF0pLH, RSCAN0TMDF0pHL, RSCAN0TMDF0pHH registers can be read/written in 8-bit units

Address: RSCAN0TMDF0p: <RSCAN0_base> + 1008_H + (10_H × p)

RSCAN0TMDF0pL: <RSCAN0_base> + 1008_H + (10_H × p),
RSCAN0TMDF0pH: <RSCAN0_base> + $100A_H$ + (10_H × p)

RSCAN0TMDF0pLL: <RSCAN0_base> + 1008_H + (10_H × p),
RSCAN0TMDF0pLH: <RSCAN0_base> + 1009_H + (10_H × p),
RSCAN0TMDF0pHL: <RSCAN0_base> + $100A_H$ + (10_H × p),
RSCAN0TMDF0pHH: <RSCAN0_base> + $100B_H$ + (10_H × p)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB3[7:0]								TMDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB1[7:0]								TMDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.74 RSCAN0TMDF0p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB3[7:0]	Transmit Buffer Data Byte 3 Transmit Buffer Data Byte 2 Transmit Buffer Data Byte 1 Transmit Buffer Data Byte 0 Set the transmit buffer data.
23 to 16	TMDB2[7:0]	
15 to 8	TMDB1[7:0]	
7 to 0	TMDB0[7:0]	

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

22.3.54 RSCAN0TMDF1p — Transmit Buffer Data Field 1 Register (p = 0 to 47)

Access: RSCAN0TMDF1p register can be read/written in 32-bit units
RSCAN0TMDF1pL, RSCAN0TMDF1pH registers can be read/written in 16-bit units
RSCAN0TMDF1pLL, RSCAN0TMDF1pLH, RSCAN0TMDF1pHL, RSCAN0TMDF1pHH registers can be read/written in 8-bit units

Address: RSCAN0TMDF1p: <RSCAN0_base> + $100C_H$ + (10_H × p)

RSCAN0TMDF1pL: <RSCAN0_base> + $100C_H$ + (10_H × p),
RSCAN0TMDF1pH: <RSCAN0_base> + $100E_H$ + (10_H × p)

RSCAN0TMDF1pLL: <RSCAN0_base> + $100C_H$ + (10_H × p),
RSCAN0TMDF1pLH: <RSCAN0_base> + $100D_H$ + (10_H × p),
RSCAN0TMDF1pHL: <RSCAN0_base> + $100E_H$ + (10_H × p),
RSCAN0TMDF1pHH: <RSCAN0_base> + $100F_H$ + (10_H × p)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB7[7:0]								TMDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB5[7:0]								TMDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.75 RSCAN0TMDF1p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB7[7:0]	Transmit Buffer Data Byte 7
23 to 16	TMDB6[7:0]	Transmit Buffer Data Byte 6
15 to 8	TMDB5[7:0]	Transmit Buffer Data Byte 5
7 to 0	TMDB4[7:0]	Transmit Buffer Data Byte 4 Set the transmit buffer data.

Modify this register when the TMTRM bit in the corresponding RSCAN0TMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

22.3.55 RSCAN0TXQCCm — Transmit Queue Configuration and Control Register (m = 0 to 2)

Access: RSCAN0TXQCCm register can be read/written in 32-bit units
RSCAN0TXQCCmL register can be read/written in 16-bit units
RSCAN0TXQCCmLL, RSCAN0TXQCCmLH registers can be read/written in 8-bit units

Address: RSCAN0TXQCCm: <RSCAN0_base> + $03A0_H$ + (04_H × m)

RSCAN0TXQCCmL: <RSCAN0_base> + $03A0_H$ + (04_H × m)

RSCAN0TXQCCmLL: <RSCAN0_base> + $03A0_H$ + (04_H × m),
RSCAN0TXQCCmLH: <RSCAN0_base> + $03A1_H$ + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TXQIM	TXQIE	TXQDC[3:0]				—	—	—	—	—	—	—	TXQE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 22.76 RSCAN0TXQCCm Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
13	TXQIM	Transmit Queue Interrupt Source Select 0: When the transmit queue becomes empty upon completion of message transmission, a transmit queue interrupt request is generated. 1: A transmit queue interrupt request is generated each time a message has been transmitted.
12	TXQIE	Transmit Queue Interrupt Enable 0: Transmit queue interrupt is disabled. 1: Transmit queue interrupt is enabled.
11 to 8	TXQDC[3:0]	Transmit Queue Depth Configuration Setting these bits to g (g = 2 to 15) makes the (g + 1) transmit queue available. Setting these bits to 0 disables the transmit queue. Setting these bits to 1 is prohibited.
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	TXQE	Transmit Queue Enable 0: The transmit queue is not used. 1: The transmit queue is used.

TXQIM Bit

This bit is used to select a transmit queue interrupt source. Modify this bit in channel reset mode.

TXQIE Bit

When the TXQIE bit is set to 1 and the source selected by the TXQIM bit occurs, an interrupt request is generated.

Set the TXQE bit to 0 before modifying the TXQIE bit.

TXQDC[3:0] Bits

These bits are used to specify the number of transmit buffers to be allocated to the transmit queues. Transmit buffers are allocated to transmit queues in descending order of buffer number, that is, from (m × 16 + 15) to (m × 16 + 0). For examples of how buffer allocation is done, see **Figure 22.9**. Modify these bits only in channel reset mode.

TXQE Bit

Setting this bit to 1 makes the transmit queue available. Modify this bit in channel communication mode or channel halt mode. This bit is cleared to 0 in channel reset mode.

Before setting the TXQE bit to 1, set the TXQDC[3:0] bits to 0010_B or more.

22.3.56 RSCAN0TXQSTSm — Transmit Queue Status Register (m = 0 to 2)

Access: RSCAN0TXQSTSm register can be read/written in 32-bit units
RSCAN0TXQSTSmL register can be read/written in 16-bit units
RSCAN0TXQSTSmLL register can be read/written in 8-bit units

Address: RSCAN0TXQSTSm: <RSCAN0_base> + $03C0_H$ + (04_H × m)

RSCAN0TXQSTSmL: <RSCAN0_base> + $03C0_H$ + (04_H × m)

RSCAN0TXQSTSmLL: <RSCAN0_base> + $03C0_H$ + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQIF	TXQFLL	TXQEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 22.77 RSCAN0TXQSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 8	Reserved	When read, an undefined value is returned. When writing these bits, write “0”.
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	TXQIF	Transmit Queue Interrupt Request Flag 0: No transmit queue interrupt request is present. 1: A transmit queue interrupt request is present.
1	TXQFLL	Transmit Queue Full Status Flag 0: The transmit queue is not full. 1: The transmit queue is full.
0	TXQEMP	Transmit Queue Empty Status Flag 0: The transmit queue contains messages. 1: The transmit queue contains no message (transmit queue empty).

TXQIF Flag

The TXQIF flag is set to 1 when the event specified by the TXQIM bit in the RSCAN0TXQCCm register has occurred.

The TXQIF flag is cleared to 0 in channel reset mode or by writing 0 to this flag. This flag is not cleared to 0 by setting the TXQE bit in the RSCAN0TXQCCm register to 0 (the transmit queue is not used).

TXQFLL Flag

The TXQFLL flag is set to 1 when the number of messages set for the transmit queue matches the transmit queue depth set by the TXQDC[3:0] bits in the RSCAN0TXQCCm register.

This flag is cleared to 0 in any of the following cases.

- The number of messages set for the transmit queue is smaller than the transmit queue depth set by the TXQDC[3:0] bits.
- In channel reset mode

TXQEMP Flag

The TXQEMP flag is cleared to 0 when even a single message is set for the transmit queue.

This flag is set to 1 in any of the following cases.

- The TXQE bit is set to 0 (the transmit queue is not used).
- The transmit queue becomes empty.
- In channel reset mode

22.3.57 RSCAN0TXQPCTRm — Transmit Queue Pointer Control Register (m = 0 to 2)

Access: RSCAN0TXQPCTRm registers can only be written in 32-bit units
RSCAN0TXQPCTRmL register is a write-only register that can only be written in 16-bit units
RSCAN0TXQPCTRmLL register is a write-only register that can only be written in 8-bit units

Address: RSCAN0TXQPCTRm: <RSCAN0_base> + $03E0_H$ + (04_H × m)

RSCAN0TXQPCTRmL: <RSCAN0_base> + $03E0_H$ + (04_H × m)

RSCAN0TXQPCTRmLL: <RSCAN0_base> + $03E0_H$ + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 22.78 RSCAN0TXQPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be the value after reset.
7 to 0	TXQPC[7:0]	Transmit Queue Pointer Control Writing FF_H to these bits moves the write pointer of the transmit queue to the next queue buffer.

TXQPC[7:0] Bits

Writing FF_H to the TXQPC[7:0] bits moves the write pointer to the next transmit queue buffer and generates a transmit request of the message. Write transmit messages to the RSCAN0TMIDp, RSCAN0TMPTRp, RSCAN0TMDF0p, and RSCAN0TMDF1p registers (p = 15, 31, and 47) before writing FF_H to the TXQPC[7:0] bits.

When writing FF_H to these bits, make sure that the TXQE bit in the RSCAN0TXQCCm register is set to 1 (the transmit queue is used) and the TXQFLL flag in the RSCAN0TXQSTSm register is 0 (he transmit queue is not full).

22.3.58 RSCAN0THLCCm — Transmit History Configuration and Control Register (m = 0 to 2)

Access: RSCAN0THLCCm register can be read/written in 32-bit units
RSCAN0THLCCmL register can be read/written in 16-bit units
RSCAN0THLCCmLL, RSCAN0THLCCmLH registers can be read/written in 8-bit units

Address: RSCAN0THLCCm: <RSCAN0_base> + 0400_H + (04_H × m)

RSCAN0THLCCmL: <RSCAN0_base> + 0400_H + (04_H × m)

RSCAN0THLCCmLL: <RSCAN0_base> + 0400_H + (04_H × m),
RSCAN0THLCCmLH: <RSCAN0_base> + 0401_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 22.79 RSCAN0THLCCm Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
10	THLDTE	Transmit History Target Buffer Select 0: Entry from transmit/receive FIFO buffers and transmit queue 1: Entry from transmit buffers, transmit/receive FIFO buffers, and transmit queue
9	THLIM	Transmit History Interrupt Source Select 0: When 12 sets of data have been stored in the transmit history buffer 1: When a single set of transmit history data has been stored
8	THLIE	Transmit History Interrupt Enable 0: Transmit history interrupt is disabled. 1: Transmit history interrupt is enabled.
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	THLE	Transmit History Buffer Enable 0: Transmit history buffer is not used. 1: Transmit history buffer is used.

THLDTE Bit

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers and the transmit queue is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers, transmit/receive FIFO buffers, and the transmit queue is stored in the transmit history buffer.

Modify this bit only in channel reset mode.

THLIM Bit

This bit is used to select a transmit history interrupt source.

Modify this bit only in channel reset mode.

THLIE Bit

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit only when the THLE bit set to 0.

THLE Bit

Setting this bit to 1 makes the transmit history buffer available. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer.

Modify this bit in channel communication mode or channel halt mode.

This bit is cleared to 0 in cahnnel reset mode.

22.3.59 RSCAN0THLSTSm — Transmit History Status Register (m = 0 to 2)

Access: RSCAN0THLSTSm register can be read/written in 32-bit units
RSCAN0THLSTSmL register can be read/written in 16-bit units
RSCAN0THLSTSmLL register can be read/written in 8-bit units
RSCAN0THLSTSmLH register is a read-only register that can be read in 8-bit units

Address: RSCAN0THLSTSm: <RSCAN0_base> + 0420_H + (04_H × m)

RSCAN0THLSTSmL: <RSCAN0_base> + 0420_H + (04_H × m)

RSCAN0THLSTSmLL: <RSCAN0_base> + 0420_H + (04_H × m),
RSCAN0THLSTSmLH: <RSCAN0_base> + 0421_H + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THLMC[4:0]					—	—	—	—	THLIF	THLELT	THLFLL	THLEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 22.80 RSCAN0THLSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 8	THLMC[4:0]	Transmit History Buffer Unread Data Counter These bits indicate the number of unread data sets stored in the transmit history buffer.
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	THLIF	Transmit History Interrupt Request Flag 0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present.
2	THLELT	Transmit History Buffer Overflow Flag 0: Transmit history buffer overflow has not occurred. 1: Transmit history buffer overflow has occurred.
1	THLFLL	Transmit history Buffer Full Status Flag 0: Transmit history buffer is not full. 1: Transmit history buffer is full.
0	THLEMP	Transmit History Buffer Empty Status Flag 0: Transmit history buffer contains unread data. 1: Transmit history buffer contains no unread data (buffer empty).

THLMC[4:0] Bits

These bits indicate the number of unread data sets stored in the transmit history buffer.

These bits are cleared to 0 in cahnnel reset mode.

THLIF Flag

The THLIF flag is set to 1 when the interrupt source specified with the THLIM bit in the RSCAN0THLCCm register occurs.

This flag is cleared to 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLELT Flag

The THLELT flag is set to 1 when an attempt is made to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded. This flag becomes 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLFLL Flag

The THLFLL flag is set to 1 when 16 data sets have been stored in the transmit history buffer, and is cleared to 0 when the number of data sets stored in the transmit history buffer has decreased to less than 16. This bit is also cleared to 0 in channel reset mode or when the THLE bit in the RSCAN0THLCCm register is set to 0 (transmit history buffer is not used).

THLEMP Flag

The THLEMP flag is cleared to 0 when even a single set of transmit history data has been stored in the transmit history buffer.

This flag is set to 1 when all the data in the transmit history buffer has been read. This flag is also set to 1 in channel reset mode or when the THLE bit in the RSCAN0THLCCm register is set to 0 (transmit history buffer is not used).

NOTE

To clear THLIF or THLELT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

22.3.60 RSCAN0THLACCm — Transmit History Access Register (m = 0 to 2)

Access: RSCAN0THLACCm register can be read only in 32-bit units
RSCAN0THLACCmL register is a read-only register that can be read only in 16-bit units
RSCAN0THLACCmLL, RSCAN0THLACCmLH registers are the read-only in 8-bit units

Address: RSCAN0THLACCm: <RSCAN0_base> + 1800_H + (04_H × m)

RSCAN0THLACCmL: <RSCAN0_base> + 1800_H + (04_H × m)

RSCAN0THLACCmLL: <RSCAN0_base> + 1800_H + (04_H × m),
RSCAN0THLACCmLH: <RSCAN0_base> + 1801_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TID[7:0]								—	BN[3:0]				BT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22.81 RSCAN0THLACCm Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is returned.
15 to 8	TID[7:0]	Label Data The label information of stored data can be read.
7	Reserved	When read, the value after reset is returned.
6 to 3	BN[3:0]	Buffer Number Data The buffer number of transmit source (transmit buffer, transmit/receive FIFO or transmit queue) can be read.
2 to 0	BT[2:0]	Buffer Type Data b2 b1 b0 0 0 1: Transmit buffer 0 1 0: Transmit/receive FIFO buffer 1 0 0: Transmit queue

TID[7:0] Bits

These bits indicate the label information of transmit history data stored in the transmit history buffer.

BN[3:0] Bits

These bits indicate the transmit source buffer number in the transmit history data stored in the transmit history buffer.

BT[2:0] Bits

These bits indicate the type of the transmit source buffer in the transmit history data stored in the transmit history buffer.

22.3.61 RSCAN0THLPCTRm — Transmit History Pointer Control Register (m = 0 to 2)

Access: RSCAN0THLPCTRm register can only be written in 32-bit units
RSCAN0THLPCTRmL register is a write-only register that can only be written in 16-bit units
RSCAN0THLPCTRmLL register is a write-only register that can only be written in 8-bit units

Address: RSCAN0THLPCTRm: <RSCAN0_base> + 0440_H + (04_H × m)

RSCAN0THLPCTRmL: <RSCAN0_base> + 0440_H + (04_H × m)

RSCAN0THLPCTRmLL: <RSCAN0_base> + 0440_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	THLPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 22.82 RSCAN0THLPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write the value after reset.
7 to 0	THLPC[7:0]	Transmit History List Pointer Control Writing FF_H to these bits moves the read pointer to the next unread data in the transmit history buffer.

THLPC[7:0] Bits

When the THLPC[7:0] bits are set to FF_H, the read pointer moves to the next data in the transmit history buffer. At this time, the THLMC[4:0] (transmit history buffer unread data counter) value in the RSCAN0THLSTSm register is decremented. Write FF_H to the THLPC[7:0] bits after reading from the RSCAN0THLACCm register.

When writing FF_H to these bits, make sure that the THLE bit in the RSCAN0THLCCm register is set to 1 (transmit history buffer is used) and the THLEMP flag in the RSCAN0THLSTSm register is 0.

22.3.62 RSCAN0GTSTCFG — Global Test Configuration Register

Access: RSCAN0GTSTCFG register can be read/written in 32-bit units
RSCAN0GTSTCFGL, RSCAN0GTSTCFGH registers can be read/written in 16-bit units
RSCAN0GTSTCFGLL, RSCAN0GTSTCFGHL registers can be read/written in 8-bit units

Address: RSCAN0GTSTCFG: <RSCAN0_base> + 0468_H

RSCAN0GTSTCFGL: <RSCAN0_base> + 0468_H,
RSCAN0GTSTCFGH: <RSCAN0_base> + $046A_H$

RSCAN0GTSTCFGLL: <RSCAN0_base> + 0468_H,
RSCAN0GTSTCFGHL: <RSCAN0_base> + $046A_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	RTMPS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	C2ICBCE	C1ICBCE	C0ICBCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 22.83 RSCAN0GTSTCFG Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
22 to 16	RTMPS[6:0]	RAM Test Page Configuration Set a value within a range of page 0 (00_H) to page 29 ($1D_H$).
15 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	C2ICBCE	CAN2 Inter-channel Communication Test Enable 0: RSCAN2 inter-channel communication test is disabled 1: RSCAN2 inter-channel communication test is enabled.
1	C1ICBCE	CAN1 Inter-Channel Communication Test Enable 0: CAN1 inter-channel communication test is disabled. 1: CAN1 inter-channel communication test is enabled.
0	C0ICBCE	CAN0 Inter-Channel Communication Test Enable 0: CAN0 inter-channel communication test is disabled. 1: CAN0 inter-channel communication test is enabled.

Modify the RSCAN0GTSTCFG register only in global test mode.

RTMPS[6:0] Bits

These bits are used to set the RAM test target page number for RAM test. Set a value in the range of 00_H to $1D_H$, inclusive.

Do not access more than 160 bytes in the last page (RTMPS[6:0] = $1D_H$) during RAM test.

C2ICBCE Bit

Setting this bit to 1 enables the channel 2 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C1ICBCE Bit

Setting this bit to 1 enables the channel 1 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C0ICBCE Bit

Setting this bit to 1 enables the channel 0 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

22.3.63 RSCAN0GTSTCTR — Global Test Control Register

Access: RSCAN0GTSTCTR register can be read/written in 32-bit units
RSCAN0GTSTCTRL register can be read/written in 16-bit units
RSCAN0GTSTCTRLL register can be read/written in 8-bit units

Address: RSCAN0GTSTCTR: <RSCAN0_base> + $046C_H$

RSCAN0GTSTCTRL: <RSCAN0_base> + $046C_H$

RSCAN0GTSTCTRLL: <RSCAN0_base> + $046C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	ICBCTME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Table 22.84 RSCAN0GTSTCTR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	RTME	RAM Test Enable 0: RAM test is disabled. 1: RAM test is enabled.
1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	ICBCTME	Communication Test between Channels Enable 0: Communication test between channels disabled 1: Communication test between channels enabled

RTME Bit

Setting this bit to 1 enables the RAM test. Modify this bit only in global test mode.

This bit is cleared to 0 in global reset mode.

1. Set the GMDC[1:0] bits in the RSCAN0GCTR register to 10_B (Global test mode).
2. Set the RTME bit to 1.
3. Check that the RTME bit is set to 1.

ICBCTME Bit

When this bit is set to 1, a communication test is enabled between the channels for which the CmICBCE bit (m = 0 to 2) in the RSCAN0GTSTCFG register has been set to 1. Modify the ICBCTME bit only in global test mode.

This bit is cleared to 0 in global reset mode.

22.3.64 RSCAN0GLOCKK — Global Lock Key Register

Access: RSCAN0GLOCKK register can be written only in 32-bit units.
RSCAN0GLOCKKL register is a write-only register that can be written only in 16-bit units.

Address: RSCAN0GLOCKK: <RSCAN0_base> + $047C_H$

RSCAN0GLOCKKL: <RSCAN0_base> + $047C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1

Note 1. Writing to these bits is effective only when the RSCAN module is in global test mode.

Table 22.85 RSCAN0GLOCKK Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing these bits, write the value after reset.
15 to 0	LOCK[15:0]	Lock Key These bits are key bits to release protection of test mode.

The RSCAN0GLOCKK register releases protection of special test bits and is write only.

For the protection release data, see Section 22.10.4.2, Procedure for Releasing the Protection.

LOCK[15:0] Bits

Writing the protection release data to the LOCK[15:0] bits in succession enables writing 1 to the RTME bit in the RSCAN0GTSTCTR register.

After the protection has been released, writing to the I/O register area (<RSCAN0_base> + 0000_H to <RSCAN0_base> + $04FF_H$) of the CAN (except the RAM) enables the protection again.

Reading from the I/O register area of the CAN or reading from/wring to other areas does not enable the protection.

22.3.65 RSCAN0RPGACCr — RAM Test Page Access Register (r = 0 to 63)

Access: RSCAN0RPGACCr register can be read/written in 32-bit units
RSCAN0RPGACCrL, RSCAN0RPGACCrH registers can be read/written in 16-bit units
RSCAN0RPGACCrLL, RSCAN0RPGACCrLH, RSCAN0RPGACCrHL, RSCAN0RPGACCrHH registers can be read/written in 8-bit units

Address: RSCAN0RPGACCr: <RSCAN0_base> + 1900_H + (04_H × n)

RSCAN0RPGACCrL: <RSCAN0_base> + 1900_H + (04_H × n),
RSCAN0RPGACCrH: <RSCAN0_base> + 1902_H + (04_H × n)

RSCAN0RPGACCrLL: <RSCAN0_base> + 1900_H + (04_H × n),
RSCAN0RPGACCrLH: <RSCAN0_base> + 1901_H + (04_H × n),
RSCAN0RPGACCrHL: <RSCAN0_base> + 1902_H + (04_H × n),
RSCAN0RPGACCrHH: <RSCAN0_base> + 1903_H + (04_H × n)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.86 RSCAN0RPGACCr Register Contents

Bit Position	Bit Name	Function
31 to 0	RDTA [31:0]	RAM Data Test Access Data can be read and written in RSCAN RAM.

Modify the RSCAN0RPGACCr register in global test mode with the RTME bit in the RSCAN0GTSTCTR register set to 1 (RAM test is enabled).

The RSCAN0RPGACCr register is readable and writable when the RTME bit is set to 1.

22.4 Interrupt Sources

The RSCAN module has 11 interrupts that are grouped into global interrupts and channel interrupts.

- Global interrupts (2 sources):

  Receive FIFO interrupt

  Global error interrupt

- Channel interrupts (3 sources/channel):
  CANm transmit interrupt (m = 0 to 2)
  - CANm transmit complete interrupt
  - CANm transmit abort interrupt
  - CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode)
  - CANm transmit history interrupt
  - CANm transmit queue Interrupt

  CANm transmit/receive FIFO receive complete interrupt (in revceive mode, gateway mode)
  CANm error interrupt

When an interrupt request is generated, the corresponding interrupt request flag is set to 1 (interrupt request present). In that case, when the interrupt enable bit is set to 1 (enabling interrupts), an interrupt request is output from the RSCAN module. (Generation of interrupts also depends on the interrupt control register settings of the interrupt controller.)

Setting the interrupt request flag to 0 (no interrupt request present) or setting the interrupt enable bit to 0 (disabling interrupts) clears the current interrupt request. The current interrupt request is still output until the interrupt request flag is cleared.

Table 22.87 lists the CAN interrupt sources. **Figure 22.2** shows the CAN global interrupt block diagram. **Figure 22.3** shows the CAN channel interrupt block diagram.

Table 22.87 List of CAN Interrupt Sources (1/2)

Interrupt Source			Corresponding Interrupt Request Flag	Corresponding Interrupt Enable Bit
Global interrupts	Receive FIFO	Receive FIFO 0	RFIF in the RSCAN0RFSTS0 register	RFIE in the RSCAN0RFCC0 register
		Receive FIFO 1	RFIF in the RSCAN0RFSTS1 register	RFIE in the RSCAN0RFCC1 register
		Receive FIFO 2	RFIF in the RSCAN0RFSTS2 register	RFIE in the RSCAN0RFCC2 register
		Receive FIFO 3	RFIF in the RSCAN0RFSTS3 register	RFIE in the RSCAN0RFCC3 register
		Receive FIFO 4	RFIF in the RSCAN0RFSTS4 register	RFIE in the RSCAN0RFCC4 register
		Receive FIFO 5	RFIF in the RSCAN0RFSTS5 register	RFIE in the RSCAN0RFCC5 register
		Receive FIFO 6	RFIF in the RSCAN0RFSTS6 register	RFIE in the RSCAN0RFCC6 register
		Receive FIFO 7	RFIF in the RSCAN0RFSTS7 register	RFIE in the RSCAN0RFCC7 register
	Global error		• DEF in the RSCAN0GERFL register • MES in the RSCAN0GERFL register • THLES in the RSCAN0GERFL register	• DEIE in the RSCAN0GCTR register • MEIE in the RSCAN0GCTR register • THLEIE in the RSCAN0GCTR register

Table 22.87 List of CAN Interrupt Sources (2/2)

Interrupt Source			Corresponding Interrupt Request Flag	Corresponding Interrupt Enable Bit
Channel interrupts (m = 0 to 2)	CANm transmit	CANm transmit complete	TMTRF[1:0] in the RSCAN0TMSTSp register	TMIE in the RSCAN0TMIECy register
		CANm transmit abort	TMTRF[1:0] in the RSCAN0TMSTSp register	TAIE in the RSCAN0CmCTR register
		CANm transmit/receive FIFO transmit complete	CFTXIF in the RSCAN0CFSTSk register	CFTXIE in the RSCAN0CFCCk register
		CANm transmit queue	TXQIF in the RSCAN0TXQSTSm register	TXQIE in the RSCAN0TXQCCm register
		CANm transmit history	THLIF in the RSCAN0THLSTSm register	THLIE in the RSCAN0THLCCm register
	CANm transmit/receive FIFO receive complete		CFRXIF in the RSCAN0CFSTSk register	CFRXIE in the RSCAN0CFCCk register
	CANm error		• BEF in the RSCAN0CmERFL register • ALF in the RSCAN0CmERFL register • BLF in the RSCAN0CmERFL register • OVLF in the RSCAN0CmERFL register • BORF in the RSCAN0CmERFL register • BOEF in the RSCAN0CmERFL register • EPF in the RSCAN0CmERFL register • EWF in the RSCAN0CmERFL register	• BEIE in the RSCAN0CmCTR register • ALIE in the RSCAN0CmCTR register • BLIE in the RSCAN0CmCTR register • OLIE in the RSCAN0CmCTR register • BORIE in the RSCAN0CmCTR register • BOEIE in the RSCAN0CmCTR register • EPIE in the RSCAN0CmCTR register • EWIE in the RSCAN0CmCTR register

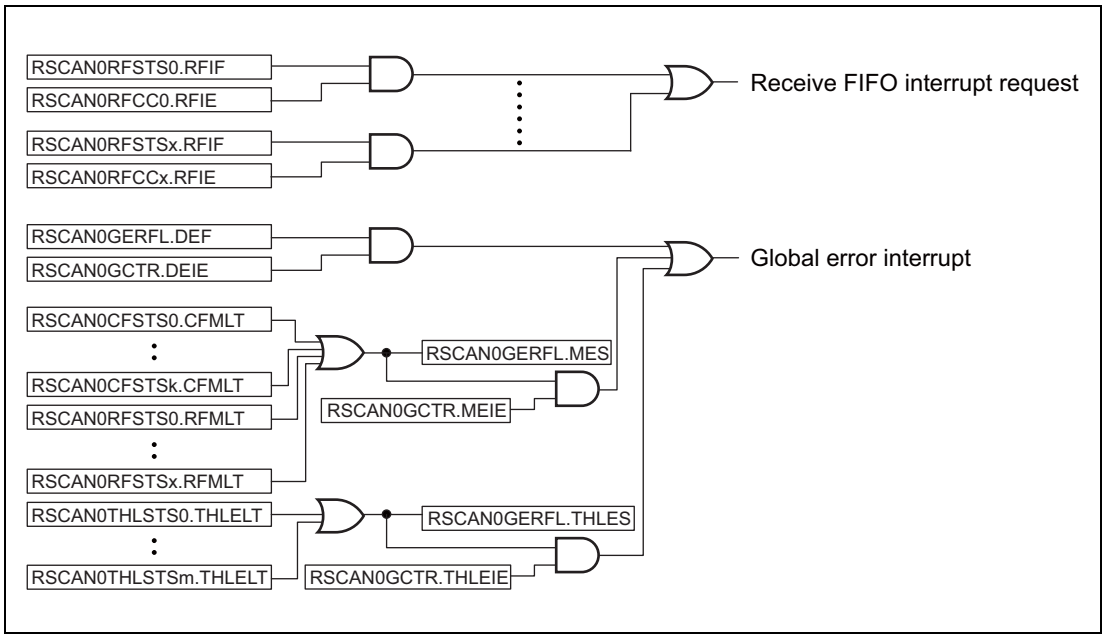


Figure 22.2 CAN Global Interrupt Block Diagram

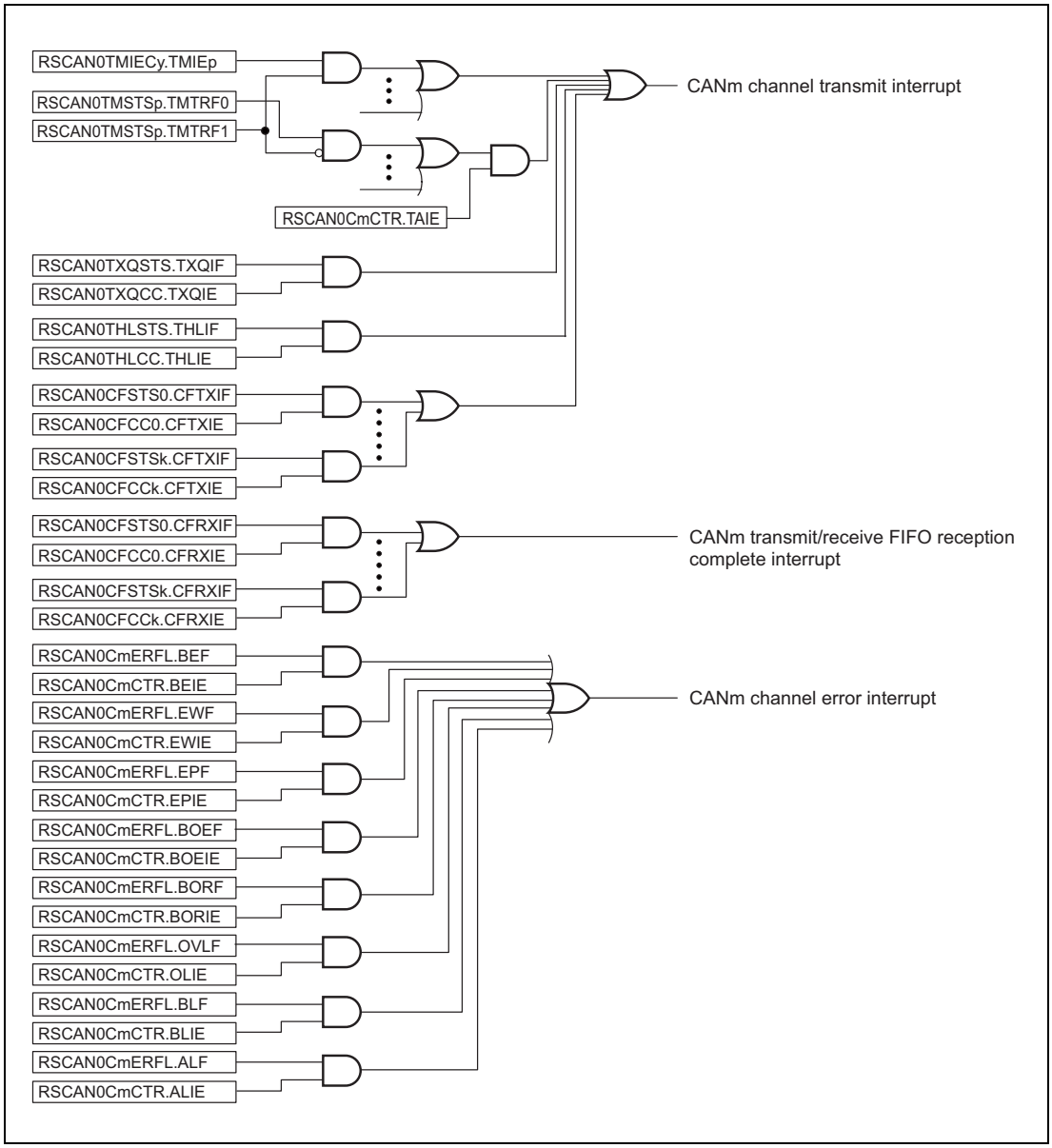


Figure 22.3 CAN Channel Interrupt Block Diagram

22.5 CAN Modes

The RSCAN module has four global modes to control the entire RSCAN module status and four channel modes to control individual channel status. Details of global modes are described in Section 22.5.1, Global Modes, and details of channel modes are described in Section 22.5.2, Channel Modes.

- Global stop mode: Stops the clocks of the entire module to achieve low power consumption.
- Global reset mode: Performs initial settings for the entire module.
- Global test mode: Performs test settings and performs the RAM test.
- Global operating mode: Makes the entire module operable.
- Channel stop mode: Stops the channel clock.
- Channel reset mode: Performs initial settings for the channels.
- Channel halt mode: Stops CAN communication and allows channel testing.
- Channel communication mode: Performs CAN communication.

22.5.1 Global Modes

Figure 22.4 shows the transitions of global modes.

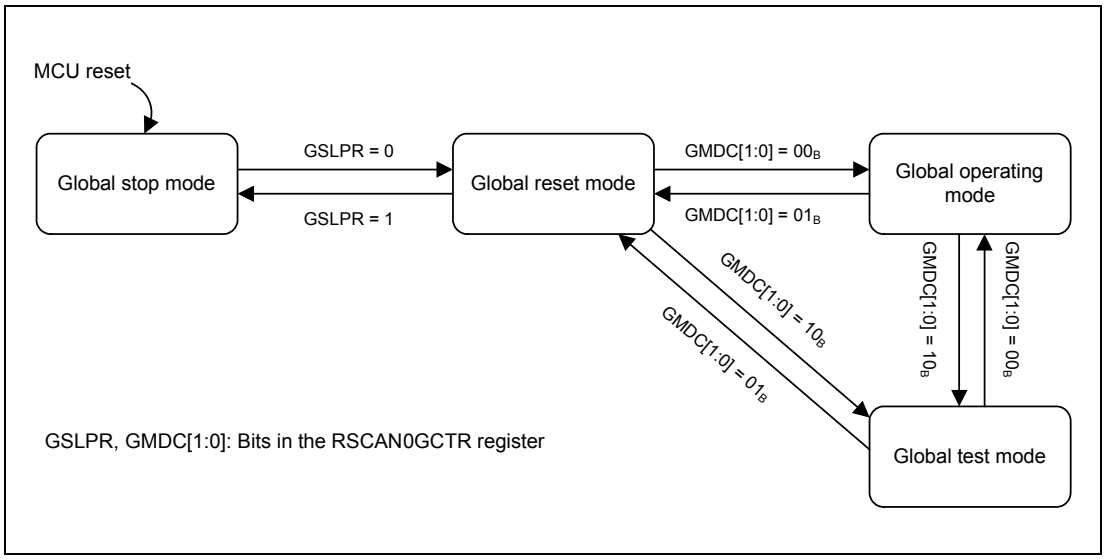


Figure 22.4 Transitions of Global Modes

In some cases, global mode transitions also force channel mode transitions. **Table 22.88** shows the channel mode transitions depending on the global mode setting dictated by the GMDC[1:0] bits and the GSLPR bit.

Table 22.88 Transitions of Channel Modes Depending on Global Mode Setting (GMDC[1:0] and GSLPR Bits)

Channel Mode before Setting	Channel Mode after Setting			
	GMDC[1:0] = 00_B GSLPR = 0 (Global Operation)	GMDC[1:0] = 10_B GSLPR = 0 (Global Test)	GMDC[1:0] = 01_B GSLPR = 0 (Global Reset)	GMDC[1:0] = 01_B GSLPR = 1 (Global Stop)
Channel communication	Channel communication	Channel halt	Channel reset	Transition prohibited
Channel halt	Channel halt	Channel halt	Channel reset	Transition prohibited
Channel reset	Channel reset	Channel reset	Channel reset	Channel stop
Channel stop	Channel stop	Channel stop	Channel stop	Channel stop

Note: GMDC[1:0], GSLPR: Bits in the RSCAN0GCTR register

Table 22.89 shows the global mode transition time.

Table 22.89 Global Mode Transition Time

Mode before Transition	Mode after Transition	Maximum Transition Time
Global stop	Global reset	Three pclk cycles
Global reset	Global stop	Three pclk cycles
Global reset	Global test	Ten pclk cycles
Global reset	Global operating	Ten pclk cycles
Global test	Global reset	Two CANm bit times*1
Global test	Global operating	Three pclk cycles
Global operating	Global reset	Two CANm bit times*1
Global operating	Global test	Two CAN frames*1

Note 1. CAN frame time of the lowest communication speed of the channels in use

22.5.1.1 Global Stop Mode

In global stop mode, clocks of the CAN do not run and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained. Only the clock used by the CPU for writing to the GSLPR bit runs in this mode.

After the MCU is reset, the CAN module transitions to global stop mode. Setting the GSLPR bit in the RSCAN0GCTR register to 1 (in global stop mode) in global reset mode sets the CSLPR bit in each of the RSCAN0CmCTR register to 1 (channel stop mode). If all channels are forced to transition to channel stop mode, the CAN module transitions to global stop mode. The GSLPR bit should not be modified in global operating mode or global test mode.

22.5.1.2 Global Reset Mode

In global reset mode, RSCAN module settings are performed. When the RSCAN module transitions to global reset mode, some registers are initialized. **Table 22.92** and **Table 22.93** list the registers to be initialized.

Setting the GMDC[1:0] bits in the RSCAN0GCTR register to 01_B sets the CHMDC[1:0] bits in each of the RSCAN0CmCTR registers (m = 0 to 2) to 01_B (channel reset mode). If all channels are forced to transition to channel reset mode, the CAN module transitions to global reset mode. Channels that are already in channel reset mode or channel stop mode do not transition (because the CHMDC[1:0] bits have already been set to 01_B).

22.5.1.3 Global Test Mode

In global test mode, settings for test-related registers are performed. When the CAN module transitions to global test mode, all CAN communications are disabled.

Setting the GMDC[1:0] bits in the RSCAN0GCTR register to 10_B sets the CHMDC[1:0] bits in each of the RSCAN0CmCTR register to 10_B (channel halt mode). If all channels are forced to transition to channel halt mode, the CAN module transitions to global test mode. Channels that are in channel stop mode, channel reset mode, or channel halt mode do not transition.

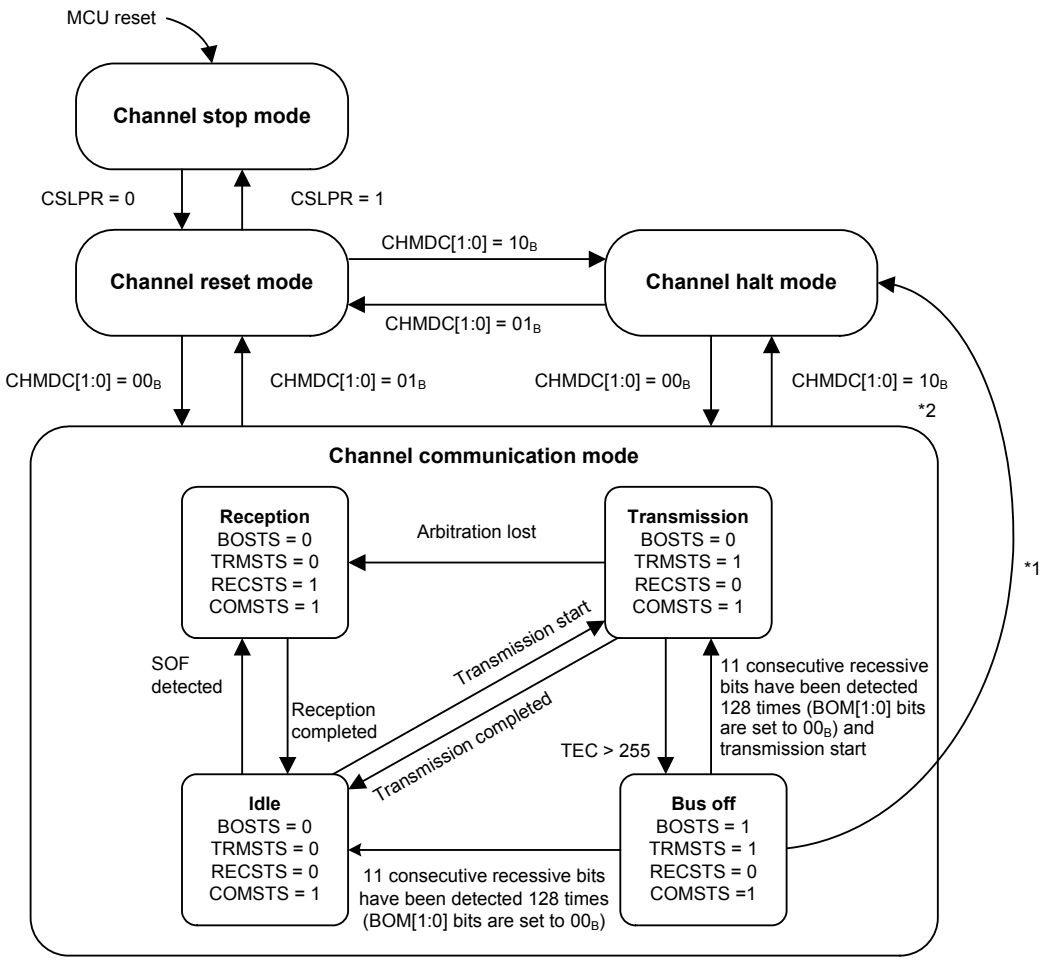
22.5.1.4 Global Operating Mode

The RSCAN module operates in global operating mode.

When the GMDC[1:0] bits in the RSCAN0GCTR register are set to 00_B, the RSCAN module transitions to global operating mode.

22.5.2 Channel Modes

Figure 22.5 shows a channel mode state transition chart. **Table 22.90** shows the channel mode transition time.



Note: CHMDC[1:0], CSLPR, BOM[1:0]: Bits in the RSCAN0CmCTR register (m = 0 to 2)
BOSTS, TRMSTS, RECSTS, COMSTS: Bits in the RSCAN0CmSTS register

Note 1. Timing of transition from bus off state to channel halt mode
- When BOM[1:0] = 01_B: Transition to channel halt mode when TEC exceeds 255
- When BOM[1:0] = 10_B: Transition to channel halt mode when 11 consecutive recessive bits have been detected 128 times
- When BOM[1:0] = 11_B: Transition to channel halt mode when the CHMDC[1:0] bits are set to 10_B

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode.

Figure 22.5 Channel Mode State Transition Chart

Table 22.90 Channel Mode Transition Time (1/2)

Mode before Transition	Mode after Transition	Maximum Transition Time
Channel stop	Channel reset	Three pclk cycles
Channel reset	Channel stop	Three pclk cycles
Channel reset	Channel halt	Three CANm bit times
Channel reset	Channel communication	Four CANm bit times
Channel halt	Channel reset	Two CANm bit times

Table 22.90 Channel Mode Transition Time (2/2)

Mode before Transition	Mode after Transition	Maximum Transition Time
Channel halt	Channel communication	Four CANm bit times
Channel communication	Channel reset	Two CANm bit times
Channel communication	Channel halt	Two CANm frames

22.5.2.1 Channel Stop Mode

In channel stop mode, clocks are not supplied to channels and therefore power consumption is reduced. Channel-related registers can be read, but writing data to them is prohibited. Register values are retained.

Each channel enters channel stop mode after the MCU is reset. Channels also transition to channel stop mode when the CSLPR bit in the RSCAN0CmCTR register (m = 0 to 2) is set to 1 (channel stop mode) in channel reset mode. The CSLPR bit should not be modified in channel communication mode and channel halt mode.

22.5.2.2 Channel Reset Mode

In channel reset mode, channel settings are performed. When a channel transitions to channel reset mode, some channel-related registers are initialized. **Table 22.92** lists the registers to be initialized.

When the CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 01_B (channel reset mode) during CAN communication, communication is terminated before it is completed and the channel transitions to channel reset mode. **Table 22.91** shows the operation when the CHMDC[1:0] bits are set to 01_B (channel reset mode) during CAN communication.

22.5.2.3 Channel Halt Mode

In channel halt mode, settings for test-related registers of channels are performed. When a channel transitions to channel halt mode, CAN communication of the channel stops.

Table 22.91 shows operation when the CHMDC[1:0] bits are set to 10_B (channel halt mode) during CAN communication.

Table 22.91 Operation a Channel Transitions to Channel Reset Mode/Channel Halt Mode

Mode	During Reception	During Transmission	Bus Off State
Channel reset (CHMDC[1:0] = 01_B)	Transitions to channel reset mode before reception is completed.*1	Transitions to channel reset mode before transmission is completed.*1	Transitions to channel reset mode before bus off recovery.
Channel halt*3 (CHMDC[1:0] = 10_B)	Transitions to channel halt mode after reception is completed.*2	Transitions to channel halt mode after transmission is completed.	[When BOM[1:0] = 00_B] Transitions to channel halt mode (CHMDC[1:0] = 10_B) only after bus off recovery. [When BOM[1:0] = 01_B] Transitions to channel halt mode automatically when the condition for transition to bus off state is met. [When BOM[1:0] = 10_B] Transitions to channel halt mode automatically after bus off recovery. [When BOM[1:0] = 11_B] Transitions to channel halt mode immediately after the CHMDC[1:0] bits are set to 10_B before bus off recovery.

Note 1. To allow transition to channel reset mode after communication is completed, set the CHMDC[1:0] bits to 10_B and confirm that communication has been completed and transition to channel halt mode has been made, and then set the CHMDC[1:0] bits to 01_B.

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode. The CAN bus status can be confirmed with the BLF flag of the RSCAN0CmERFL register that becomes 1 when dominant lock is detected.

Note 3. When the transition from channel reset mode to channel halt mode is to be made, set the RSCAN0CmCFG register in channel reset mode and then shift to channel halt mode.

22.5.2.4 Channel Communication Mode

In channel communication mode, CAN communication is performed. Each channel has the following communication states during CAN communication.

- Idle: Neither reception nor transmission is in progress.
- Reception:Receiving a message sent from another node.
- Transmission:Transmitting a message.
- Bus off:Isolated from CAN communication.

When the CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 00_B, the channel transitions to channel communication mode. After that, once 11 consecutive recessive bits have been detected, the COMSTS flag in the RSCAN0CmSTS register (m = 0 to 2) is set to 1 (communication is ready) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

22.5.2.5 Bus Off State

A channel transitions to the bus off state according to the transmit/receive error counter increment/decrement rules of the CAN specifications.

The conditions for returning from the bus off state are determined by the BOM[1:0] bits in the RSCAN0CmCTR register.

- When BOM[1:0] = 00_B:
  Bus off recovery is compliant with the CAN specifications. After 11 consecutive recessive bits have been detected 128 times, a channel returns from the bus off state to the CAN communication ready state (error active state). At that time, the TEC[7:0] and REC[7:0] bits in the RSCAN0CmSTS register are initialized to 00_H, the BORF flag in the RSCAN0CmERFL register is set to 1 (bus off recovery is detected), and a bus off recovery interrupt request is generated. When the CHMDC[1:0] bits in the RSCAN0CmCTR register are set to 10_B (channel halt mode) in the bus off state, the channel transitions to channel halt mode after bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times).
- When BOM[1:0] = 01_B:
  When a channel transitions to the bus off state, the CHMDC[1:0] bits are set to 10_B and the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H, but the BORF flag is not set to 1. Also, a bus off recovery interrupt request is not generated.
- When BOM[1:0] = 10_B:
  When a channel has transitioned to the bus off state, the CHMDC[1:0] bits are set to 10_B. After bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times), the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H, the BORF flag is set to 1, and a bus off recovery interrupt request is generated.
- When BOM[1:0] = 11_B:
  When the CHMDC[1:0] bits are set to 10_B in the bus off state, the channel transitions to channel

halt mode before bus off recovery is completed. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H, but the BORF flag is not set to 1. Also, a bus off recovery interrupt is not generated.
However, the BORF flag becomes 1 and a bus off recovery interrupt request is generated if a CAN module transitions to error active state (by detecting 128 times of 11 consecutive recessive bits) before CHMDC[1:0] bits are set to 10_B.

If the RSCAN module causes the channel to transition to channel halt mode simultaneously with a program write to the CHMDC[1:0] bits, the program write takes precedence. An automatic transition to channel halt mode when the BOM[1:0] bits are set to 01_B or 10_B is made only when the CHMDC[1:0] bits are 00_B (channel communication mode).

Furthermore, setting the RTBO bit in the RSCAN0CmCTR register to 1 allows a forced return from the bus off state. As soon as the RTBO bit is set to 1, the state changes to the error active state. After 11 consecutive recessive bits have been detected, the CAN module becomes ready for communication. In this case, the BORF flag is not set to 1 and the TEC[7:0] and REC[7:0] bits are initialized to 00_H. Write 1 to the RTBO bit only when the BOM[1:0] value is 00_B. Writing the RTBO bit to 1 in a state other than the bus off state is ignored, and the RTBO bit is immediately set to 0.

Table 22.92 Registers Initialized in Global Reset Mode or Channel Reset Mode

Register	Bit / Flag
RSCAN0CmCTR register	CTMS[1:0], CTME, CHMDC[1:0]
RSCAN0CmSTS register	CHLTSTS, EPSTS, BOSTS, TRMSTS, RECSTS, COMSTS, REC[7:0], TEC[7:0]
RSCAN0CmERFL register	CRCREG[14:0], ADERR, B0ERR, B1ERR, CERR, AERR, FERR, SERR, ALF, BLF, OVLF, BORF, BOEF, EPF, EWF, BEF
RSCAN0CFCCk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFE
RSCAN0CFSTSk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFMC[7:0], CFFLL, CFEMP, CFMLT, CFRXIF, CFTXIF
RSCAN0CFTISTS register	CFkTXIF
RSCAN0TMCp register	TMOM, TMTAR, TMTR
RSCAN0TMSTSp register	TMTARM, TMTRM, TMTRF[1:0], TMTSTS
RSCAN0TMTRSTSy register	TMTRSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TMTARSTSy register	TMTARSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TMTCSTSy register	TMTCSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TMTASTSy register	TMTASTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCAN0TXQCCm register	TXQE
RSCAN0TXQSTSm register	TXQIF, TXQFLL, TXQEMP
RSCAN0THLCCm register	THLE
RSCAN0THLSTSm register	THLMC[4:0], THLIF, THLELT, THLFLL, THLEMP
RSCAN0GTINTSTS0 register	TSIFm, TAIFm, TQIFm, CFTIFm, THIFm (m = 0 to 2)

Table 22.93 Registers Initialized Only in Global Reset Mode (1/2)

Register	Bit / Flag
RSCAN0GSTS register	GHLTSTS
RSCAN0GERFL register	THLES, MES, DEF
RSCAN0GTSC register	TS[15:0]
RSCAN0RMNDy register	RMNSq

Table 22.93 Registers Initialized Only in Global Reset Mode (2/2)

Register	Bit / Flag
RSCAN0RFCCx register	RFE
RSCAN0RFSTSx register	RFMC[7:0], RFIF, RFMLT, RFFLL, RFEMP
RSCAN0CFCCk register	When transmit/receive FIFO buffer is in receive mode: CFE
RSCAN0CFSTSk register	When transmit/receive FIFO buffer is in receive mode: CFMC[7:0], CFFLL, CFEMP, CFTXIF, CFRXIF, CFMLT
RSCAN0FESTS register	CFkEMP, RFxEMP
RSCAN0FFSTS register	CFkFLL, RFxFLL
RSCAN0FMSTS register	CFkMLT, RFxMLT
RSCAN0RFISTS register	RFxIF
RSCAN0CFRISTS register	CFkRXIF
RSCAN0GTSTCFG register	RTMPS[6:0], C0ICBCE, C1ICBCE, C2ICBCE
RSCAN0GTSTCTR register	RTME, ICBCTME

22.6 Reception Function

There are two reception types.

- Reception by receive buffers:
  Zero to 48 receive buffers can be shared by all channels. Since messages stored in receive buffers are overwritten at each reception, the latest receive data can always be read.
- Reception by receive FIFO buffers and transmit/receive FIFO buffers (receive mode):
  Eight receive FIFO buffers can be shared by all channels and three dedicated transmit/receive FIFO buffers are provided for each channel. Messages of up to the number of buffer stages specified with the RFDC[2:0] and CFDC[2:0] bits can be stored in FIFO buffers and can be read sequentially from the oldest.

22.6.1 Data Processing Using the Receive Rule Table

Data processing using the receive rule table allows dispatching of selected messages to the specified buffer. Data processing includes acceptance filter processing, DLC filter processing, routing processing, label addition processing, and mirror function processing.

Up to 128 receive rules can be registered per channel and up to (64 × number of channels) total receive rules can be registered in the entire module. (Up to 192 receive rules can be registered in this module). Set receive rules for each channel. Receive rules cannot be shared with other channels. If receive rules are not set, no messages can be received. **Figure 22.6** illustrates how receive rules are registered.

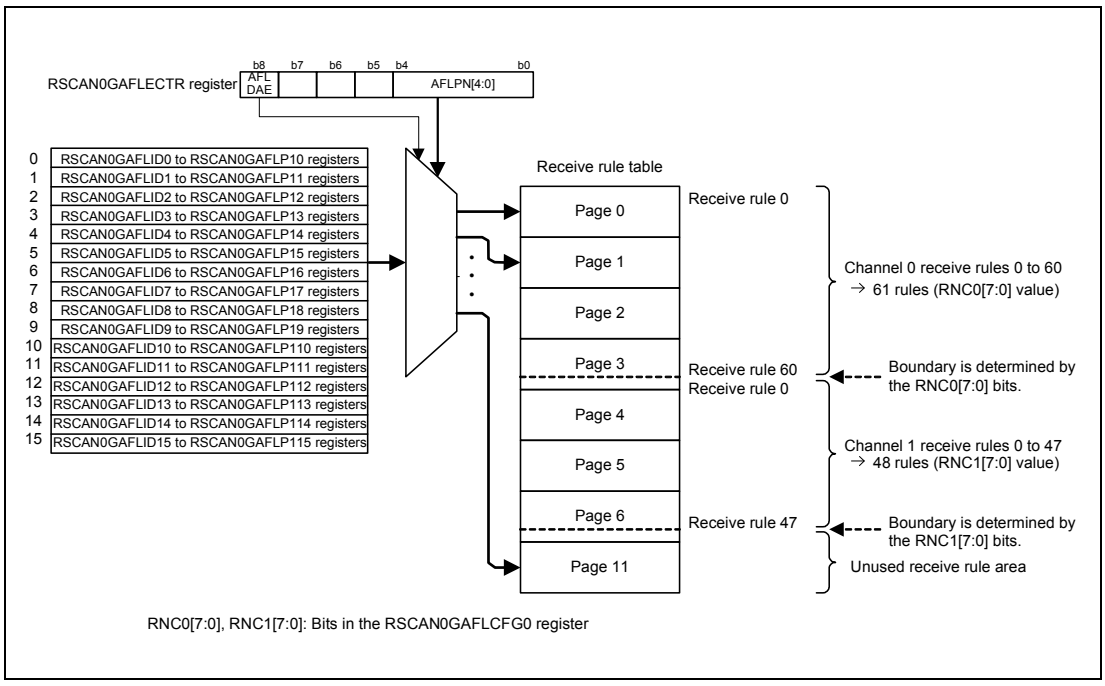


Figure 22.6 Entry of Receive Rules (for Setting Channel 0 and 1)

CAUTION

Receive rules for each channel must be set in contiguous blocks.
Channel 1 rules and channel 0 rules must be set separately.

Each receive rule consists of 16 bytes in the RSCAN0GAFLIDj, RSCAN0GAFLMj, RSCAN0GAFLP0j, and RSCAN0GAFLP1j registers (j = 0 to 15). The RSCAN0GAFLIDj register is

used to set GAFLID[28:0], GAFLIDE bit, GAFLRTR bit, and the mirror function, the RSCAN0GAFLMj register is used to set mask, the RSCAN0GAFLP0j register is used to set label information to be added, DLC value, and storage receive buffer, and the RSCAN0GAFLP1j register is used to set storage FIFO buffer. Up to 16 receive rules can be set per page.

22.6.1.1 Acceptance Filter Processing

In the acceptance filter processing, the ID data, IDE bit, and RTR bit in a received message are compared with the ID data, IDE bit, and RTR bit set in the receive rule of the corresponding channel. When all these bits match, the message passes through the acceptance filter processing. The ID data, IDE bit, and RTR bit in the received message which correspond to the bits set to 0 (bits are not compared) in the RSCAN0GAFLMj register are not compared and are regarded as matched.

Check begins with the receive rule of the minimum number for the corresponding channel. When all the bits to be compared in a received message match the bits set in the receive rule or when all the receive rules are compared without any match, filter processing stops. If there is no matching receive rule, the received message is not stored in the receive buffer or FIFO buffer.

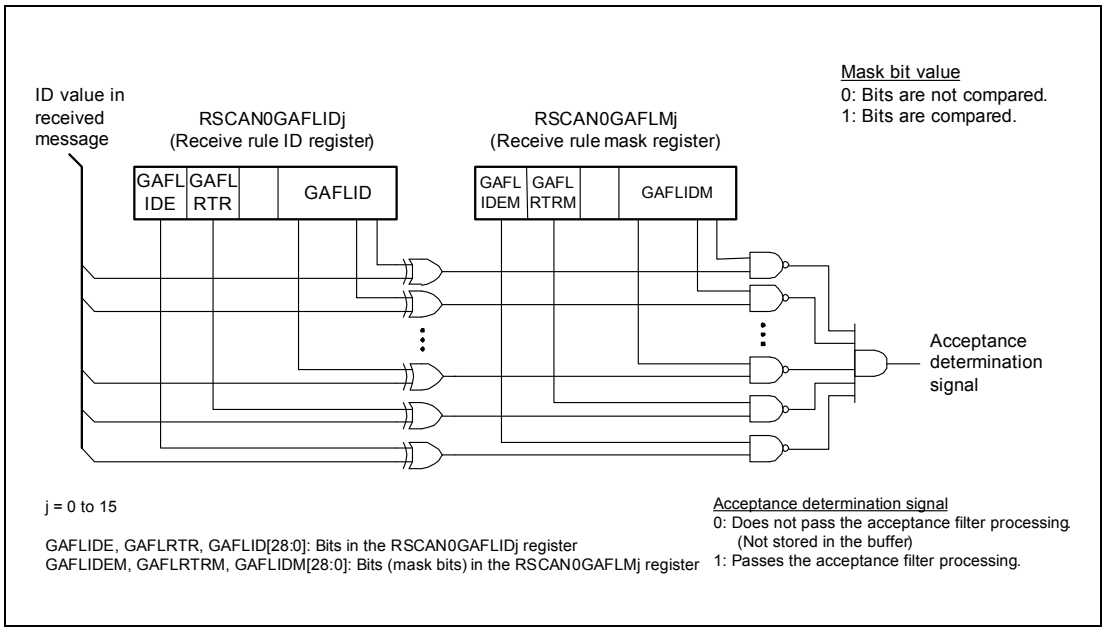


Figure 22.7 Acceptance Filter Function

22.6.1.2 DLC Filter Processing

When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), DLC filter processing is added to messages that passed through the acceptance filter processing. When the DLC value in a message is equal to or larger than the DLC value set in the receive rule, the message passes through the DLC filter processing.

When a message has passed through the DLC filter processing with the DRE bit in the RSCAN0GCFG register set to 0 (DLC replacement is disabled), the DLC value in the received message is stored in the buffer. In this case, all the data bytes in the received message are stored in the buffer.

When a message has passed through the DLC filter processing with the DRE bit in the RSCAN0GCFG register set to 1 (DLC replacement is enabled), the DLC value in the receive rule is stored in the buffer instead of the DLC value in the received message. In this case, a value of 00_H is stored in each data byte beyond the number of bytes which is indicated by the DLC value in the receive rule.

When the DLC value in the received message is smaller than that in the receive rule, the message does not pass through the DLC filter processing. In this case, the message is not stored in the receive buffer or the FIFO buffer and the DEF flag in the RSCAN0GERFL register is set to 1 (a DLC error is present).

22.6.1.3 Routing Processing

Messages that passed through the acceptance filter processing and the DLC filter processing are stored in receive buffers, receive FIFO buffers, or transmit/receive FIFO buffers (set to receive mode or gateway mode). Message storage destination is set by the GAFLRMV and GAFLRMDP[6:0] bits in the RSCAN0GAFLP0j register (j = 0 to 15) and by the RSCAN0GAFLP1j register. Messages that passed through the acceptance filter processing and the DLC filter processing can be stored in up to eight buffers.

22.6.1.4 Label Addition Processing

It is possible to add 12-bit label information to messages that passed through the filter processing and store them in buffers. This label information is set in the GAFLPTR[11:0] bits in the RSCAN0GAFLP0j register.

22.6.1.5 Mirror Function Processing

The mirror function allows the CAN node to receive its own transmitted messages. The mirror function is made available by setting the MME bit in the RSCAN0GCFG register to 1 (mirror function is enabled).

When the mirror function is in use, receive rules for which the GAFLLB bit in the RSCAN0GAFLIDj register is set to 0 are used for data processing when receiving messages transmitted from other CAN nodes. When the CAN node is receiving its own transmitted messages, receive rules for which the GAFLLB bit is set to 1 are used for data processing.

22.6.1.6 Timestamp

The timestamp counter is a 16-bit free-running counter used for recording message receive time. The timestamp counter value is fetched at the start-of-frame (SOF) timing of a message and is then stored in a receive buffer or a FIFO buffer together with the message ID and data. Either pclk/2 or the CANm bit time clock (m = 0 to 2) may be selected as a timestamp counter clock source using the TSBTCS[2:0] and TSSS bits in the RSCAN0GCFG register. The timestamp counter count source is obtained by dividing the selected clock source by the TSP[3:0] value in the RSCAN0GCFG register.

When the CANm bit time clock is used as a clock source, the timestamp counter stops when the corresponding channel transitions to channel reset mode or channel halt mode. When the pclk/2 is used as a clock source, the timestamp function is not affected by channel mode.

The timestamp counter value is reset to 0000_H by setting the TSRST bit in the RSCAN0GCTR register to 1.

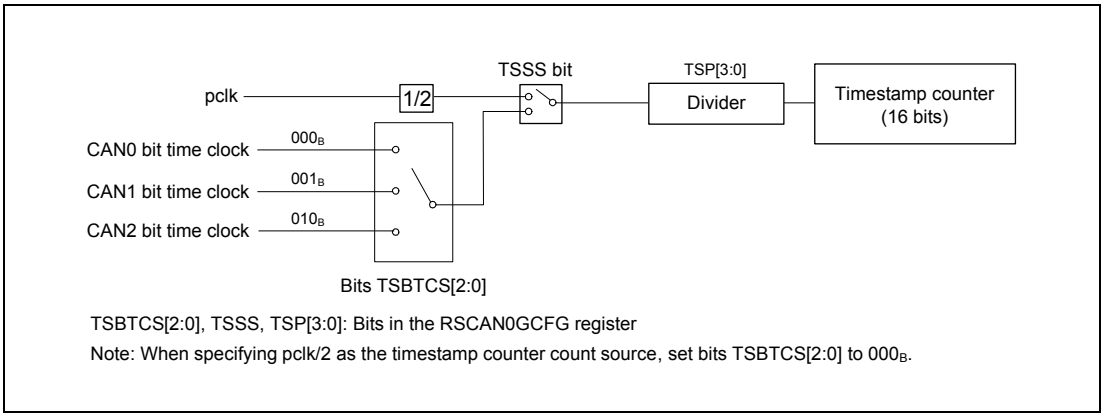


Figure 22.8 Timestamp Function Block Diagram

22.7 Transmission Functions

There are three types of transmission.

- Transmission using transmit buffers:
  Each channel has 16 buffers.
- Transmission using transmit/receive FIFO buffers (transmit mode):
  Each channel has three FIFO buffers. Up to 128 messages can be contained in a single FIFO buffer. Each FIFO buffer is used with a link to a transmit buffer. Only the message to be transmitted next in a FIFO buffer becomes the target of transmit priority determination. Messages are transmitted sequentially on a first-in, first-out basis.
- Transmission using transmit queues:
  Up to 16 transmit buffers per channel can be allocated to the transmit queues. Transmit buffer ((16 × m) + 15) is used as an access window of a corresponding channel. Transmit buffers are allocated to transmit queues in descending order of buffer number. All messages in transmit queues, which are targets of priority determination, are transmitted in the order of ID number.

Figure 22.9 shows the allocation of transmit queues and transmit/receive FIFO buffer link.

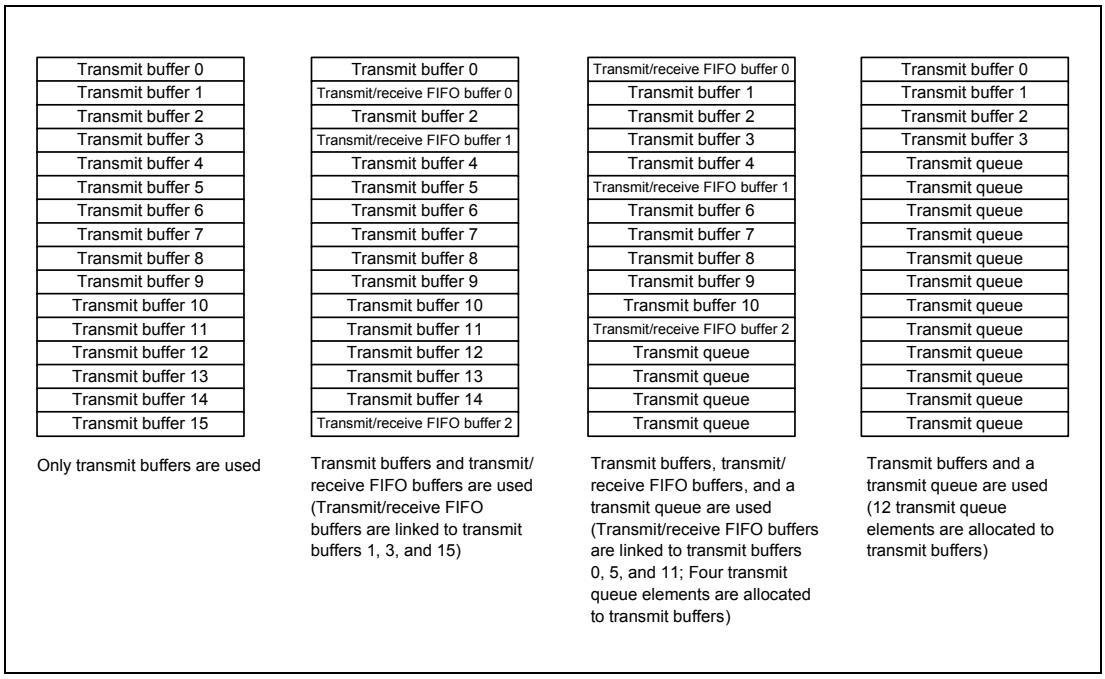


Figure 22.9 Allocation of Transmit Queues and Transmit/Receive FIFO Buffer Links

22.7.1 Transmit Priority Determination

If transmit requests are issued from multiple buffers or from the queue on the same channel, transmit priority is determined using one of the following methods.

- ID priority (TPRI bit = 0)
- Transmit buffer number priority (TPRI bit = 1)

All CAN channels use the setting of the TPRI bit in the RSCAN0GCFG register.

When the TPRI bit is set to 0, messages are transmitted according to the priority of stored message IDs. ID priority conforms to the CAN bus arbitration specification defined in the CAN specifications. All IDs of pending transmit messages are targets of priority determination, regardless of whether they are stored in transmit buffers, transmit/receive FIFO buffers (set to transmit mode or gateway mode), or the transmit queue. If even a single transmit queue is used, select ID priority. When transmit/receive FIFO buffers are used, the oldest message in a FIFO buffer becomes the target of priority determination. When a message is being transmitted from a transmit/receive FIFO buffer, the next message in the FIFO buffer becomes the target of priority determination. When a transmit queue is used, all messages in the transmit queue are targets of priority determination. If the same ID is set for two or more buffers, the buffer with the smaller buffer number takes precedence.

When the TPRI bit is set to 1, the message in the transmit buffer with the minimum buffer number among all buffers with a transmit request is transmitted first. When transmit/receive FIFO buffers are linked to transmit buffers, transmit priority is determined according to linked transmit buffer numbers.

When messages are retransmitted due to an arbitration-lost or an error, transmit priority determination is made again according to the TPRI bit.

22.7.2 Transmission Using Transmit Buffers

Setting the transmit request bit (TMTR bit in the RSCAN0TMCp register) in a transmit buffer to 1 (transmission is requested) allows transmission of data frames or remote frames.

The transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCAN0TMSTSp register (p = 0 to 47). When transmit completes successfully, the TMTRF[1:0] flag is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)).

22.7.2.1 Transmit Abort Function

With respect to transmit buffers for which the TMTRM bit in the RSCAN0TMSTSp register is set to 1 (a transmit request is present), when the TMTAR bit in the RSCAN0TMCp register is set to 1 (transmit abort is requested), the transmit request is canceled. When transmit abort is completed, the TMTRF[1:0] flag in the RSCAN0TMSTSp register is set to 01_B (transmit abort has been completed) and the transmit request is canceled (clearing the TMTRM bit to 0).

A message that is being transmitted or a message to be transmitted next according to the transmit priority determination cannot be aborted. However, when an arbitration-lost or an error occurs during transmission of a message for which the TMTAR bit is set to 1, retransmission is not performed.

22.7.2.2 One-Shot Transmission Function (Retransmission Disabling Function)

When the TMOM bit in the RSCAN0TMCp register is set to 1 (one-shot transmission is enabled), transmission is performed only once. Even if an arbitration-lost or an error occurs, retransmission is not performed.

The one-shot transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCAN0TMSTSp register. When one-shot transmission completes successfully, the TMTRF[1:0] flag is set to 10_B or 11_B. When an arbitration-lost or an error occurs, the TMTRF[1:0] flag is set to 01_B (transmit abort has been completed).

22.7.3 Transmission Using FIFO Buffers

Multiple messages can be stored in a single transmit/receive FIFO buffers, up to the number specified by the FIFO buffer depth, which is set by the CFDC[2:0] bits in the RSCAN0CFCCk register (k = 0 to 8). Messages are transmitted sequentially on a first-in, first-out basis.

Each transmit/receive FIFO buffer is linked to a transmit buffer selected by the CFTML[3:0] bits in the RSCAN0CFCCk register. When the CFE bit in the RSCAN0CFCCk register is set to 1 (transmit/receive FIFO buffers are used), transmit/receive FIFO buffers become targets of transmit priority determination. Priority of only the next transmit message is determined in the FIFO buffer.

When the CFE bit is set to 0 (no transmit/receive FIFO buffer is used), the CFEMP flag is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) at the timing below.

- The transmit/receive FIFO buffer becomes empty immediately if the message in it is not being transmitted or is not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, arbitration-lost or the transition to channel halt mode in the case that a message in it is being transmitted or to be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages cannot be stored in FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

22.7.3.1 Interval Transmission Function

A message transmission interval time can be set to space the transmission of messages from the same FIFO buffer when using a transmit/receive FIFO buffer set to transmit mode or gateway mode.

Immediately after the first message has been transmitted successfully from the FIFO buffer with the CFE bit in the RSCAN0CFCCk register set to 1, the interval timer starts counting (after EOF7 of the CAN protocol). After that, when the interval time has passed, the next message is transmitted. The interval timer stops in channel reset mode or by clearing the CFE bit to 0.

The interval time is set by the CFITT[7:0] bits in the RSCAN0CFCCk register. When the interval timer is not used, set the CFITT[7:0] bits to 00_H.

Select an interval timer count source using the CFITR and CFITSS bits in the RSCAN0CFCCk register. When the CFITR and CFITSS bits are set to 00_B, the count source is obtained by dividing pclk/2 by the value of the ITRCP[15:0] bits. When the CFITR and CFITSS bits are set to 10_B, the count source is obtained by dividing pclk/2 by (the value of the ITRCP[15:0] bits in the RSCAN0GCFG register × 10). When the CFITR and CFITSS bits are set to $x1_B$, the CANm bit time clock is used as a count source.

The interval time is calculated by the following equations where M is the value of ITRCP[15:0] and N is the value of CFITT[7:0].

- When CFITR and CFITSS = 00_B (fPBA is the frequency of pclk):

$$\frac{1}{fPBA} \times 2 \times M \times N$$

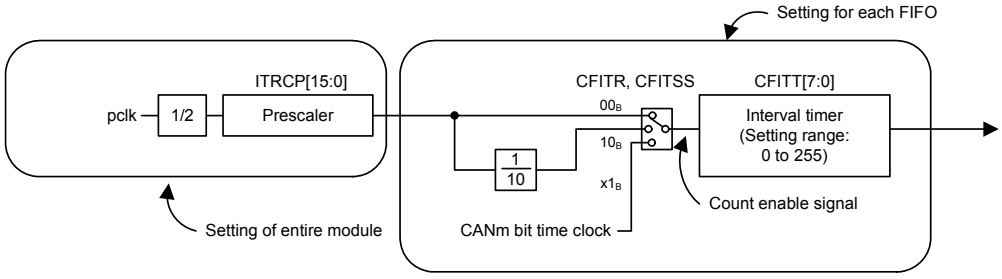
- When CFITR and CFITSS = 10_B:

$$\frac{1}{fPBA} \times 2 \times M \times 10 \times N$$

- When CFITR and CFITSS = $x1_B$ (fCANBIT is the frequency of CANm bit time clock):

$$\frac{1}{fCANBIT} \times N$$

Figure 22.10 shows the interval timer block diagram.

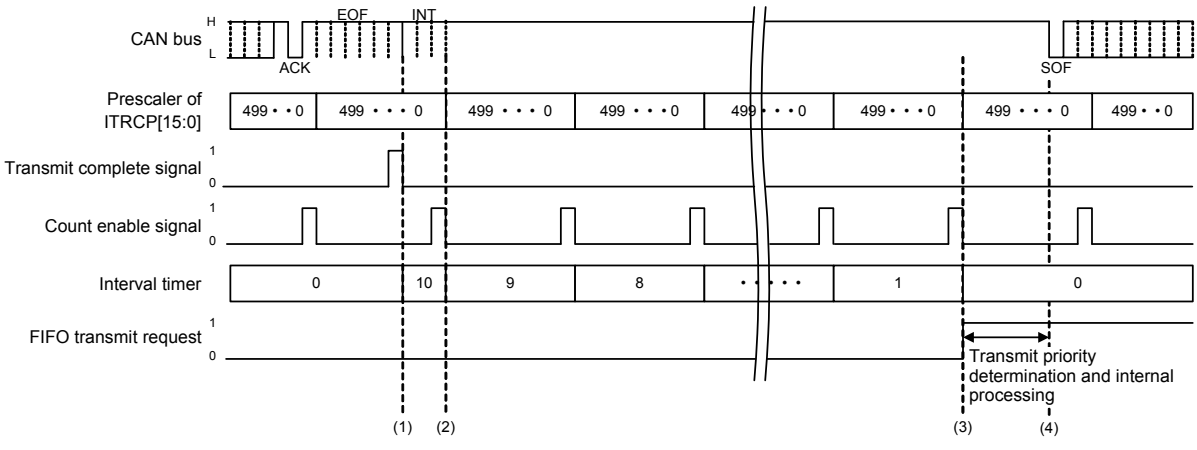


ITRCP[15:0]: Bits in the RSCAN0GCFG register
CFITR, CFITSS, CFITT[7:0] : Bits in the RSCAN0CFCCk register

Note: m = 0 to 2, k = 0 to 8

Figure 22.10 Interval Timer Block Diagram

Figure 22.11 shows the interval timer timing diagram.



Interval time (logical value) = 2/fPBA × (set ITRCP[15:0] value) × set CFITT[7:0] value

fPBA: Frequency of pclk
ITRCP[15:0]: Bits in the RSCAN0GCFG register (The set value is 500 in this figure.)
CFITT[7:0]: Bits in the RSCAN0CFCCk register (The set value is 10 in this figure.)

Figure 22.11 Interval Timer Timing Chart

(1) The interval timer starts counting upon completion of transmission. Since the prescaler is not initialized at the time of transmission completion, the first interval time contains an error of up to one count of the interval timer.

(2) The interval timer is decremented by 1 upon the next count enable signal.

(3) When the interval timer has decreased to 0, the transmit/receive FIFO buffer issues a transmit request.

(4) The transmit/receive FIFO buffer is determined for the next transmission by the priority determination, it starts transmitting data. Transmission starts usually with a delay of three CANm bit time clock cycles or less from the issue of transmit request. If multiple internal processes (such as receive filter processing, message routing, and transmit priority determination) take place in all channels, a delay of up to 582 cycles of the pclk may be generated.

22.7.4 Transmission Using Transmit Queues

Three to sixteen buffers are allocated to a transmit queue for each channel, and transmit buffer ((16 × m) + 15) is used as an access window of a corresponding channel.

All messages in a transmit queue are targets of transmit priority determination and are transmitted in the ID priority order regardless of storage sequence. If two messages having the same ID are stored in a transmit queue, these messages are not always transmitted in the order of their storage in the transmit queue.

Setting the TXQE bit in the RSCAN0TXQCCm register to 0 disables transmit queues. When the TXQE bit is set to 0, the TXQEMP flag in the RSCAN0TXQSTSm register is set to 1 (the transmit queue contains no messages (transmit queue empty) at the timing below.

- The transmit queue becomes empty immediately when no message in it is being transmitted or will be transmitted next.
- The transmit queue becomes empty after transmission completion, CAN bus error detection, arbitration-lost, or the transition to channel halt mode when a message in it is being transmitted or will be transmitted next.

When the TXQE bit is cleared to 0, all messages in transmit queues are lost and messages cannot be stored in transmit queues. Confirm that the TXQEMP flag is set to 1 before setting the TXQE bit to 1 again.

22.7.5 Transmit History Function

Information about transmission-completed messages can be stored in the transmit history buffer. Each channel has a single transmit history buffer that can contain 16 sets of transmit history data.

A message transmit source buffer type can be selected by the THLDTE bit in the RSCAN0THLCCm register. The THLEN bit in the RSCAN0CFIDk register (k = 0 to 8) determines whether transmit history data is stored for each message.

The following information on a transmitted message will be stored in the transmission history buffer after the successful completion of transmission.

Storage of the transmission history data after the successful completion of transmission may take up to 150 cycles of pclk.

- Buffer type
  001_B: Transmit buffer
  010_B: Transmit/receive FIFO buffer
  100_B: Transmit queue
- Buffer number
  Number of source transmit buffer, transmit queue, or transmit/receive FIFO buffer. This number depends on buffer types. See **Table 22.94**.

- Label data Label information of the transmit message

Table 22.94 Transmit History Data Buffer Numbers

Buffer type / Buffer No.	001_B	010_B	100_B
0000_B	Transmit buffer 16 × m + 0	Buffer numbers of the transmit buffer linked to the transmit/receive FIFO buffer by the CFTML[3:0] bits in the RSCAN0CFCCk register (k = 0 to 8)	Buffer numbers of the transmit buffer allocated to the transmit queue that performed transmission
0001_B	Transmit buffer 16 × m + 1		
0010_B	Transmit buffer 16 × m + 2		
0011_B	Transmit buffer 16 × m + 3		
0100_B	Transmit buffer 16 × m + 4		
0101_B	Transmit buffer 16 × m + 5		
0110_B	Transmit buffer 16 × m + 6		
0111_B	Transmit buffer 16 × m + 7		
1000_B	Transmit buffer 16 × m + 8		
1001_B	Transmit buffer 16 × m + 9		
1010_B	Transmit buffer 16 × m + 10		
1011_B	Transmit buffer 16 × m + 11		
1100_B	Transmit buffer 16 × m + 12		
1101_B	Transmit buffer 16 × m + 13		
1110_B	Transmit buffer 16 × m + 14		
1111_B	Transmit buffer 16 × m + 15		

Label data is used to identify each message. Unique label data can be added to each message transmitted from a transmit buffer, transmit queue, or transmit/receive FIFO buffer.

Transmit history data can be read from the RSCAN0THLACCm register. If an attempt is made to store new transmit history data while the buffer is full, the buffer overflows and the new data is discarded.

22.8 Gateway Function

When a transmit/receive FIFO buffer is set to gateway mode, receive massages can be transmitted from an arbitrary channel without CPU intervention.

When the CFM[1:0] bits in the RSCAN0CFCCk register are set to 10_B (gateway mode) and the transmit/receive FIFO buffer of a channel transmitting a message is selected in RSCAN0GAFLP1j register, messages that passed through the filter processing of the receive rule are stored in the specified transmit/receive FIFO buffer and are automatically transmitted from the buffer.

Messages stored in a transmit/receive FIFO buffer are transmitted sequentially on a first-in, first-out basis. Only the message to be transmitted next becomes the target of transmit priority determination.

Transmit/receive FIFO buffers in the gateway mode are disabled by setting the CFE bit in the RSCAN0CFCCk register to 0 and the CFEMP flag becomes 1 according to the timing below.

- The transmit/receive FIFO buffer becomes empty immediately when the message in it is not being transmitted and will not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration-lost when the message in it is being transmitted or will be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages can no longer be stored in transmit/receive FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

22.9 Test Function

The test function is classified into communication tests and global tests.

- Communication tests: Performed for each channel.
  - Standard test mode
  - Listen-only mode
  - Self-test mode 0 (external loopback mode)
  - Self-test mode 1 (internal loopback mode)
- Global tests: Performed for the entire module
  - RAM test (read/write test)
  - Inter-channel communication test

22.9.1 Standard Test Mode

Standard test mode allows CRC test.

22.9.2 Listen-Only Mode

Listen-only mode allows reception of data frames and remote frames. Only recessive bits are transmitted on the CAN bus, and the ACK bit, overload flag, and active error flag are not transmitted.

Listen-only mode is available for detecting the communication speed.

Do not make a transmit request from any buffer or queue in listen-only mode.

Figure 22.12 shows the connection when listen-only mode is selected.

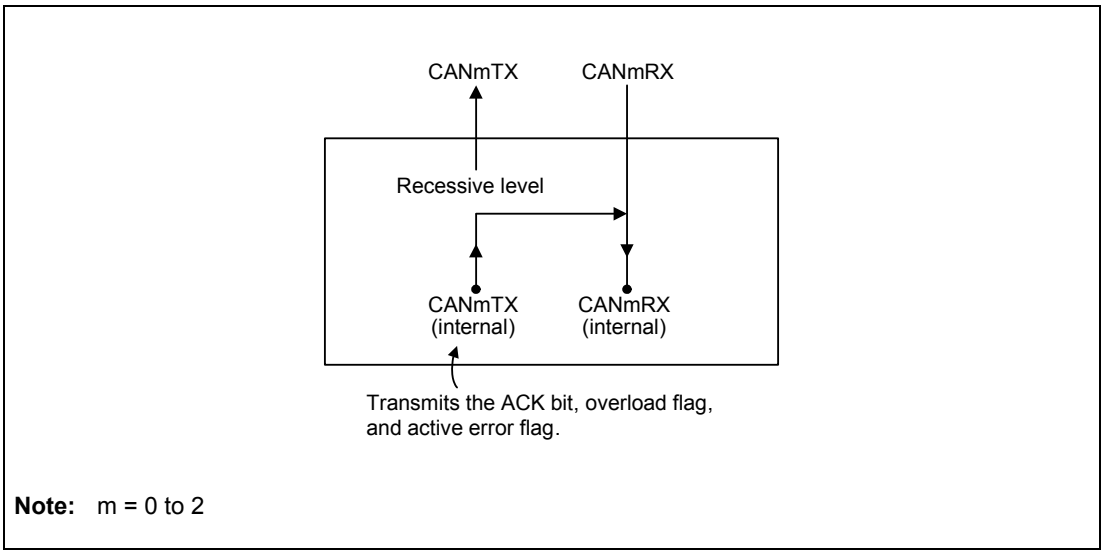


Figure 22.12 Connection when Listen-Only Mode is Selected

22.9.3 Self-Test Mode (Loopback Mode)

In self-test mode, transmitted messages are compared with the receive rule of the own channel and the messages are stored in a buffer if they have passed through the filter processing. Messages transmitted from other CAN nodes are compared only with the receive rule for which the GAFLLB bit in the RSCAN0GAFLIDj register (j = 0 to 15) is set to 0 (when a message transmitted from another CAN node is received).

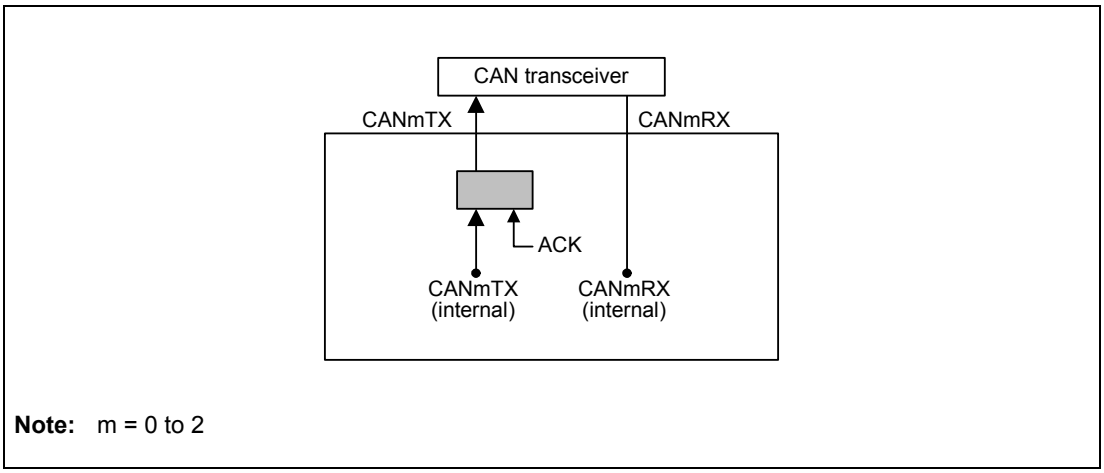
If the mirror function and self-test mode are both enabled, the self-test mode setting takes precedence.

22.9.3.1 Self-Test Mode 0 (External Loopback Mode)

Self-test mode 0 is used to perform a loopback test within a channel including the CAN transceiver.

In self-test mode 0, transmitted messages are handled as messages received through the CAN transceiver and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

Figure 22.13 shows the connection when self-test mode 0 is selected.



Note: m = 0 to 2

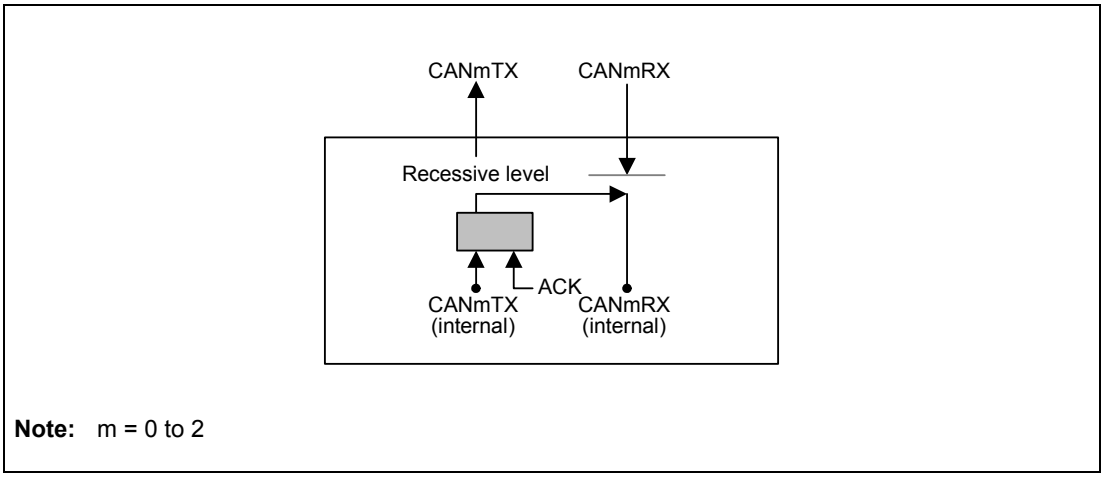
Figure 22.13 Connection when Self-Test Mode 0 is Selected

22.9.3.2 Self-Test Mode 1 (Internal Loopback Mode)

In self-test mode 1, transmitted messages are handled as received messages and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

In self-test mode 1, internal feedback from the internal CANmTX pin (m = 0 to 2) to the internal CANmRX pin is performed. The external CANmRX pin input is isolated. The external CANmTX pin outputs only recessive bits.

Figure 22.14 shows the connection when self-test mode 1 is selected.



Note: m = 0 to 2

Figure 22.14 Connection when Self-Test Mode 1 is Selected

22.9.4 RAM Test

The RAM test function allows accesses to all CAN RAM addresses.

When the RAM test function is used, the RAM is divided into pages of 256 bytes each. RAM test page is set by the RTMPS[6:0] bits in the RSCAN0GTSTCFG register. Data in the set page can be read from and written to the RSCAN0RPGACCr register (r = 0 to 63). The available total RAM size is 7584 bytes ($1DA0_H$).

Do not access more than 160 bytes in the last page (RTMPS[6:0] = $1D_H$) during RAM test.

22.9.5 Inter-Channel Communication Test

The inter-channel communication test function allows communication test by internally connecting CAN channels to each other. During this test, channels are isolated from the external CAN bus.

Before starting data transmission/reception in channel communication mode, make transmission/reception settings for each channel.

Channels not included in the test must be placed in channel halt mode.

Figure 22.15 shows the connection for inter-channel communication test.

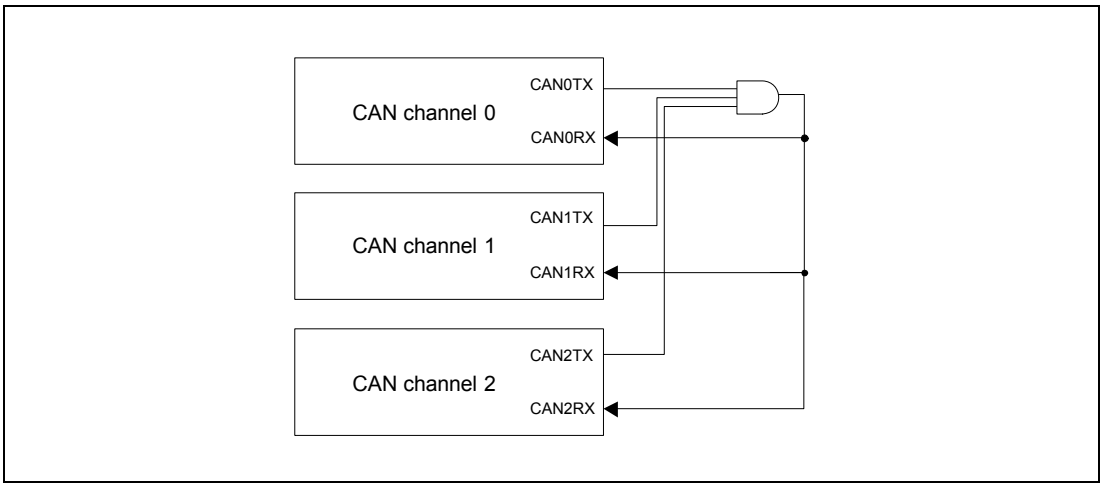
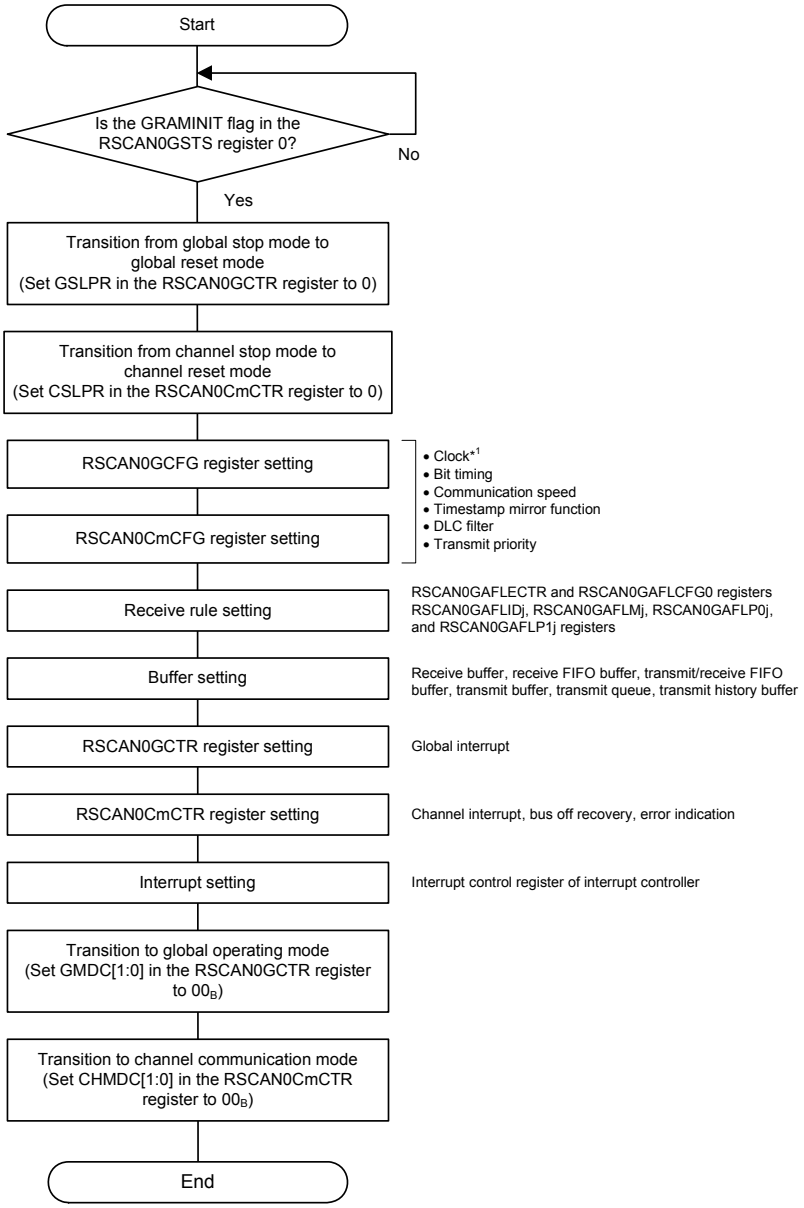


Figure 22.15 Connection for Inter-Channel Communication Test

22.10 RSCAN Setting Procedure

22.10.1 Initial Settings

The RSCAN module initializes the CAN RAM after the MCU is reset. The RAM initialization time is 3794 cycles of the pclk. The GRAMINIT flag in the RSCAN0GSTS register is set to 1 (CAN RAM initialization is ongoing) during the RAM initialization and is cleared to 0 (CAN RAM initialization is finished) when the initialization is completed. Make CAN settings after the GRAMINIT flag is cleared to 0. **Figure 22.16** shows the CAN setting procedure after the MCU is reset.



Note 1. For the setting of CAN clock frequency, see the Table 22.8, Range of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/D1L/D1M.

Note: m = 0 to 2, j = 0 to 15

Figure 22.16 CAN Setting Procedure after the MCU is Reset

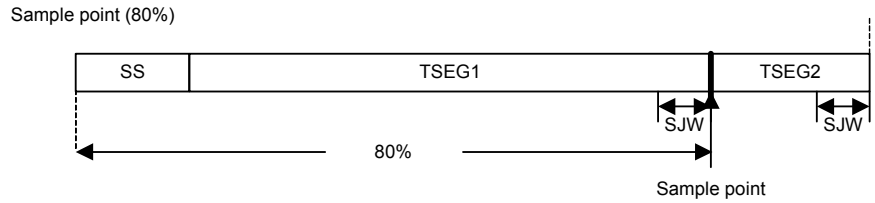
22.10.1.1 Clock Setting

Set the CAN clock (fCAN) as a clock source of the RSCAN module. Select the clk_xincan or clkc using the DCS bit in the RSCAN0GCFG register.

22.10.1.2 Bit Timing Setting

In the CAN protocol, one bit of a communication frame consists of three segments SS, TSEG1, and TSEG2, of which two segments TSEG1 and TSEG2 can be set by the RSCAN0CmCFG register for each channel. Sample point timing can be determined by setting these two segments. This timing can be adjusted in units of 1 Time Quantum (hereafter referred to as Tq). 1 Tq is equal to one CANmTq clock cycle. The CANmTq clock is obtained by selecting the clock source with the DCS bit in the RSCAN0GCFG register and selecting the clock division ratio with the BRP[9:0] bits in the RSCAN0CmCFG register.

Figure 22.17 shows the bit timing chart. **Table 22.95** shows an example of bit timing setting.



SS = 1 Tq fixed
Set TSEG1 to a range of 4 Tq to 16 Tq
Set TSEG2 to a range of 2 Tq to 8 Tq
Set SJW to a range of 1 Tq to 4 Tq
Set SS + TSEG1 + TSEG2 to a range of 8 Tq to 25 Tq
TSEG1 > TSEG2 ≥ SJW

- SS (synchronization segment):
  The SS is a segment that performs synchronization by monitoring the edge from recessive to dominant bits in the Interframe Space.
  Interframe Space consists of Intermission, Suspend Transmission, and Bus Idle. All nodes can start transmission during Bus Idle.
- TSEG1 (Time segment 1):
  TSEG1 is a segment that absorbs physical delay on the CAN network. The physical delay on the CAN network is twice the total of the delay on the CAN bus, the delay in the input comparator, and the delay in the output driver.
- TSEG2 (Time segment 2):
  TSEG2 is a segment that compensates phase error due to an error in frequency.
- SJW (Resynchronization jump width)
  The SJW is a length to extend or reduce the time segment to compensate for an error in phase due to phase error.

Figure 22.17 Bit Timing Chart

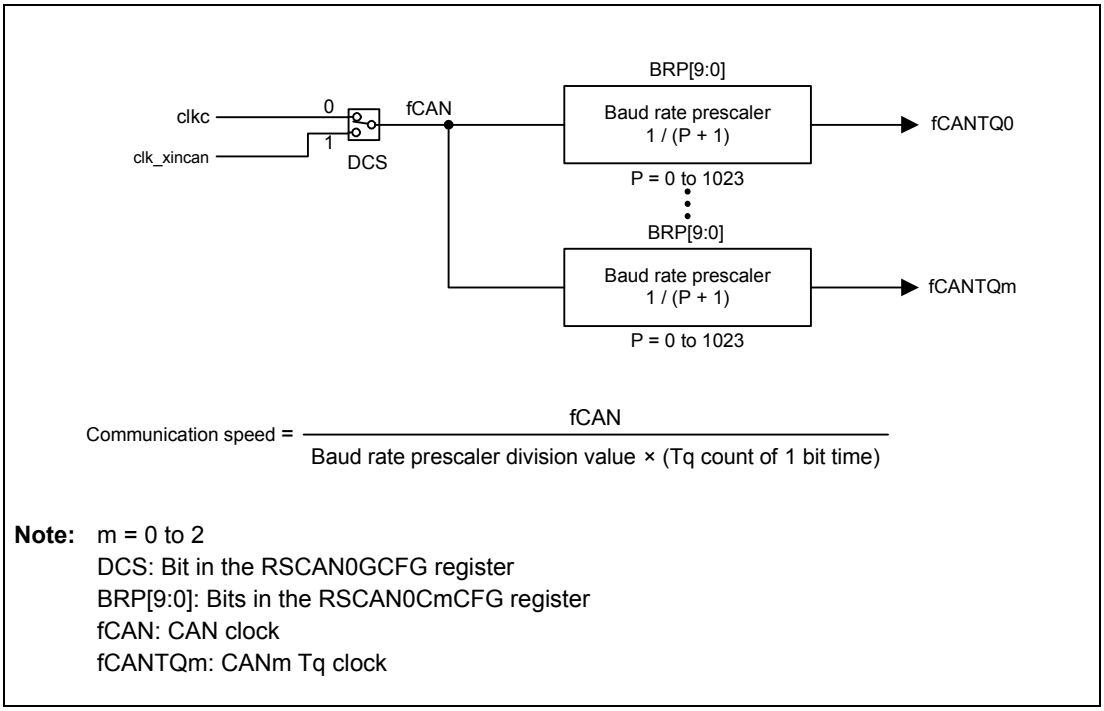
Table 22.95 Example of Bit Timing Setting

1 Bit	Set Value (Tq)				Sample Point (%) Note: See Figure 22.17.
	SS	TSEG1	TSEG2	SJW	
20 Tq	1	12	7	1	65.00
	1	13	6	1	70.00

22.10.1.3 Communication Speed Setting

Set the CAN communication speed for each channel using the fCAN, baud rate prescaler division value (BRP[9:0] bits in the RSCAN0CmCFG register), and Tq count per bit time.

Figure 22.18 shows the CAN clock control block diagram, and **Table 22.96** shows an example of the communication speed setting.



Note: m = 0 to 2
DCS: Bit in the RSCAN0GCFG register
BRP[9:0]: Bits in the RSCAN0CmCFG register
fCAN: CAN clock
fCANTQm: CANm Tq clock

Figure 22.18 CAN Clock Control Block Diagram

Table 22.96 Example of Communication Speed Setting

fCAN / Communication rate	40 MHz
1 Mbps	8 Tq (5) 20 Tq (2)
500 Kbps	8 Tq (10) 20 Tq (4)
250 Kbps	8 Tq (20) 20 Tq (8)

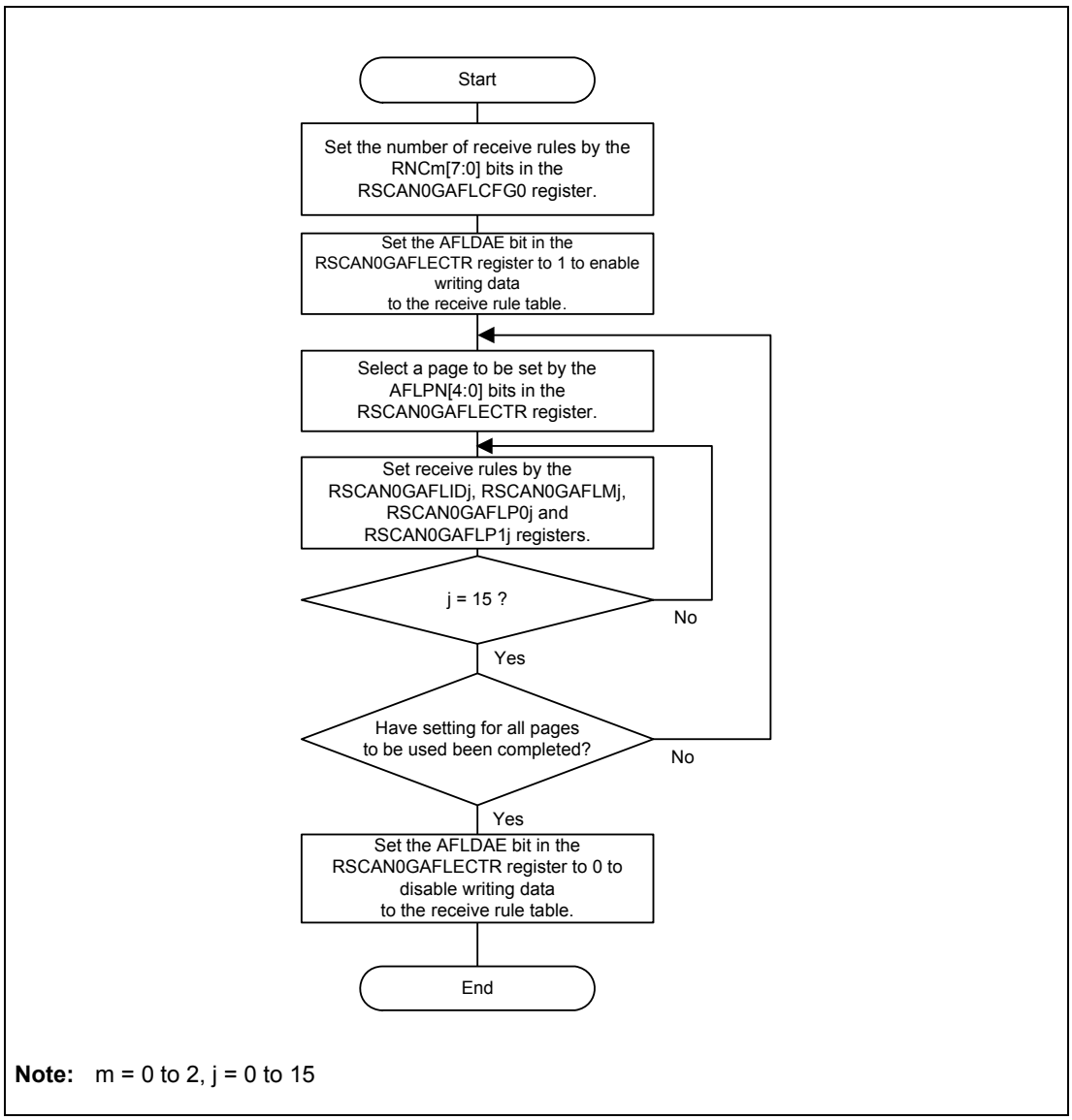
Note: Values in ( ) are baud rate prescaler division values.

22.10.1.4 Receive Rule Setting

Receive rules can be set using receive rule-related registers.

Up to 16 receive rules can be registered per page. Specify pages 0 to 11 by the AFLPN[4:0] bits in the RSCAN0GAFLECTR register. Set receive rule table write enable/disable using the AFLDAE bit.

Figure 22.19 shows the receive rule setting procedure.



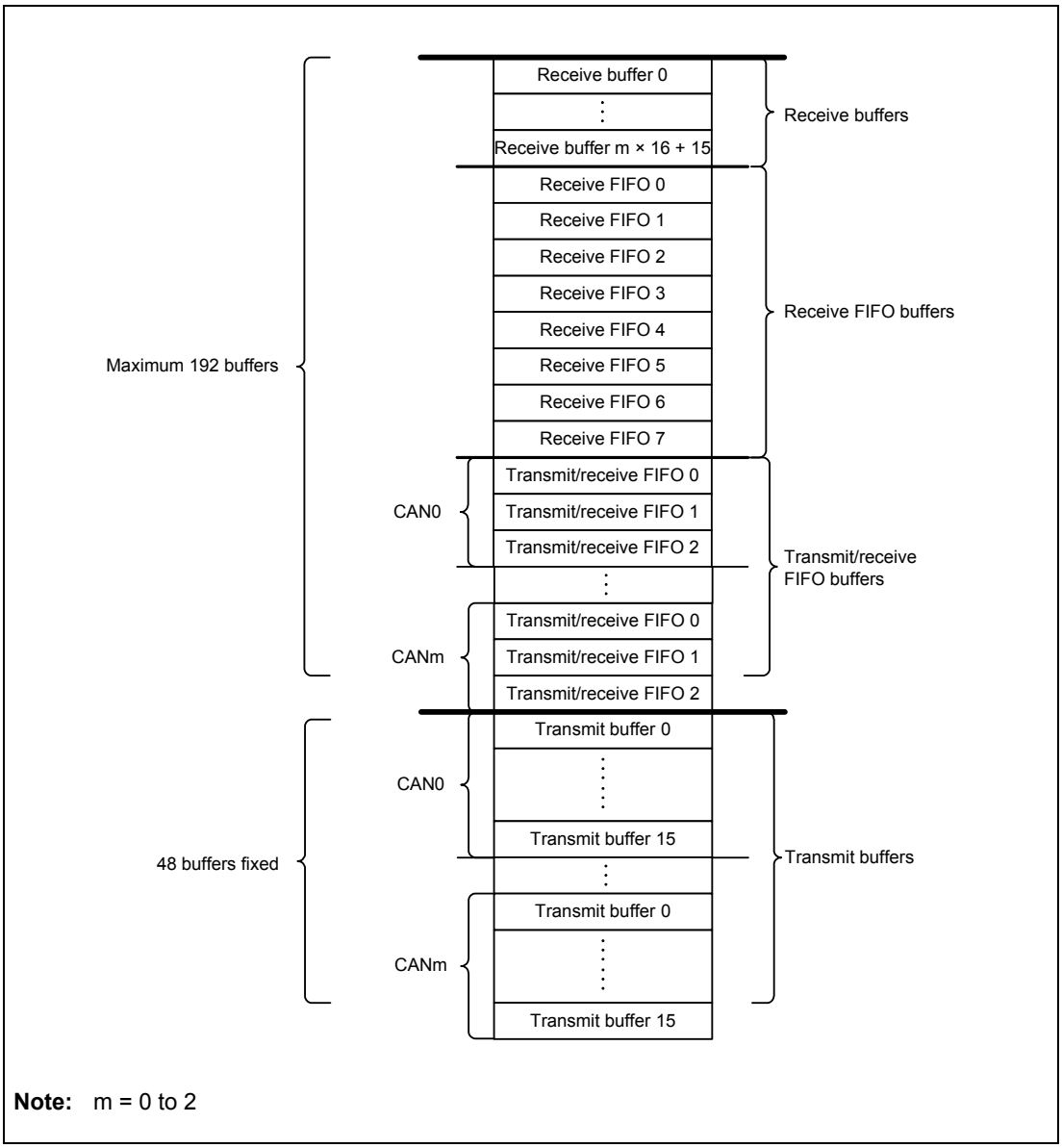
Note: m = 0 to 2, j = 0 to 15

Figure 22.19 Receive Rule Setting Procedure

22.10.1.5 Buffer Setting

Set sizes and interrupt sources of buffers. For transmit/receive FIFO buffers that are set to transmit mode, set transmit buffers to be linked.

Figure 22.20 shows the buffer configuration. **Figure 22.21** shows the buffer setting procedure.



Note: m = 0 to 2

Figure 22.20 Buffer Configuration

CAUTION

Receive buffers, receive FIFO buffers, transmit/receive FIFO buffers, and transmit buffers are located in succession.

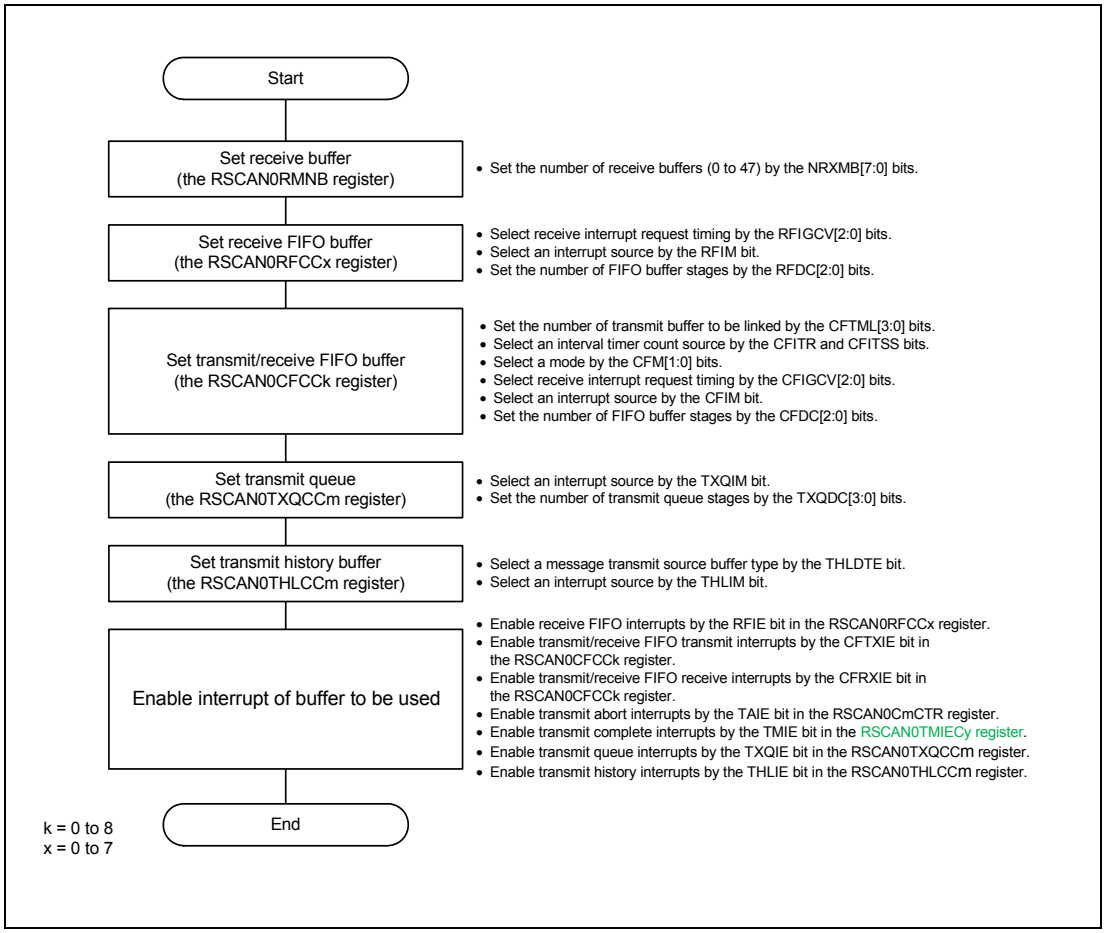
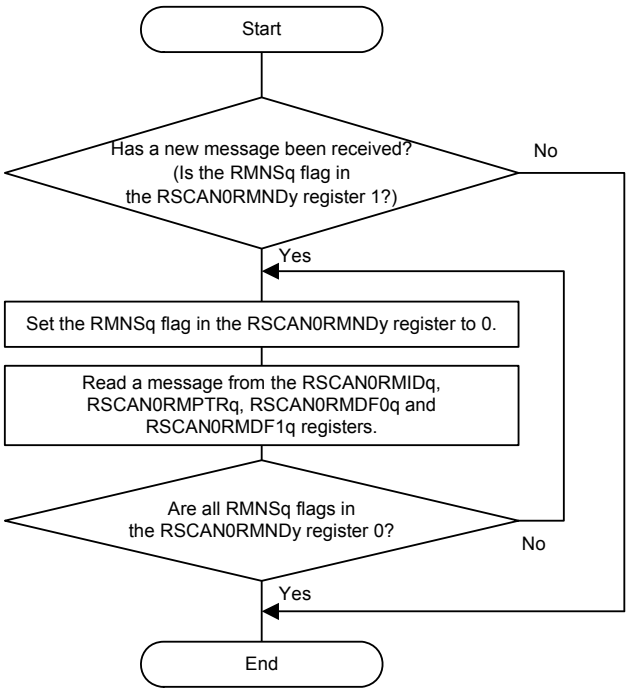


Figure 22.21 Buffer Setting Procedure

22.10.2 Reception Procedure

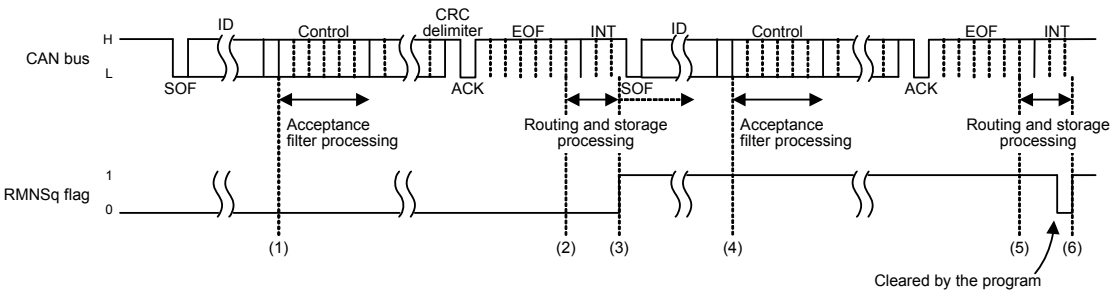
22.10.2.1 Receive Buffer Reading Procedure

When the processing to store received messages in a receive buffer starts, the RMNSq flag in the RSCAN0RMNDy register (y = 0 to 1, q = 0 to 47) is set to 1 (receive buffer q contains a new message). Messages can be read from the RSCAN0RMIDq, RSCAN0RMPTRq, RSCAN0RMDF0q, and RSCAN0RMDF1q registers. If the next message has been received before the current message is read from the receive buffer, the message is overwritten. **Figure 22.22** shows the receive buffer reading procedure.



Note: y = 0, 1, q = 0 to 47

Figure 22.22 Receive Buffer Reading Procedure



Note: y = 0, 1, q = 0 to 47
RMNSq: Flag in the RSCAN0RMNDy register

Figure 22.23 Receive Buffer Reception Timing Chart

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing, the processing to store the message in the specified receive buffer starts.
When the message storage processing starts, the RMNSq flag in the corresponding RSCAN0RMNDy register is set to 1 (the receive buffer contains a new message). If other channels are performing filter processing or transmit priority determination processing, the routing processing and the storage processing may be delayed.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(6) When the corresponding RMNSq flag is cleared to 0 (the receive buffer contains no new message), this flag is set to 1 again when the message storage processing starts. Even if the RMNSq flag remains 1, a new message is overwritten to the receive buffer. The RMNSq flag should not be cleared to 0 during storage of messages.

22.10.2.2 FIFO Buffer Reading Procedure

When received messages have been stored in one or more receive FIFO buffers or a transmit/receive FIFO buffer that is set to receive mode or gateway mode, the corresponding message count display counter (RFMC[7:0] bits in the RSCAN0RFSTSx register (x = 0 to 7) or CFMC[7:0] bits in the RSCAN0CFSTSk register (k = 0 to 8)) is incremented by 1. At this time, when the RFIE bit (receive FIFO interrupt is enabled) in the RSCAN0RFCCx register or the CFRXIE bit (transmit/receive FIFO receive interrupt is enabled) in the RSCAN0CFCCk register is set to 1, an interrupt request is generated. Received messages can be read from the RSCAN0RFIDx, RSCAN0RFPTRx, RSCAN0RFDF0x, and RSCAN0RFDF1x registers for receive FIFO buffers, or from the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers for transmit/receive FIFO buffers. Messages in FIFO buffers can be read sequentially on a first-in, first-out basis.

When the message count display counter value matches the FIFO buffer depth (a value set by the RFDC[2:0] bits in the RSCAN0RFCCx register or the CFDC[2:0] bits in the RSCAN0CFCCk register), the RFFLL or CFFLL flag is set to 1 (the receive FIFO buffer is full).

When all messages have been read out of the FIFO buffer, the RFEMP flag in the RSCAN0RFSTSx register or the CFEMP flag in the RSCAN0CFSTSk register is set to 1 (the receive FIFO buffer contains no unread message (buffer empty)).

If the RFE bit or the CFE bit is cleared to 0 (no receive FIFO buffer is used) with the interrupt request flag (RFIF flag in the RSCAN0RFSTSx register or CFRXIF flag in the RSCAN0CFSTSk register) set to 1 (a receive FIFO interrupt request is present), the interrupt request flag is not automatically cleared to 0. The program must clear the interrupt request flag to 0.

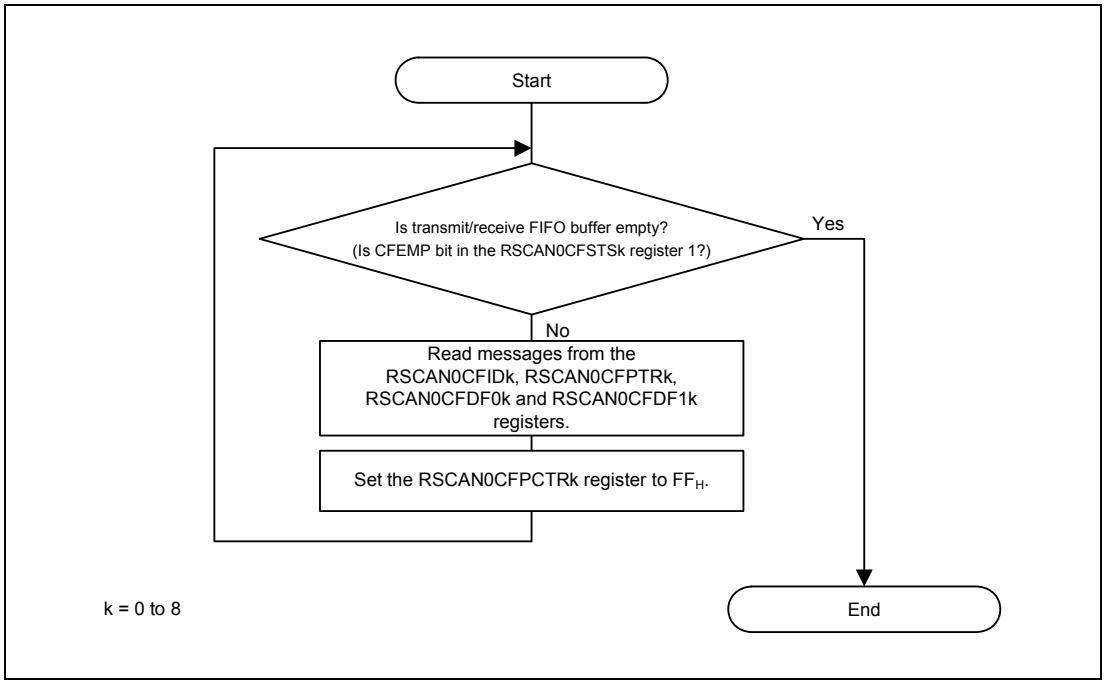


Figure 22.24 Transmit/Receive FIFO Buffer Reading Procedure

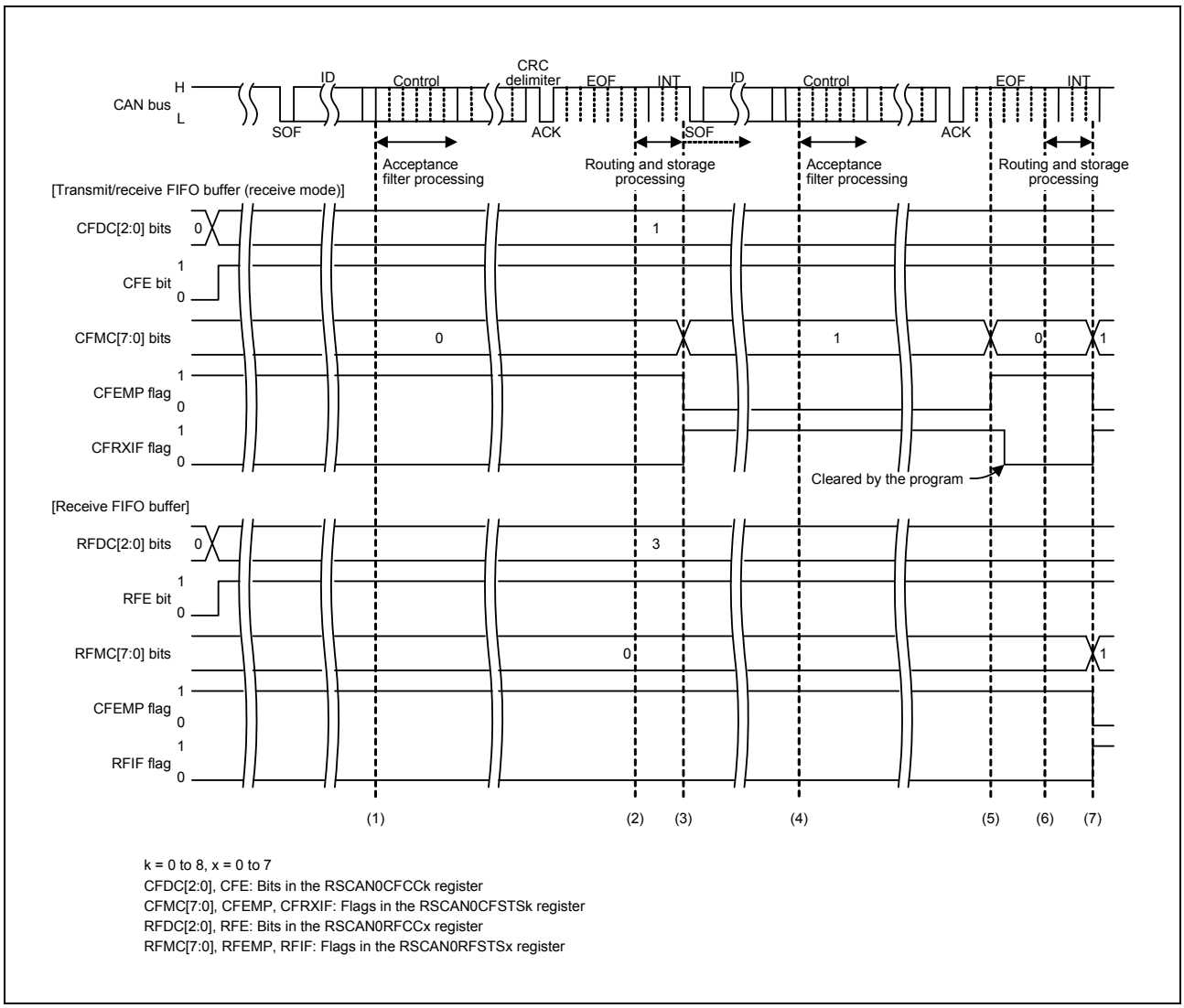


Figure 22.25 FIFO Buffer Reception Timing Chart

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing and the CFE bit in the RSCAN0CFCCk register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCAN0CFCCk register is 001_B or more, the message is stored in the transmit/receive FIFO buffer that is set to receive mode. The CFMC[7:0] value in the RSCAN0CFSTSk register is incremented and becomes 01_H. When the CFIM bit in the RSCAN0CFCCk register is set to 1 (a FIFO receive interrupt request is generated each time a message has been received), the CFRXIF flag in the RSCAN0CFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). The CFRXIF flag can be reset to 0 by the program.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) Read received messages from the RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers and write FF_H to the RSCAN0CFPCTRk register. This causes the

CFMC[7:0] bits in the RSCAN0CFSTSk register to be decremented. When CFMC[7:0] becomes 00_H, the CFEMP flag in the RSCAN0CFSTSk register becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

(6) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCAN0GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(7) The message is stored in the transmit/receive FIFO buffer set in receive mode when the message has passed through the DLC filter process if the CFE bit is set to 1 (transmit/receive FIFO buffers are used), and the CFDC[2:0] bits are set to 001_B or more. The CFMC[7:0] bit value is incremented by 1 to be 01_H. When the CFIM bit is set to 1 (an interrupt occurs each time a message has been received), the CFRXIF flag is set to 1 (a transmit/receive FIFO receive interrupt request is present).
The message is stored in the receive FIFO buffer if the RFE bit in the RSCAN0RFCCx register is set to 1 (receive FIFO buffers are used), and the RFDC[2:0] bits in the RSCAN0RFCCx register are set to 001_B or more. The RFMC[7:0] bits in the RSCAN0RFSTSx register are set to 01_H by being incremented by 1. When the RFIM bit in the RSCAN0RFCCx register is set to 1 (an interrupt occurs each time a message has been received), the RFIF flag in the RSCAN0RFSTSx register is set to 1 (a receive FIFO interrupt request is present).

22.10.3 Transmission Procedure

22.10.3.1 Procedure for Transmission from Transmit Buffers

Figure 22.26 shows the procedure for transmission from transmit buffers.

Figure 22.27 shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmission has been successfully completed. **Figure 22.28** shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmit abort has been completed.

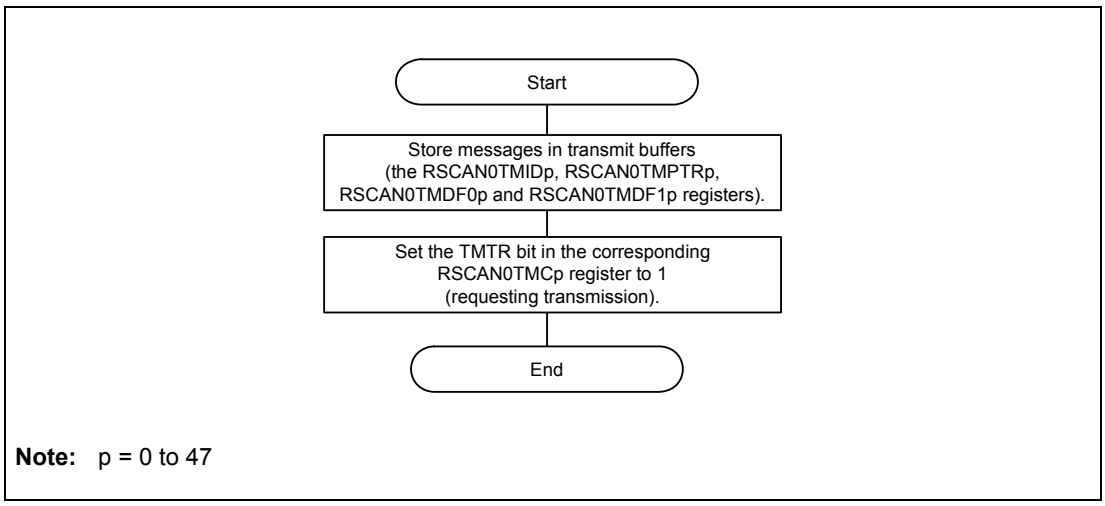
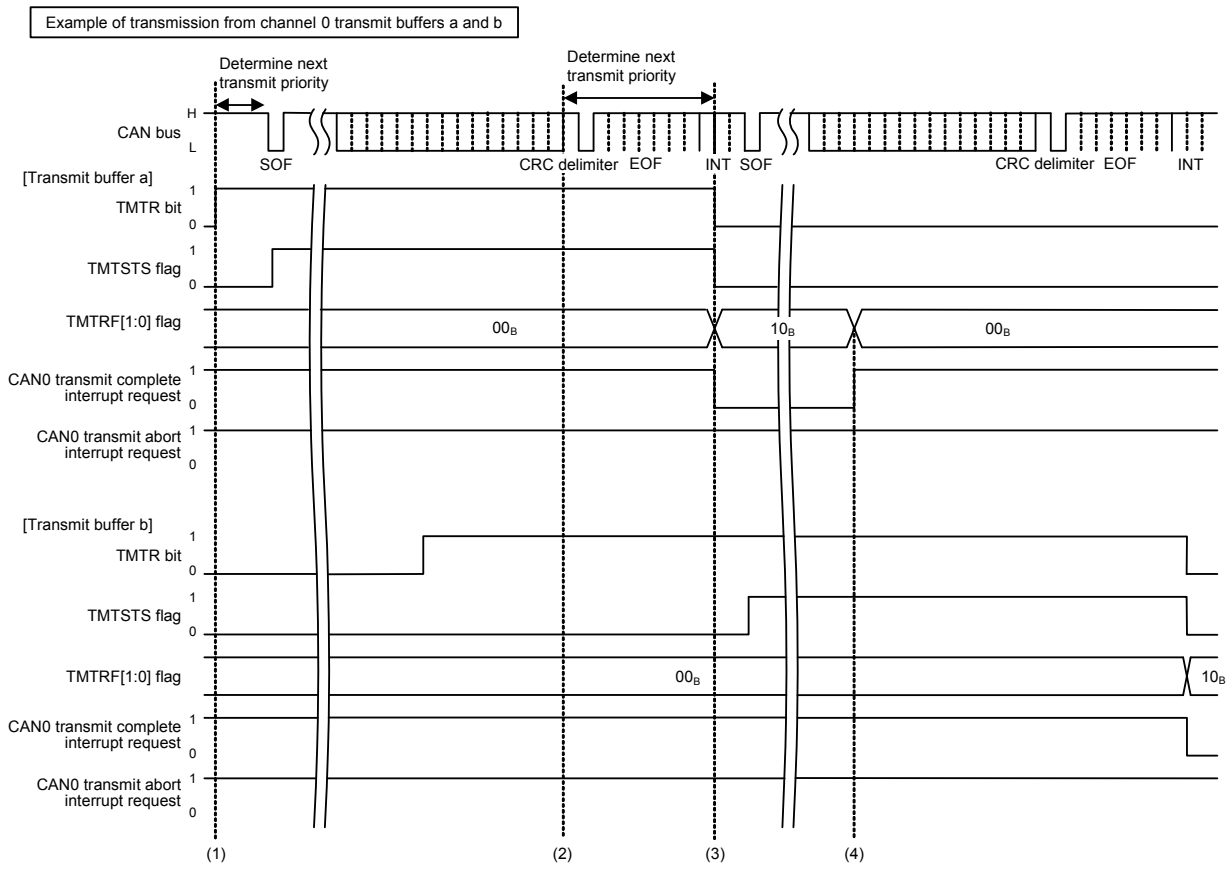


Figure 22.26 Procedure for Transmission from Transmit Buffers



Note: a = 0 to 47, b = 0 to 47
TMTR: Bit in the RSCAN0TMCp register
TMTSTS, TMTRF[1:0]: Flags in the RSCAN0TMSTSp registers

Figure 22.27 Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully)

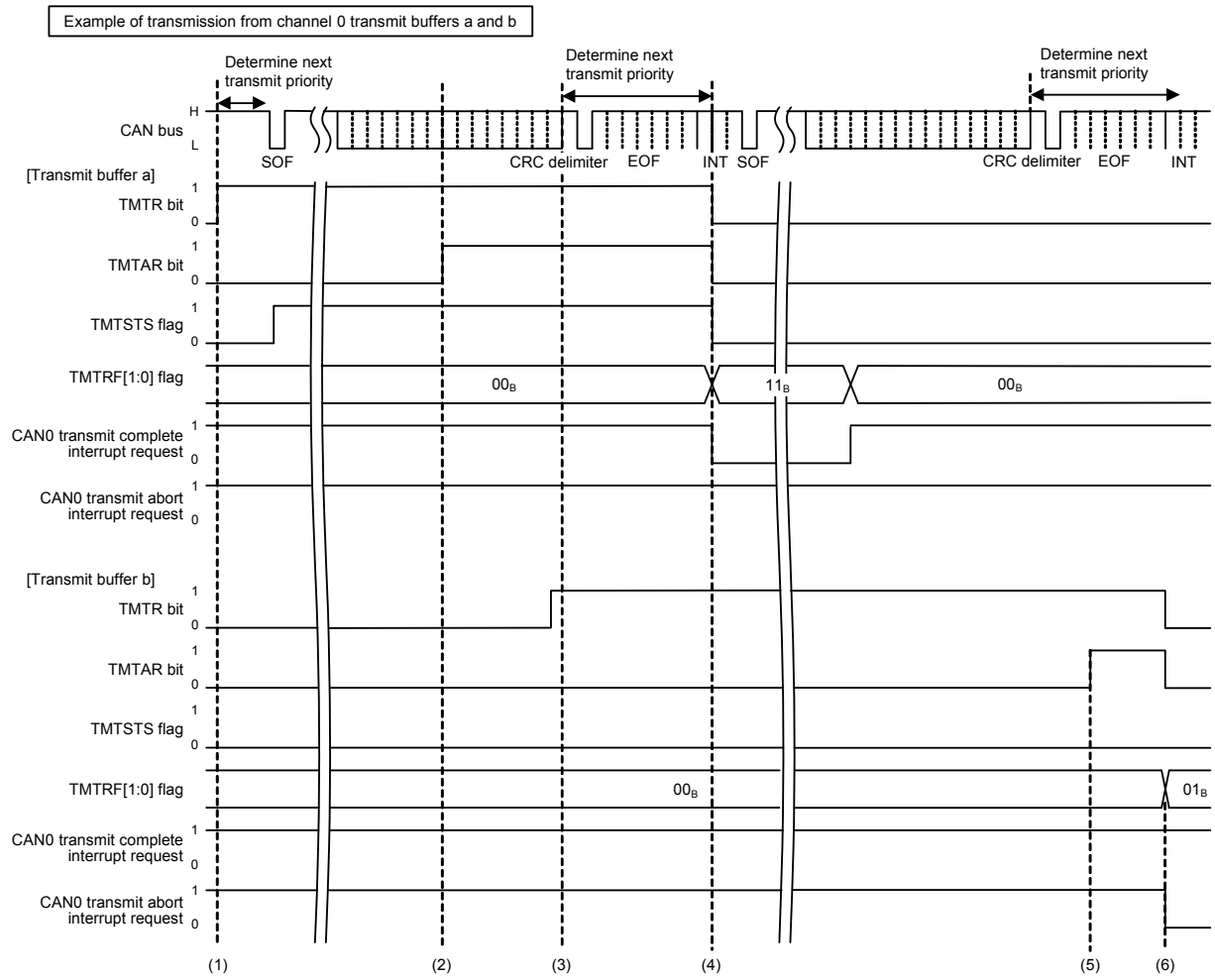
(1) When the TMTR bit in the RSCAN0TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCAN0TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.

(2) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(3) When transmission completes successfully, the TMTRF[1:0] flag in the RSCAN0TMSTSa register is set to 10_B (transmission has been completed (without transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCAN0TMCa register are cleared to 0. When the TMIEa bit in the RSCAN0TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).

(4) Before starting the next transmission, set the TMTRF[1:0] flag to 00_B. Write the next message to the transmit buffer, and then set the TMTR bit to 1 (transmission is requested). The TMTR bit can be set to 1 only when the TMTRF[1:0] flag value is 00_B.

If an arbitration-lost has occurred after transmission is started, the TMTSTS flag is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.



Note: a = 0 to 47, b = 0 to 47
TMTR, TMTAR: Bits in the RSCAN0TMCp register
TMTSTS, TMTRF[1:0]: Flags in the RSCAN0TMSTSp register

Figure 22.28 Transmit Buffer Transmission Timing Chart (Transmit Abort Completed)

(1) When the TMTR bit in the RSCAN0TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCAN0TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.

(2) When it is determined that the transmit buffer is used for the next transmission or transmission is in progress, message transmission is not aborted unless an error or arbitration loss occurs even if the TMTAR bit is set to 1 (transmit abort is requested).

(3) The priority determination starts with the CRC delimiter for the next transmission. In this timing chart, buffer b is not selected as the next transmit buffer. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(4) When transmission completes successfully, the TMTRF[1:0] flag in the RSCAN0TMSTSa register is set to 11_B (transmission has been completed (with transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCAN0TMCa register are cleared to 0. When the TMIEa value in the RSCAN0TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).

(5) While another CAN node is transmitting data on the CAN bus (TMTSTS flag = 0), if the TMTAR bit is set to 1 while the corresponding channel is determining transmit priority, the TMTR bit cannot be cleared to 0.

(6) After the internal processing time has passed, the transmission is terminated and the TMTRF[1:0] flag is set to 01_B. When the transmit buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not being made, an abort request is immediately accepted and the TMTRF[1:0] flag is set to 01_B. At this time, the TMTR and TMTAR bits are cleared to 0. When transmit abort is completed with the TAIE bit in the RSCAN0CmCTR register set to 1 (transmit abort interrupt is enabled), an interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B.

If an arbitration loss has occurred after the CAN channel started transmission, the TMTSTS bit is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC delimiter to find the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.

22.10.3.2 Procedure for Transmission from Transmit/Receive FIFO Buffers

Figure 22.29 shows the procedure for transmission from transmit/receive FIFO buffers.

Figure 22.30 shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmission has been successfully completed. **Figure 22.31** shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmit abort has been completed.

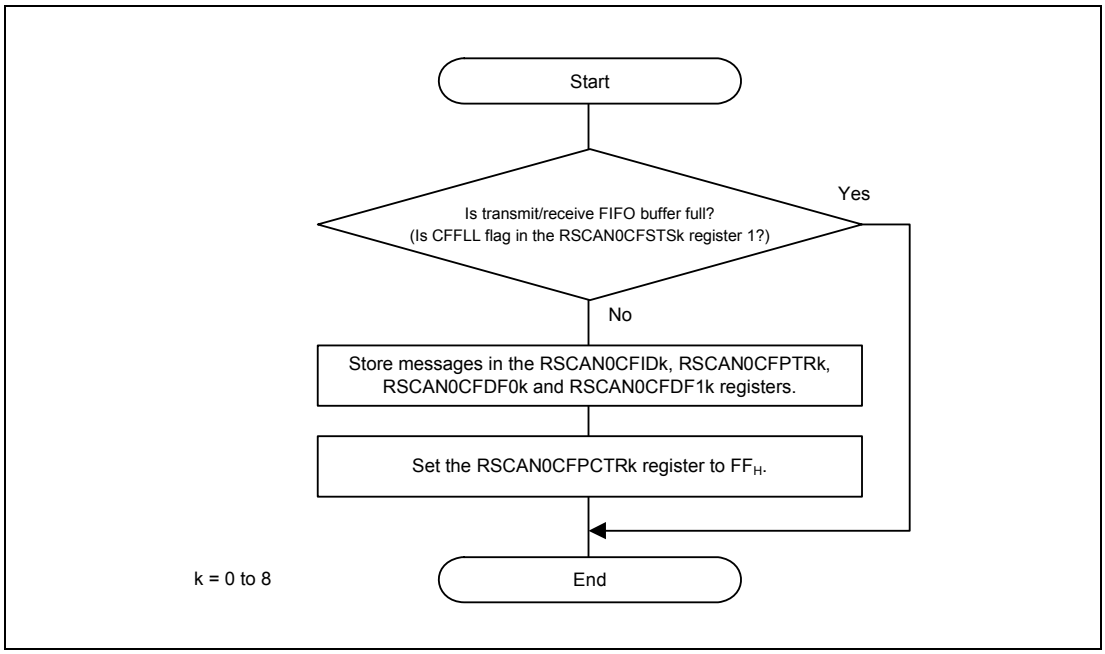


Figure 22.29 Procedure for Transmission from Transmit/Receive FIFO Buffers

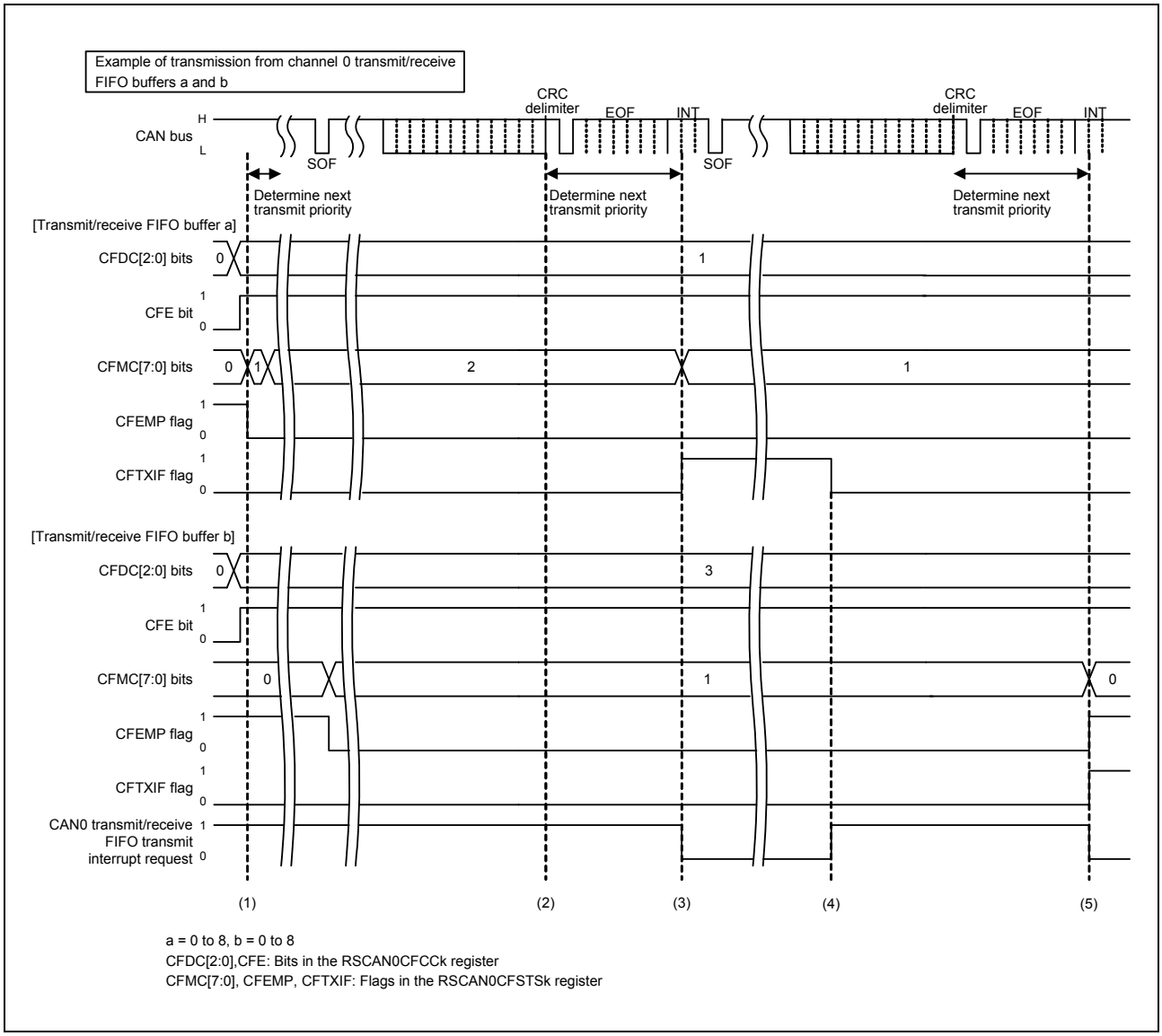


Figure 22.30 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully)

(1) While the CAN bus is idle, when the CFE bit in the RSCA0CFCCa register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCAN0CFCCa register is 001_B (4 messages) or more and the CFMC[7:0] value in the RSCAN0CFSTSa register is 01_H or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer a of channel 0.

(2) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(3) When transmission completes successfully, the CFMC[7:0] value in the RSCAN0CFSTSa register is decremented by 1. Setting the CFIM bit in the RSCAN0CFCCa register to 1 (a FIFO transmit

interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RSCAN0CFSTSk register to 1 (a transmit/receive FIFO transmit interrupt request is present).

(4) The program can clear the CFTXIF flag.

(5) Message transmission from transmit/receive FIFO buffer b of channel 0 has been completed and the CFMC[7:0] value in the RSCAN0CFSTSb register is decremented by 1. The CFMC[7:0] bits are cleared to 00_H and therefore the CFEMP flag in the RSCAN0CFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

Transmission is continued until the CFEMP flag is set to 1. It is possible to continuously store transmit messages in FIFO buffers until the CFFLL flag in the RSCAN0CFSTSa and RSCAN0CFSTSb register is set to 1 (the transmit/receive FIFO buffer is full).

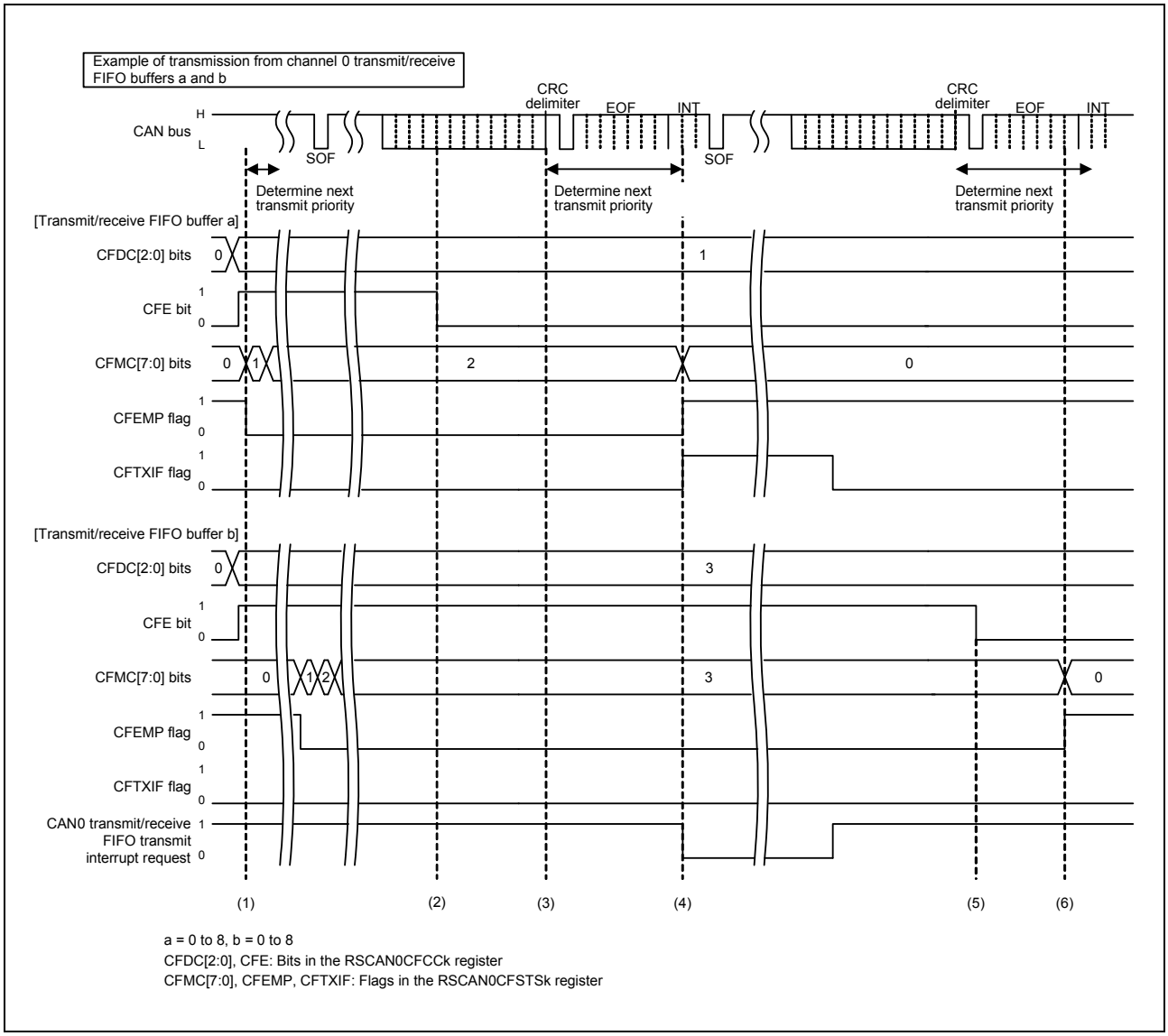


Figure 22.31 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)

(1) While the CAN bus is idle, when the CFE bit in the RSCAN0CFCCa register (a = 0 to 8) is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCAN0CFCCa register is 001_B (4 messages) or more and the CFMC[7:0] value in the RSCAN0CFSTSa register is 01_H or more, the priority determination processing starts to determine the highest-priority transmit

message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer a of channel 0.

(2) When transmission is in progress or it is determined that the transmit/receive FIFO buffer is used for the next transmission, message transmission is not aborted unless an error or arbitration loss occurs even if the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

(3) When a transmit request from a buffer is present, the priority determination starts with the CRC delimiter for the next transmission. In this figure, transmit/receive FIFO buffer b is not selected as a buffer for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(4) When transmit completes successfully, the CFMC[7:0] value is cleared to 00_H. Setting the CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RSCAN0CFSTSa register to 1 (a transmit/receive FIFO transmit interrupt request is present). The program can clear the CFTXIF flag.

(5) If another CAN node on the CAN bus is transmitting data (not from transmit/receive FIFO buffer b), transmit/receive FIFO buffers cannot be disabled immediately even if the CFE bit in the RSCAN0CFCCb register is cleared to 0 (no transmit/receive FIFO buffer is used) during transmit priority determination. (The CFEMP flag in the RSCAN0CFSTSb register is not set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) immediately.)

(6) After the internal processing time has passed, transmit/receive FIFO buffers are disabled and the CFMC[7:0] bits in the RSCAN0CFSTSb register are cleared to 00_H and the CFEMP flag is set to 1. When the transmit/receive FIFO buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not in progress, the transmit/receive FIFO buffer is immediately disabled. (The CFMC[7:0] bits are cleared to 00_H and the CFEMP flag is set to 1.)

22.10.3.3 Procedure for Transmission from the Transmit Queue

Figure 22.32 shows the procedure for transmission from the transmit queue.

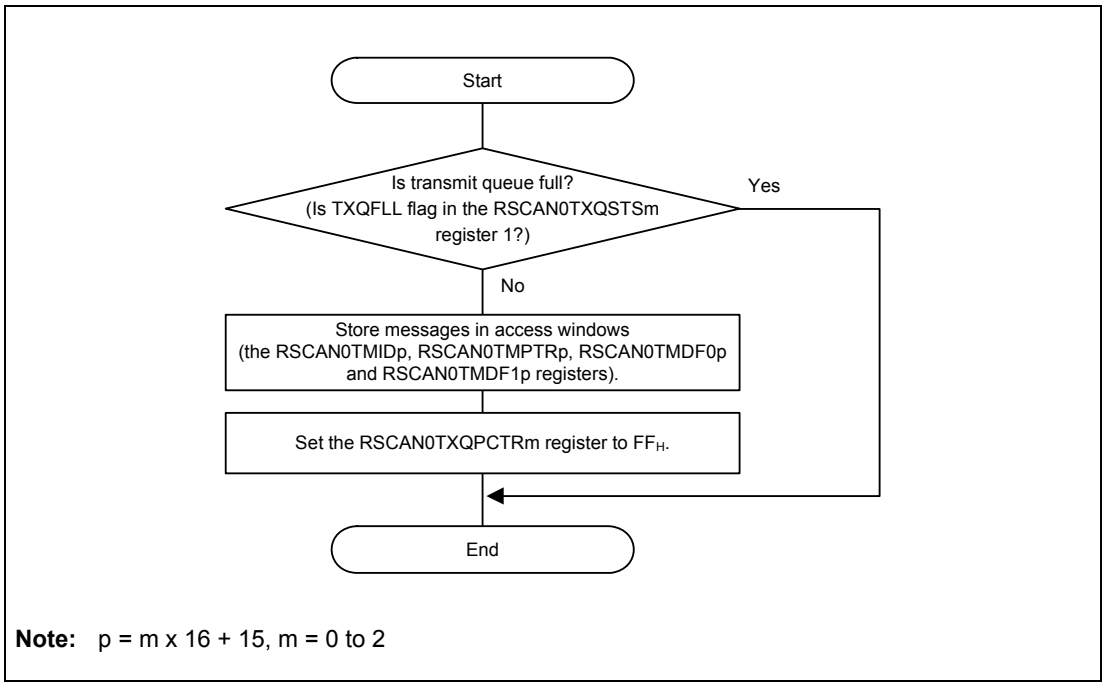


Figure 22.32 Procedure for Transmission from the Transmit Queue

22.10.3.4 Transmit History Buffer Reading Procedure

Transmit history data can be read from the RSCAN0THLACCm register. The next data can be accessed by writing FF_H to the corresponding RSCAN0THLPCTRm register (m = 0 to 2) after reading a set of data. **Figure 22.33** shows the transmit history buffer reading procedure.

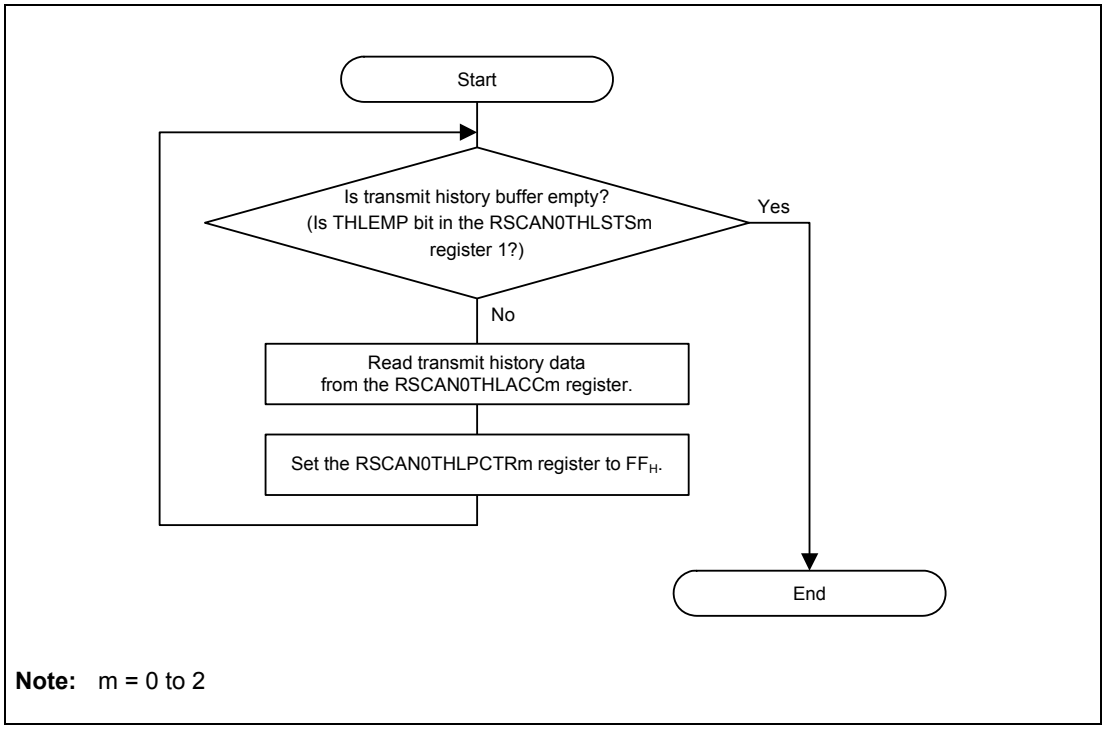


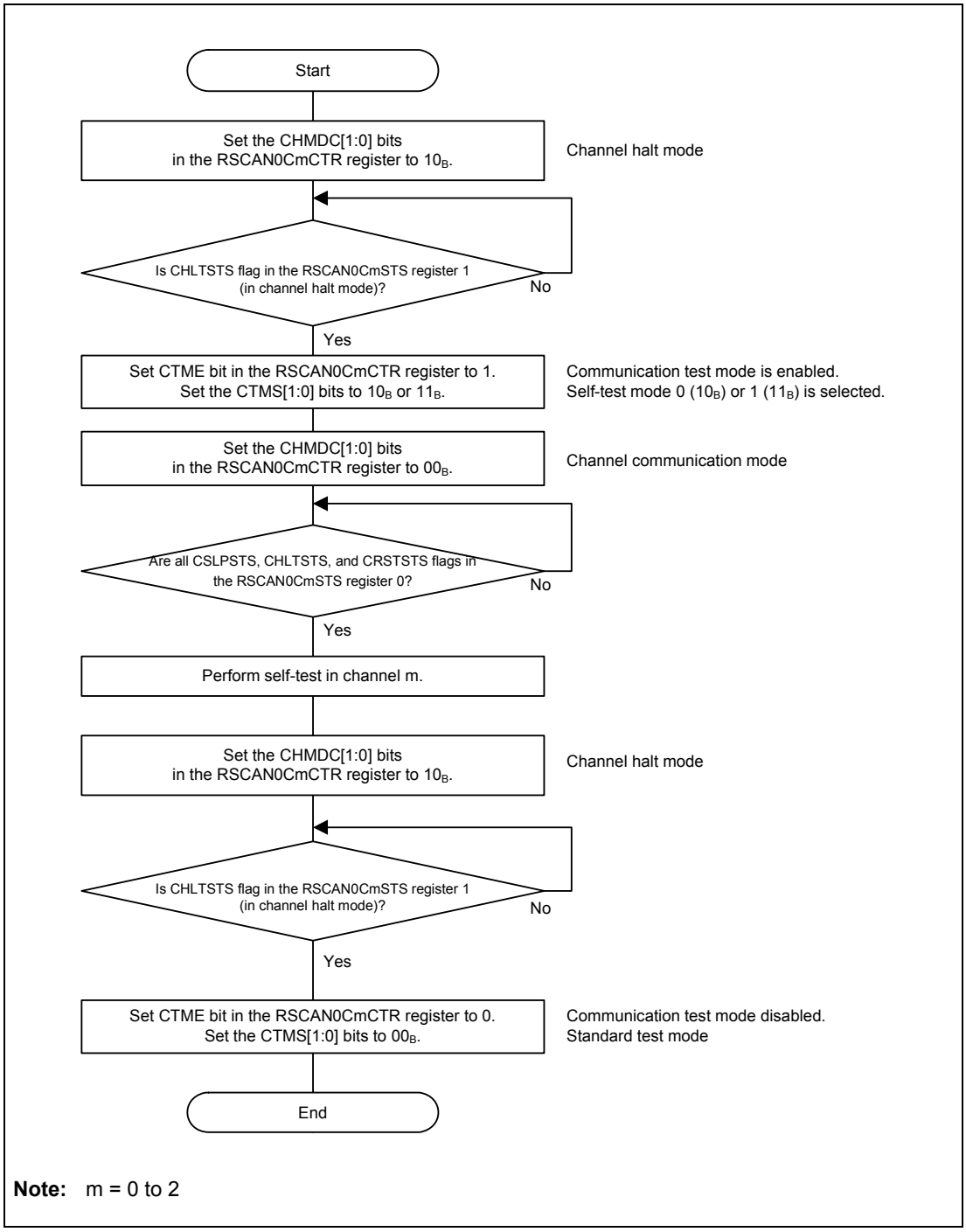
Figure 22.33 Transmit History Buffer Reading Procedure

22.10.4 Test Settings

22.10.4.1 Self-Test Mode Setting Procedure

Self-test mode allows communication test on a channel basis by enabling a CAN node to receive its own transmitted messages.

Figure 22.34 shows the self-test mode setting procedure.



Note: m = 0 to 2

Figure 22.34 Self-Test Mode Setting Procedure

22.10.4.2 Procedure for Releasing the Protection

Since the global test function in **Table 22.97** is protected, write the protection release data 1 and release data 2 in succession to the LOCK[15:0] bits in the RSCAN0GLOCKK register, then set the target test bit to 1.

Table 22.97 Protection Release Data for Test Function

Test Function	Protection Release Data 1	Protection Release Data 2	Target Bit
RAM test	7575_H	$8A8A_H$	RTME bit in the RSCAN0GTSTCTR register

If an incorrect value is written to the LOCK[15:0] bits, restart from writing the protection release data 1. **Figure 22.35** shows the procedure for releasing the protection.

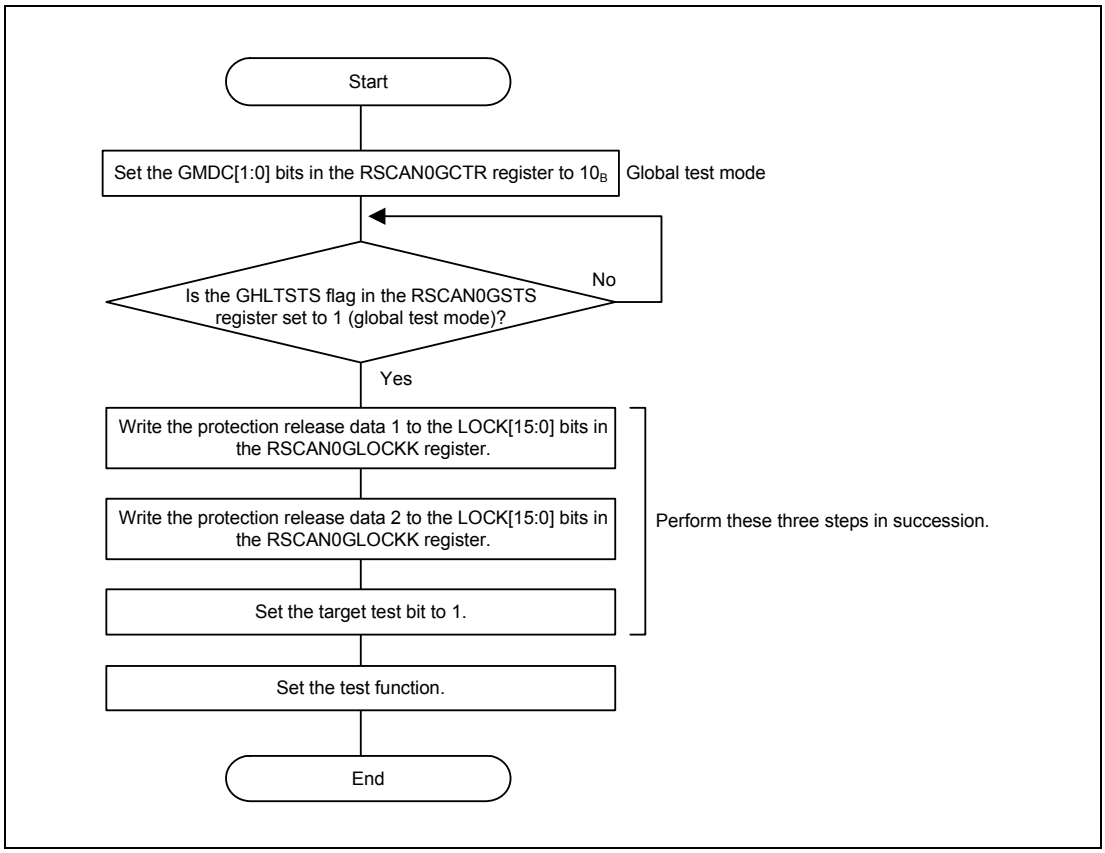


Figure 22.35 Protection Release Procedure

22.10.4.3 RAM Test Setting Procedure

RAM tests include CAN RAM read/write test. The read/write test verifies that data written to the RAM is read correctly. Before closing the RAM test, write 0000 0000_H to all pages of the CAN RAM.

Figure 22.36 shows the RAM test setting procedure.

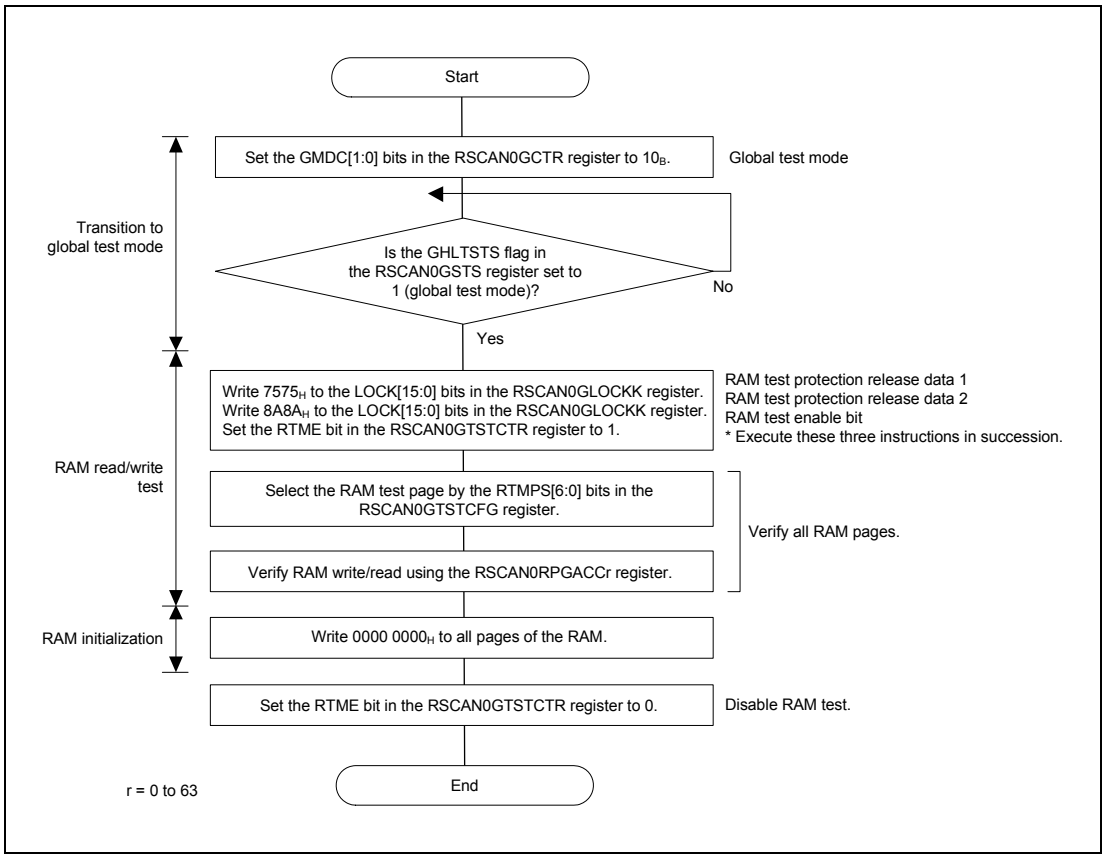
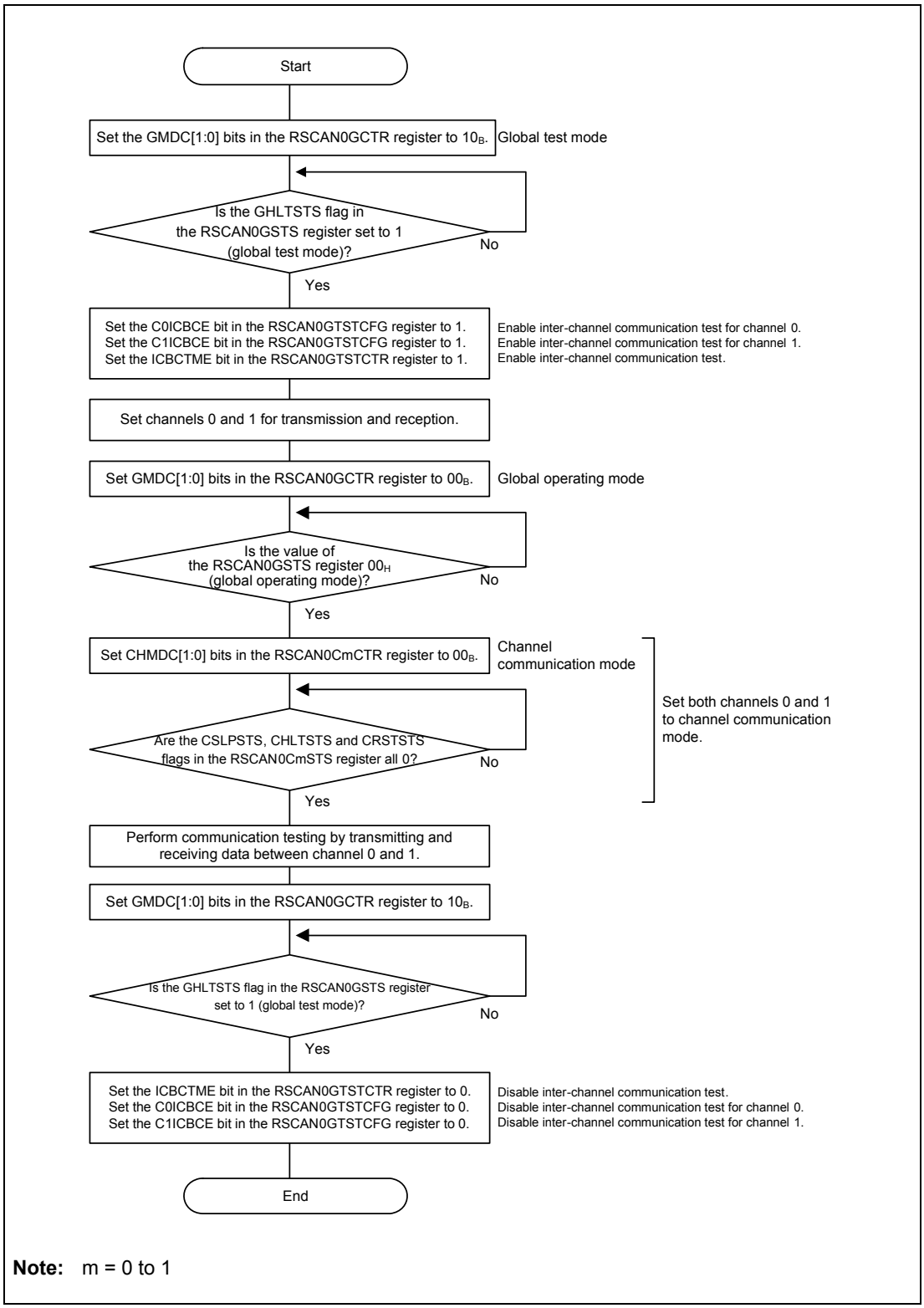


Figure 22.36 RAM Test Setting Procedure

22.10.4.4 Inter-Channel Communication Test Setting Procedure

Communication testing can be performed by transmitting and receiving data between different channels.

Figure 22.37 shows the inter-channel communication test setting procedure.



Note: m = 0 to 1

Figure 22.37 Inter-Channel Communication Test Setting Procedure (Example of Communication Test between Channel 0 and Channel 1)

22.11 Detection and Correction of Errors in RSCAN RAM

22.11.1 ECC for the RSCAN0 RAM

Table 22.98 gives an outline of the ECC functions for the RSCAN0 RAM.

Table 22.98 List of the ECC Functions for the RSCAN0 RAM

Item	Outline of Functions
ECC error detection/correction	The RAM is checked for ECC errors. The following options are selectable. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection The ECC error detection/correction can be disabled by using through mode. With the initial settings, error detection/correction is enabled.
Error notification	When an ECC 2-bit error is generated, the error is notified. • Error notification can be enabled or disabled when an ECC 2-bit error is detected. In the initial setting, 2-bit error notification is enabled.
Error status	Monitoring for the detection of two-bit ECC errors and for the detection of one-bit ECC errors is available. A register for clearing the error status is provided.
Address capture	• Only one address at which an ECC error has occurred can be captured. • A signal is generated upon detection of a 2-bit or ECC 1-bit error, and the signal is used as a trigger to capture the error-causing address (when the first (1-bit or 2-bit) error is detected after the flag is cleared).

CAUTION

When ECC error detection/correction is performed, confirm initialization of the RSCAN0 RAM (RSCAN0GSTS.GRAMINIT = 0) before it is used.

NOTE

There are two ECC decoders for the RSCAN0 RAM.
One is for page 0 to 11 of RSCAN0 RAM.
The other is for page 12 to 29.
In case of testing ECC, test two decoders.

22.11.2 Interrupt Request

Table 22.99 lists the ECC interrupt request of RSCAN0 RAM.

Table 22.99 Interrupt Requests

Unit Interrupt Signal	Outline	Connected to
—	RSCAN ECC 2 bit error interrupt	Error Control Module INTECCDPERIRAM
—	RSCAN ECC 1 bit error interrupt	Error Control Module INTECCSPERIRAM

22.11.3 ECCRCAN0CTL — RSCAN0 ECC Control Register

The ECCRCAN0CTL register controls the mode of the ECC and the status for RSCAN0.

Bits 7, 5 and 4 should be set (written) while the RSCAN0 operation is stopped.

In addition, when writing to bit 7, EMCA1 and EMCA0 need to be 01_B.

Access: This register can be read/written in 16-bit units.

Address: FFC7 1000_H

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA1	EMCA0	—	—	—	ECER2C	ECER1C	—	ECTHM	ECERVF	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	Undefined
R/W	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R	R/W	R/W	R/W	R/W	R/W	R	R	R

Note 1. These bits are always read as 0.

Table 22.100 ECCRCAN0CTL Register Contents (1/2)

Bit position	Bit Name	Function
15	EMCA1	Access control bits 1 and 0 to ECC mode selection bit These bits specify whether modifying the ECTHM bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 01_B, writing to bit 7 is enabled.
14	EMCA0	
13 to 11	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
10	ECER2C	2-bit ECC error detection flag clear bit This bit is used to clear the 2 bit error detection flag of ECER2F (bit 2). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER2F bit is set to clear the ECER2F bit. When a conflict between this bit writing and ECER2F bit setting occurs, writing to this bit has a priority.
9	ECER1C	1-bit ECC error correction accumulation flag clear bit This bit is used to clear the 1 bit error detection/correction flag of ECER1F (bit 1). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER1F bit is set to clear the ECER1F bit. When a conflict between this bit writing and ECER1F bit setting occurs, writing to this bit has a priority.
8	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
7	ECTHM	ECC function through mode selection bit This bit is used to set enabling and disabling of ECC. Setting this bit to 1 disables ECC function. When writing to this bit, (0, 1) must be written to (EMCA1, EMCA0) at the same time. 0: Through mode is disabled (normal operation mode). 1: Through mode is enabled. (ECC function disable)
6	ECERVF	ECC error judgment enable 0: error judgement disabled 1: error judgement enabled Error judgement is only effective in normal operation mode, i.e. if ECCRCAN0CTL.ECTHM = 0. If error judgement is disabled no interrupts are asserted in case of a single or double bit error detection. However detected single bit errors are corrected. Modification of this bit is only possible, if EMCA[1:0] = 01_B. Otherwise any write to this bit is ignored.

Table 22.100 ECCRCAN0CTL Register Contents (2/2)

Bit position	Bit Name	Function
5	EC1ECP	1-bit error correction enable bit This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected.
4	EC2EDIC	2-bit error detection interrupt control bit This bit controls whether to generate an interrupt when 2-bit error is detected. 0: When 2-bit error is detected, a INTECCDCNRAM interrupt will not be generated. 1: When 2-bit error is detected, a INTECCDCNRAM interrupt will be generated. (initial value) When ECC 2-bit detection is used, EC2EDIC must be set to 1.
3	EC1EDIC	1-bit error detection interrupt control bit This bit controls whether to generate an interrupt when 1-bit error is detected. 0: When 1-bit error is detected, a INTECCSPERIRAM interrupt will not be generated. 1: When 1-bit error is detected, a INTECCSPERIRAM interrupt will be generated. When ECC 1-bit detection is used, EC1EDIC must be set to 1.
2	ECER2F	2-bit error detection flag bit This flag indicates whether 2-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). When 2-bit error interrupt is enabled (EC2EDIC = 1) and this flag is set, an ECC 2-bit error interrupt (INTECCDCNRAM) is output. Write 1 to the ECER2C bit (bit 10) to clear the flag. This bit is cleared at the time of through mode enable selection (ECTHM = 1). If 2-bit error is detected again while this bit is set, an interrupt signal will not be generated. 0: 2-bit error has not occurred since this bit was cleared. 1: 2-bit error has occurred. When ECC 2-bit detection is used, EC2EDIC must be set to 1.
1	ECER1F	1-bit error detection/correction flag bit This flag indicates whether 1-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). Write 1 to the ECER1C bit (bit 9) to clear the flag. This bit is cleared at the time of through mode enable selection (ECTHM = 1). 0: 1-bit error has not occurred since this bit was cleared. 1: 1-bit error has occurred. When ECC 1-bit detection is used, EC1EDIC must be set to 1.
0	ECEMF	ECC error message flag This flag indicates whether an error exists in the current read data. This bit is updated whenever the RAM reads data. This bit might be set if it is read before initialization of the RAM. This bit is cleared at the time of through mode enable selection (ECTHM = 1) and when there is no 1 bit error in the decode circuit input data. 0: The currently-read RAM data does not have bit errors. 1: The currently-read RAM data have bit errors.

CAUTION

Bits 2 and 1 should be cleared when the ECC error message flag (ECEMF) is not set.

We recommend initializing the RAM before clearing bits 2 and 1.

22.11.4 ECCRCAN0TMC — RSCAN0 ECC Test Mode Control Register

The ECCRCAN0TMC register switches to and controls the test mode.

This register can be used when RSCAN is not accessed to RAM.

Access: This register can be read or written in 16-bit units.

Address: FFC7 1004_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA1	ETMA0	—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. These bits are always read as 0.

Table 22.101 ECCRCAN0TMC Register Contents (1/2)

Bit position	Bit Name	Function
15	ETMA1	Access control bits 1 and 0 to ECC test mode bit These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 10_B, writing to bit 7 is enabled.
14	ETMA0	
13 to 8	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
7	ECTMCE	ECC test mode enable bit This bit specifies whether to enable access to the test registers and test control bits. When writing to this bit, (1, 0) should be written to (ETMA1, ETMA0) at the same time. 0: Access to the test mode registers and bits is disabled. 1: Access to the test mode registers and bits is enabled. Test registers: ECCRCAN0TED, ECCRCAN0TRC, ECCRCAN0SYND, ECCRCAN0HORD, ECCRCAN0ECRD, ECCRCAN0ERDB Register test control bit: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
4	ECTRRS	ECC RAM read test mode selection bit Select reading destination when reading ECCRCAN0TED register and reading destination when reading ECCRCAN0ERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0:The read value of the ECCRCAN0TED register is the write value of the ECCRCAN0TED register. The read value of the ECCRCAN0ERDB register is the write value of the ECCRCAN0ERDB register. 1: The read value of the ECCRCAN0TED register can read RAM data. The read value of the ECCRCAN0ERDB register is the ECC data to be written to RAM.
3	ECREOS	ECC redundant bit output data selection bit This bit is used to identifyand select the ECC data to be stored in RAM as ECC data generated for write data or the value of the ECCRCAN0ERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Store ECC data generated for write data to RAM. 1: Store the value of ECCRCAN0ERDB register to RAM.

Table 22.101 ECCRCAN0TMC Register Contents (2/2)

Bit position	Bit Name	Function
2	ECENS	ECC encoder input selection bit This bit is used to identify the targeted data for ECC data generation as write data to RAM or the value of the ECCRCAN0TED register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate ECC data from the write data to RAM. 1: Generate ECC data from the value of the ECCRCAN0TED register.
1	ECDCS	ECC decoder input selection bit This bit is used to indetify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the ECCRCAN0TED register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate syndrome code from RAM data and detect errors. 1: Generate syndrome code from the value of the ECCRCAN0TED register and detect errors.
0	ECREIS	ECC redundant bit input data selection bit This bit is used to indetify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the ECCRCAN0ERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate syndrome code from ECC data stored in RAM and detect errors. 1: Generate syndrome code from the value of the ECCRCAN0ERDB register and detect errors.

22.11.5 ECCRCAN0TED — RSCAN0 ECC Encode/Decode Input/Output Replacement Test Register

In ECC test mode, this register handles test data.

The value of the register can be used to generate ECC data or syndrome code.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), it is accesible. When ECCRCAN0TMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when RSCAN is not accessed to RAM.

Access: This register can be read/written in 32-bit units.

Address: FFC7 $100C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.102 ECCRCAN0TED Register Contents

Bit position	Bit Name	Function
31 to 0	ECEDB[31:0]	When ECCRCAN0TMC.ECENS = 1, the value of this register is used to generate ECC data and the value this register to RAM is stored to RAM. When ECCRCAN0TMC.ECDCS = 1, the value of this register is used to generate syndrome code and the value of this register is stored in ECC decode syndrome data register (ECCRCAN0SYND). In addition, when ECCRCAN0TMC.ECTRRS = 1, RAM data [31:0], instead of written data, is read for the value of this register.

22.11.6 ECCRCAN0TRC — RSCAN0 ECC Redundant Bit Data Control Test Register

In ECC test mode, this test register, for ECC data, consists of four 8-bit registers, ECCRCAN0SYND, ECCRCAN0HORD, ECCRCAN0ECRD, and ECCRCAN0ERDB.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), this register can be accessed. When ECCRCAN0TMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when RSCAN is not accessed to RAM.

Access: This register is a read-only register that can be read in 32-bit units.

Address: FFC7 1008_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECCRCAN0SYND (See **Section 22.11.7**)								ECCRCAN0HORD (See **Section 22.11.8**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECCRCAN0ECRD (See **Section 22.11.9**)								ECCRCAN0ERDB (See **Section 22.11.10**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

22.11.7 ECCRCAN0SYND — RSCAN0 ECC Decode Syndrome Data Register

In ECC test mode, this is a read-only register for storing generated syndrome code.

Writing to this register is ignored.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), this register can be accessed. When ECC test mode is disabled (ECCRCAN0TMC.ECTMCE = 0), 00_H is read.

Access: This register can be read/written in 8-bit units.

Address: FFC7 $100B_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SYND6	SYND5	SYND4	SYND3	SYND2	SYND1	SYND0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 22.103 ECCRCAN0SYND Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	SYND[6:0]	The generated syndrome code is stored as needed.

22.11.8 ECCRCAN0HORD — RSCAN0 ECC 7-Bit Redundant Bit Data Hold Test Register

In ECC test mode, this register is used to store ECC data for read RAM data.

Writing to this register is ignored.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), this register can be accessed. When ECC test mode is disabled (ECCRCAN0TMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FFC7 $100A_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	HORD6	HORD5	HORD4	HORD3	HORD2	HORD1	HORD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 22.104 ECCRCAN0HORD Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	HORD[6:0]	ECC code for read RAM data is stored as needed. When ECCRCAN0TMC.ECTRRS = 1 and ECCRCAN0TED register is read, ECC code is stored.

22.11.9 ECCRCAN0ECRD — RSCAN0 ECC Encode Test Register

In ECC test mode, this is a read-only register for storing ECC data generated for write RAM data.

Writing to this register is ignored.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (ECCRCAN0TMC.ECTMCE = 0), 00_H is read.

Access: This register is a read-only register that can be read in 8-bit units.

Address: FFC7 1009_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ECRD6	ECRD5	ECRD4	ECRD3	ECRD2	ECRD1	ECRD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 22.105 ECCRCAN0ECRD Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	ECRD[6:0]	These bits can read ECC data generated at the time of RAM data writing, and can read ECC data for the data written to the ECCRCAN0TED register when ECCRCAN0TMC.ECENS = 1.

22.11.10 ECCRCAN0ERDB — RSCAN0 ECC Redundant Bit Input/Output Replacement Buffer Register

In ECC test mode, this register handles ECC data.

The value of this register can be used as ECC data generated at the time of RAM writing or ECC data read at the time of RAM data reading.

When ECC test mode is enabled (ECCRCAN0TMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (ECCRCAN0TMC.ECTMCE = 0), 00_H is read.

Access: This register can be read/written in 8-bit units.

Address: FFC7 1008_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ERDB6	ERDB5	ERDB4	ERDB3	ERDB2	ERDB1	ERDB0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.106 ECCRCAN0ERDB Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6 to 0	ERDB[6:0]	When ECCRCAN0TMC.ECREOS = 1, these bits store the value of this register as ECC data to RAM. When ECCRCAN0TMC.ECREIS = 1, the value of this register is read as ECC data read from RAM. When ECCRCAN0TMC.ECTRRS = 1, ECC data to be stored in RAM is read instead of the write data for the read value of this register.

22.11.11 ECCRCAN0EAD0 — RSCAN0 ECC Error Address Register

ECCRCAN0EAD0 is a read-only register to hold the address at which an ECC error has occurred.

Access: This register can be read/written in 32-bit units.

Address: FFC7 1010$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEAD[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEAD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22.107 ECCRCAN0EAD0 Register Contents

Bit position	Bit Name	Function
31 to 0	ECEAD[31:0]	ECCRCAN0EAD0 is a read-only register to hold the address at which an ECC error has occurred. If an ECC error is detected while ECC error detection is enabled, the RAM address is latched using the detection signal as a trigger, and the address is stored in ECCRCAN0EAD0 as the address at which the ECC error has occurred. The address is stored upon detection of the first ECC error while no error status is set. However, if a 1-bit error is followed by a 2-bit error, the address of the latter is stored. Only one address can be held in ECCRCAN0EAD0.

22.12 Notes on the RSCAN Module

- When changing a global mode, check the GSLPSTS, GHLTSTS, and GRSTSTS flags in the RSCAN0GSTS register for transitions. When changing a channel mode, check the CSLPSTS, CHLTSTS, and CRSTSTS flags in the RSCAN0CmSTS register (m = 0 to 2) for transitions.
- The acceptance filter processing checks receive rules sequentially in ascending order from the minimum rule number. If the same ID, IDE bit, or RTR bit value is set for multiple receive rules, the minimum number of receive rule is used for the acceptance filter processing. If the message does not pass through the subsequent DLC filter processing, the data processing is terminated without returning to the acceptance filter processing and the message is not stored in the buffer.
- When linking transmit buffers to transmit/receive FIFO buffers or allocating transmit buffers to transmit queues, set the control register (RSCAN0TMCp) of the corresponding transmit buffer to 00_H. The status register (RSCAN0TMSTSp) of the corresponding transmit buffer should not be used. Flags in other status registers (registers RSCAN0TMTRSTS0 to RSCAN0TMTRSTS2, RSCAN0TMTARSTS0 to RSCAN0TMTARSTS2, RSCAN0TMTCSTS0 to RSCAN0TMTCSTS2, and RSCAN0TMTASTS0 to RSCAN0TMTASTS2), which correspond to transmit buffers linked to transmit/receive FIFO buffers or allocated to transmit queues remain unchanged. Set the enable bit in the corresponding interrupt enable register (registers RSCAN0TMIEC0 to RSCAN0TMIEC2) to 0 (transmit buffer interrupt is disabled).
- Transmit buffers that are linked to transmit/receive FIFO buffers must not be allocated to transmit queues.
- Only a single transmit/receive FIFO buffer can be linked to a transmit buffer. Do not link two or more transmit/receive FIFO buffers to transmit buffers of the same number.
- When the CANm bit time clock is selected as a timestamp counter clock source, the timestamp counter stops when the corresponding channel has transitioned to channel reset mode or channel halt mode.
- In case of an attempt to store a new received message when the receive FIFO buffer and the transmit/receive FIFO buffer are full, the new message is discarded. If you wish to store a new transmit message in the transmit/receive FIFO buffer or the transmit queue, check that the transmit/receive FIFO buffer or the transmit queue is not full.
- The values of unused receive buffers (RSCAN0RMIDq, RSCAN0RMPTRq, RSCAN0RMDF0q, and RSCAN0RMDF1q registers), receive FIFO buffer access registers (RSCAN0RFIDx, RSCAN0RFPTRx, RSCAN0RFDF0x, and RSCAN0RFDF1x registers), and transmit/receive FIFO buffer access registers (RSCAN0CFIDk, RSCAN0CFPTRk, RSCAN0CFDF0k, and RSCAN0CFDF1k registers) are undefined when the RSCAN module transitions to global operation mode or global test mode after exiting from global reset mode.

Section 23 CANFD Interface (RS-CANFD)

This section contains a generic description of the CANFD Interface (RS-CANFD).

The first part of this section describes all RH850/D1L/D1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of RS-CANFD.

23.1 Features of RH850/D1L/D1M RS-CANFD

23.1.1 Number of Units and Channels

This microcontroller has the following number of RS-CANFD units.

Table 23.1 Number of Units for D1L Devices

	D1L1		D1L2(H)	
Product Name	R7F701401	R7F701421	R7F701402 R7F701403	R7F701422 R7F701423
Number of Units	0	1	0	1
Name	–	RSCANFD0	–	RSCANFD0

Table 23.2 Number of Units for D1M Devices

	D1M1(H)	D1M1-V2		D1M1A		D1M2(H)	
Product Name	R7F701404 R7F701405 R7F701406 R7F701407	R7F701442	R7F701462	R7F701441	R7F701461	R7F701408 R7F701410 R7F701411 R7F701412	R7F701428 R7F701430 R7F701431 R7F701432
Number of Units	0	0	1	0	1	0	1
Name	–	–	RSCANFD0	–	RSCANFD0	–	RSCANFD0

The individual products have the RS-CANFD Interface Channel listed below.

Table 23.3 Unit Configurations and Channels for D1L Devices

	D1L1		D1L2(H)	
Channel Name	R7F701401	R7F701421	R7F701402 R7F701403	R7F701422 R7F701423
Unit Name: RSCANFD0				
CAN0	–	√	–	√
CAN1	–	√	–	√
CAN2	–	√	–	√

Table 23.4 Unit Configurations and Channels for D1M Devices

	D1M1(H)	D1M1-V2		D1M1A		D1M2(H)	
Channel Name	R7F701404 R7F701405 R7F701406 R7F701407	R7F701442	R7F701462	R7F701441	R7F701461	R7F701408 R7F701410 R7F701411 R7F701412	R7F701428 R7F701430 R7F701431 R7F701432
Unit Name: RSCANFD0							

Table 23.4 Unit Configurations and Channels for D1M Devices

	D1M1(H)	D1M1-V2		D1M1A		D1M2(H)	
Channel Name	R7F701404 R7F701405 R7F701406 R7F701407	R7F701442	R7F701462	R7F701441	R7F701461	R7F701408 R7F701410 R7F701411 R7F701412	R7F701428 R7F701430 R7F701431 R7F701432
CAN0	–	–	√	–	√	–	√
CAN1	–	–	√	–	√	–	√
CAN2	–	–	√	–	√	–	√

The RS-CANFD has two interface modes (classical CAN mode and CAN FD mode) and uses different registers for each mode. There are two types of register names RSCANnXXX and RSCFDnCFDXXX (XXX: arbitrary) depending on interface modes. When explaining specifications common to two registers, register names are described as RSCFDn(CFD)XXX.

Table 23.5 Index

Index	Meaning
n	Throughout this section, the individual RS-CANFD units are generically indicated by the index “n” (n = 0); for example, RSCFDn(CFD)GCTR is the global control register of the RSCFDn unit.
m	Throughout this section, the individual channels of RS-CANFD units are generically indicated by the index “m” (m = 0 to 2); for example, RSCFDn(CFD)CmSTS is the channel m status register.
j	The individual registers associated with receive rule table are generically indicated by the index “j” (j = 0 to 15); for example, RSCFDn(CFD)GAFLIDj is the receive rule ID register.
k	The individual transmit/receive FIFO buffers are generically indicated by the index “k” (k = 0 to [channel m × 3 + 2]); for example, RSCFDn(CFD)CFCCk is the transmit/receive FIFO buffer configuration/control register.
x	The individual receive FIFO buffers are generically identified by the index “x” (x = 0 to 7); for example, RSCFDn(CFD)RFSTSx is the receive FIFO buffer status register.
d	Data field registers of transmit/receive FIFO buffers and receive FIFO buffers are identified by “d” (classical CAN mode: d = 0 to 1, CAN FD mode: d = 0 to 15). For example, the transmit/receive FIFO buffer data field register is described as RSCFDn(CFD)CFDFd_k.
q	The individual receive buffers are generically indicated by the index “q” (q = 0 to [channel m × 16 + 15]); for example, RSCFDn(CFD)RMIDq is the receive buffer ID register.
p	The individual transmit buffers are generically indicated by the index “p” (p = 0 to [channel m × 16 + 15]; for example, RSCFDn(CFD)TMCp is the transmit buffer control register.
b	Data field registers of receive buffers and transmit buffers are identified by “b” (classical CAN mode: b = 0 to 1, CAN FD mode: b = 0 to 4). For example, the receive buffer data field register is described as RSCFDn(CFD)RMDFb_q.
r	The individual RAM tests for CAN are generically indicated by the index “r” (r = 0 to 63); for example, RSCFDn(CFD)RPGACCr is the RAM test page access register.
y	The registers not covered above are indicated by the index “y” (y = 0, 1); for example, RSCFDn(CFD)RMNDy is a receive buffer new data register.

23.1.2 Register Base Address

RSCFDn base addresses are listed in the following table.

RSCFDn register addresses are given as offsets from the base addresses in general.

Table 23.6 Register Base Address

Base Address Name	Base Address
<RSCFD0_base>	FFD0 0000_H

23.1.3 Clock Supply

The RSCFDn clock supply is shown in the following table.

Table 23.7 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RSCFDn	clk_xincan	Clock Controller C_ISO_RSCANXIN
	clkc	Clock Controller C_ISO_RSCAN
	pclk	Clock Controller RSCANPCLK

The operating frequency of the RSCFDn depends on the transfer rate and the number of channels in use. **Table 23.8** shows the range of the frequency.

Table 23.8 Setting Range example of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/D1L/D1M

Condition			Range of Operating Frequency	
Nominal Bit Rate	Data Bit Rate	No. of Channels in Use	pclk	clkc, clk_xincan*1
1 Mbps	8 Mbps (5Tq setting)	3ch	pclk ≥ 80 MHz	40 MHz ≤ clk_xincan ≤ pclk/2
		2ch	pclk ≥ 80 MHz	
		1ch	pclk ≥ 80 MHz	
500 Kbps	2 Mbps (8Tq setting)	3ch	pclk ≥ 32 MHz	16 MHz ≤ clk_xincan ≤ pclk/2
		2ch	pclk ≥ 32 MHz	
		1ch	pclk ≥ 32 MHz	
1 Mbps	5 Mbps (8Tq setting)	3ch	pclk ≥ 80 MHz	40 MHz ≤ clk_xincan ≤ pclk/2
		2ch	pclk ≥ 80 MHz	
		1ch	pclk ≥ 80 MHz	

Note 1. Setting the DCS bit in the RSCFDn(CFD)GCFG register enables to select either clk_xincan or clkc. Set clocks less than or equal to pclk/2.

23.1.4 Interrupt Request

RSCFDn interrupt requests are listed in the following table.

Table 23.9 Interrupt Requests (1/2)

Unit Interrupt Signal	Outline	Connected to
RSCFD0		
INTRCANGERR	CAN global error interrupt	Interrupt Controller INTRCANGERR
INTRCANGRECC	CAN receive FIFO interrupt	Interrupt Controller INTRCANGRECC
CAN0		
INTRCANmERR (m = 0)	CAN0 error interrupt	Interrupt Controller INTRCAN0ERR
INTRCANmREC (m = 0)	CAN0 transmit/receive FIFO receive completion interrupt	Interrupt Controller INTRCAN0REC
INTRCANmTRX (m = 0)	CAN0 transmit interrupt	Interrupt Controller INTRCAN0TRX
CAN1		
INTRCANmERR (m = 1)	CAN1 error interrupt	Interrupt Controller INTRCAN1ERR
INTRCANmREC (m = 1)	CAN1 transmit/receive FIFO receive completion interrupt	Interrupt Controller INTRCAN1REC
INTRCANmTRX (m = 1)	CAN1 transmit interrupt	Interrupt Controller INTRCAN1TRX
CAN2		

Table 23.9 Interrupt Requests (2/2)

Unit Interrupt Signal	Outline	Connected to
INTRCANmERR (m = 2)	CAN2 error interrupt	Interrupt Controller INTRCAN2ERR
INTRCANmREC (m = 2)	CAN2 transmit/receive FIFO receive completion interrupt	Interrupt Controller INTRCAN2REC
INTRCANmTRX (m = 2)	CAN2 transmit interrupt	Interrupt Controller INTRCAN2TRX

NOTE

For the wake-up factors from standby mode, see Section 13.1.2.1, Wake-up factors.

23.1.5 Reset Sources

RSCFDn reset sources are listed in the following table. RSCFDn is initialized by these reset sources.

Table 23.10 Reset Sources

Unit Name	Reset Source
RSCFDn	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

23.1.6 External Input/Output Signals

External input/output signals of RSCFDn are listed below.

Table 23.11 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
RSCFD0		
CANmRX (m = 0 to 2)	CANm receive data input	CANmRX (m = 0 to 2)
CANmTX (m = 0 to 2)	CANm transmit data output	CANmTX (m = 0 to 2)
CANmDREN (m = 0 to 2)	CANm data phase output	CANmDREN (m = 0 to 2)

23.2 Overview

23.2.1 Functional Overview

Table 23.12 shows the RS-CANFD module specifications. **Figure 23.1** shows the RS-CANFD module block diagram.

Table 23.12 RS-CANFD module Specifications (1/3)

Item	Specification
Number of channels	3
Protocol	ISO11898-1 compliant Using CAN FD frames is selectable by switching interface modes.
Communication speed	Classical CAN mode: • Maximum 1 Mbps $\text{Communication speed (CANm bit time clock)} = \frac{1}{\text{CANm bit time}}$ CANm bit time = CANmTq × Tq count per bit $\text{CANmTq} = \frac{\text{(BRP[9:0] bits in the RSCANnCmCFG register + 1)}}{\text{fCAN}}$ fCAN: Frequency of CAN clock (selected by the DCS bit in the RSCANnGCFG register) CAN FD mode: • Nominal bit rate: max.1 Mbps, data bit rate: max. 5 Mbps $\text{Transmission rate (CANm nominal bit time clock)} = \frac{1}{\text{CANm nominal bit time}}$ $\text{Transmission rate (CANm data bit time clock)} = \frac{1}{\text{CANm data bit time}}$ CANm nominal bit time = CANmTq(N) × Tq count per nominal bit CANm data bit time = CANmTq(D) × Tq count per data bit $\text{CANmTq(N)} = \frac{\text{(NBRP[9:0] bits in the RSCFDnCFDCmNCFG register + 1)}}{\text{fCAN}}$ $\text{CANmTq(D)} = \frac{\text{(DBRP[7:0] bits in the RSCFDnCFDCmDCFG register + 1)}}{\text{fCAN}}$ fCAN: Frequency of CAN clock (selected by the DCS bit in the RSCFDnCFDGCFG register) m = 0 to 2 Tq: Time quantum **Note:** To check calculation of sample point, see Section 23.11.1.2, Bit Timing Setting.
Buffer	240 buffers in total • Individual buffers: 48 buffers (16 buffers × 3 channels) Transmit buffer: 16 buffers per channel Transmit queue: Single queue per channel (shared with the transmit buffer; up to 16 buffers allocatable) • Shared buffers: 192 buffers for all channels Receive buffer: 48 buffers Receive FIFO buffer: 8 FIFO buffers (up to 128 buffers allocatable to each) Transmit/receive FIFO buffer: 3 FIFO buffers per channel (up to 128 buffers allocatable to each) • ECC included The 7-bit ECC value is calculated for each 32-bit transfer.

Table 23.12 RS-CANFD module Specifications (2/3)

Item	Specification
Reception function	• Receives data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be received. • Sets interrupt enable/disable for each FIFO. • Mirror function (reception of messages transmitted from the own CAN node) • Timestamp function (to record message reception time as a 16-bit timer value)
Reception filter function	• Selects receive messages according to 192 receive rules. • Sets the number of receive rules (0 to 127) for each channel. • Acceptance filter processing: Sets ID and mask for each receive rule. • DLC filter processing: Enables DLC filter check for each acceptance rule.
Receive message transfer function	• Routing function Transfers receive messages to arbitrary destinations (can be transferred to up to 8 buffers) Transfer destination: Receive buffer, receive FIFO buffer, and/or transmit/receive FIFO buffer • Label addition function Stores label information together with a message in a receive buffer and FIFO buffer.
Transmission function	• Transmits data frames and remote frames. • Selects ID format (standard ID, extended ID, or both IDs) to be transmitted. • Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer. • Selects ID priority transmission or transmit buffer number priority transmission. • Transmit request can be aborted (possible to confirm with a flag) • One-shot transmission function
Interval transmission function	Transmit messages at configurable intervals (transmit mode or gateway mode of transmit/receive FIFO buffers)
Transmit queue function	Transmits all stored messages according to the ID priority.
Transmit history function	Stores the history information of transmission-completed messages Adds timestamp (recording message transmission time as a 16-bit timer value) to the history information.
Gateway function	Transmits a received message automatically.
Bus off recovery mode selection	Selects the method for returning from bus off state. • ISO11898-1 compliant • Automatic entry to channel halt mode at bus-off entry • Automatic entry to channel halt mode at bus-off end • Transition to channel halt mode by program request • Transition to the error-active state by program request (forcible return from the bus off state)
Error status monitoring	• Monitors CAN protocol errors (stuff error, form error, ACK error, CRC error, bit error, ACK delimiter error, and bus dominant lock). • Detects error status transitions (error warning, error passive, bus off entry, and bus off recovery) • Reads the error counter. • Monitors DLC errors.
Interrupt source	11 sources • Global Interrupts (2 sources) Receive FIFO interrupt Global error interrupt • Channel interrupts (3 sources/channel) CANm transmit interrupt (m = 0 to 2) – CANm transmit complete interrupt – CANm transmit abort interrupt – CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode) – CANm transmit history interrupt – CANm transmit queue interrupt CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode) CANm error interrupt
CAN stop mode	Reduces power consumption by stopping clock supply to the RS-CANFD module.

Table 23.12 RS-CANFD module Specifications (3/3)

Item	Specification
CAN clock source	Selects the clkc or the clk_xincan. As for the range of operating frequency, see **Table 23.8**.
Test function	Test function for user evaluation • Listen-only mode • Self-test mode 0 (external loopback) • Self-test mode 1 (internal loopback) • Restricted operation mode • RAM test (read/write test) • Inter-channel communication test [CRC error test enabled]

23.2.2 Interface Modes

The RS-CANFD has two interface modes.

- Classical CAN mode: Handles only classical CAN frames.
- CAN FD mode: Handles classical CAN frames and CAN FD frames.

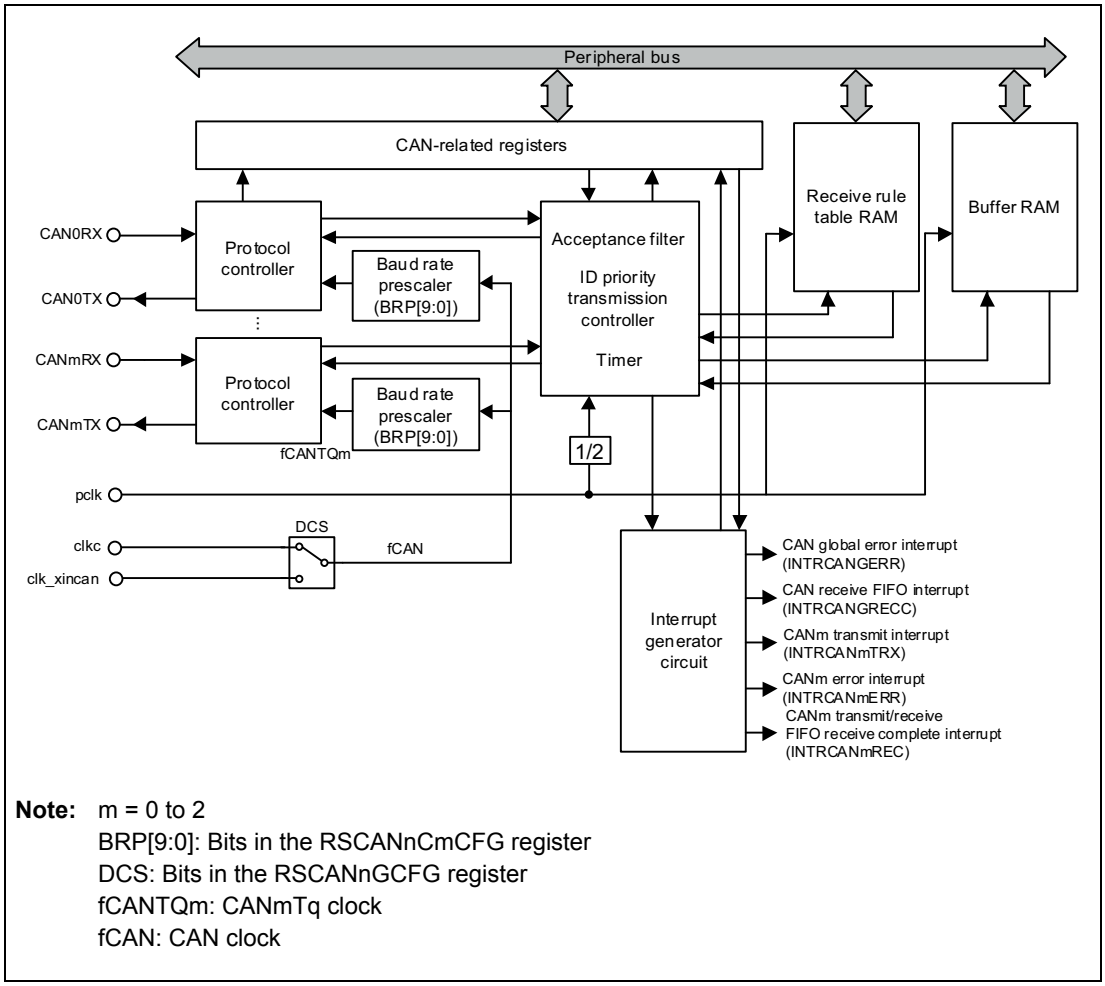
These two modes use different register maps with the same base address. Register maps change by switching interface modes.

Interface modes are switched by the RCMC bit in the RSCFDn(CFD)GRMCFG register.

23.2.3 CAN FD protocol switchover

This product supports CAN FD according to the ISO 11898-1:2015 protocol that specifies the new CRC field including stuff counters. In addition, by setting the NIE bit in the RSCFDnCFDGCRCCFG register, this product can support the CRC field that has compatibility to CAN FD protocol (ISO/CD 11898-1 (2014-08-12 version)).

23.2.4 Block Diagram



Note: m = 0 to 2
BRP[9:0]: Bits in the RSCANnCmCFG register
DCS: Bits in the RSCANnGCFG register
fCANTQm: CANmTq clock
fCAN: CAN clock

Figure 23.1 RS-CANFD module Block Diagram (Classical CAN Mode)

In CAN FD mode, different clock signals are input to the baud rate prescaler and the protocol controller respectively. See Section 23.11.1.3, Communication Speed Setting.

23.3 Registers (Classical CAN Mode)

23.3.1 List of Registers

The following tables list RS-CANFD registers to be used in classical CAN mode.

For details about <RSCFDn_base>, see Section 23.1.2, Register Base Address.

For details about registers initialized in Global reset mode or Channel reset mode, see following:

- Table 23.177, Registers Initialized in Global Reset Mode or Channel Reset Mode
- Table 23.178, Registers Initialized Only in Global Reset Mode

Table 23.13 Registers (1/2)

Module	Register	Symbol	Address
Interface mode-related registers			
RSCANn	Global interface mode select register	RSCANnGRMCFG	<RSCFDn_base> + $04FC_H$
Channel-related registers			
RSCANn	Channel m Configuration Register	RSCANnCmCFG	<RSCFDn_base> + 0000_H + (10_H × m)
RSCANn	Channel m control register	RSCANnCmCTR	<RSCFDn_base> + 0004_H + (10_H × m)
RSCANn	Channel m status register	RSCANnCmSTS	<RSCFDn_base> + 0008_H + (10_H × m)
RSCANn	Channel m error flag register	RSCANnCmERFL	<RSCFDn_base> + $000C_H$ + (10_H × m)
Global-related registers			
RSCANn	Global configuration register	RSCANnGCFG	<RSCFDn_base> + 0084_H
RSCANn	Global control register	RSCANnGCTR	<RSCFDn_base> + 0088_H
RSCANn	Global status register	RSCANnGSTS	<RSCFDn_base> + $008C_H$
RSCANn	Global error flag register	RSCANnGERFL	<RSCFDn_base> + 0090_H
RSCANn	Global timestamp counter register	RSCANnGTSC	<RSCFDn_base> + 0094_H
RSCANn	Global TX Interrupt Status Register 0	RSCANnGTINTSTS0	<RSCFDn_base> + 0460_H
Receive rule-related registers			
RSCANn	Receive Rule Entry Control Register	RSCANnGAFLECTR	<RSCFDn_base> + 0098_H
RSCANn	Receive Rule Configuration Register 0	RSCANnGAFLCFG0	<RSCFDn_base> + $009C_H$
RSCANn	Receive Rule ID Register j	RSCANnGAFLIDj	<RSCFDn_base> + 0500_H + (10_H × j)
RSCANn	Receive Rule Mask Register j	RSCANnGAFLMj	<RSCFDn_base> + 0504_H + (10_H × j)
RSCANn	Receive Rule Pointer 0 Register j	RSCANnGAFLP0_j	<RSCFDn_base> + 0508_H + (10_H × j)
RSCANn	Receive Rule Pointer 1 Register j	RSCANnGAFLP1_j	<RSCFDn_base> + $050C_H$ + (10_H × j)
Receive buffer-related registers			
RSCANn	Receive Buffer Number Register	RSCANnRMNB	<RSCFDn_base> + $00A4_H$
RSCANn	Receive Buffer New Data Register y	RSCANnRMNDy	<RSCFDn_base> + $00A8_H$ + (04_H × y)
RSCANn	Receive Buffer ID Register q	RSCANnRMIDq	<RSCFDn_base> + 0600_H + (10_H × q)
RSCANn	Receive Buffer Pointer Register q	RSCANnRMPTRq	<RSCFDn_base> + 0604_H + (10_H × q)
RSCANn	Receive Buffer Data Field 0 Register q	RSCANnRMDF0_q	<RSCFDn_base> + 0608_H + (10_H × q)
RSCANn	Receive Buffer Data Field 1 Register q	RSCANnRMDF1_q	<RSCFDn_base> + $060C_H$ + (10_H × q)
Receive FIFO buffer-related registers			
RSCANn	Receive FIFO Buffer Configuration and Control Register x	RSCANnRFCCx	<RSCFDn_base> + $00B8_H$ + (04_H × x)
RSCANn	Receive FIFO Buffer Status Register x	RSCANnRFSTSx	<RSCFDn_base> + $00D8_H$ + (04_H × x)
RSCANn	Receive FIFO Buffer Pointer Control Register x	RSCANnRFPCTRx	<RSCFDn_base> + $00F8_H$ + (04_H × x)
RSCANn	Receive FIFO Buffer Access ID Register x	RSCANnRFIDx	<RSCFDn_base> + $0E00_H$ + (10_H × x)
RSCANn	Receive FIFO Buffer Access Pointer Register x	RSCANnRFPTRx	<RSCFDn_base> + $0E04_H$ + (10_H × x)
RSCANn	Receive FIFO Buffer Access Data Field 0 Register x	RSCANnRFDF0_x	<RSCFDn_base> + $0E08_H$ + (10_H × x)
RSCANn	Receive FIFO Buffer Access Data Field 1 Register x	RSCANnRFDF1_x	<RSCFDn_base> + $0E0C_H$ + (10_H × x)

Table 23.13 Registers (2/2)

Module	Register	Symbol	Address
Transmit/Receive FIFO buffer related registers			
RSCANn	Transmit/receive FIFO Buffer Configuration and Control Register k	RSCANnCFCCk	<RSCFDn_base> + 0118_H + (04_H × k)
RSCANn	Transmit/receive FIFO Buffer Status Register k	RSCANnCFSTSk	<RSCFDn_base> + 0178_H + (04_H × k)
RSCANn	Transmit/receive FIFO Buffer Pointer Control Register k	RSCANnCFPCTRk	<RSCFDn_base> + $01D8_H$ + (04_H × k)
RSCANn	Transmit/receive FIFO Buffer Access ID Register k	RSCANnCFIDk	<RSCFDn_base> + $0E80_H$ + (10_H × k)
RSCANn	Transmit/receive FIFO Buffer Access Pointer Register k	RSCANnCFPTRk	<RSCFDn_base> + $0E84_H$ + (10_H × k)
RSCANn	Transmit/receive FIFO Buffer Access Data Field 0 Register k	RSCANnCFDF0_k	<RSCFDn_base> + $0E88_H$ + (10_H × k)
RSCANn	Transmit/receive FIFO Buffer Access Data Field 1 Register k	RSCANnCFDF1_k	<RSCFDn_base> + $0E8C_H$ + (10_H × k)
FIFO status-related registers			
RSCANn	FIFO Empty Status Register	RSCANnFESTS	<RSCFDn_base> + 0238_H
RSCANn	FIFO Full Status Register	RSCANnFFSTS	<RSCFDn_base> + $023C_H$
RSCANn	FIFO Message Lost Status Register	RSCANnFMSTS	<RSCFDn_base> + 0240_H
RSCANn	Receive FIFO Buffer Interrupt Flag Status Register	RSCANnRFISTS	<RSCFDn_base> + 0244_H
RSCANn	Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register	RSCANnCFRISTS	<RSCFDn_base> + 0248_H
RSCANn	Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register	RSCANnCFTISTS	<RSCFDn_base> + $024C_H$
Transmit buffer-related registers			
RSCANn	Transmit Buffer Control Register p	RSCANnTMCp	<RSCFDn_base> + 0250_H + (01_H × p)
RSCANn	Transmit Buffer Status Register p	RSCANnTMSTSp	<RSCFDn_base> + $02D0_H$ + (01_H × p)
RSCANn	Transmit Buffer ID Register p	RSCANnTMIDp	<RSCFDn_base> + 1000_H + (10_H × p)
RSCANn	Transmit Buffer Pointer Register p	RSCANnTMPTRp	<RSCFDn_base> + 1004_H + (10_H × p)
RSCANn	Transmit Buffer Data Field 0 Register p	RSCANnTMDF0_p	<RSCFDn_base> + 1008_H + (10_H × p)
RSCANn	Transmit Buffer Data Field 1 Register p	RSCANnTMDF1_p	<RSCFDn_base> + $100C_H$ + (10_H × p)
RSCANn	Transmit Buffer Interrupt Enable Configuration Register y	RSCANnTMIECy	<RSCFDn_base> + 0390_H + (04_H × y)
Transmit buffer status-related registers			
RSCANn	Transmit Buffer Transmit Request Status Register y	RSCANnTMTRSTSy	<RSCFDn_base> + 0350_H + (04_H × y)
RSCANn	Transmit Buffer Transmit Abort Request Status Register y	RSCANnTMTARSTSy	<RSCFDn_base> + 0360_H + (04_H × y)
RSCANn	Transmit Buffer Transmit Complete Status Register y	RSCANnTMTCSTSy	<RSCFDn_base> + 0370_H + (04_H × y)
RSCANn	Transmit Buffer Transmit Abort Status Register y	RSCANnTMTASTSy	<RSCFDn_base> + 0380_H + (04_H × y)
Transmit queue-related registers			
RSCANn	Transmit Queue Configuration and Control Register m	RSCANnTXQCCm	<RSCFDn_base> + $03A0_H$ + (04_H × m)
RSCANn	Transmit Queue Status Register m	RSCANnTXQSTSm	<RSCFDn_base> + $03C0_H$ + (04_H × m)
RSCANn	Transmit Queue Pointer Control Register m	RSCANnTXQPCTRm	<RSCFDn_base> + $03E0_H$ + (04_H × m)
Transmit history-related registers			
RSCANn	Transmit History Configuration and Control Register m	RSCANnTHLCCm	<RSCFDn_base> + 0400_H + (04_H × m)
RSCANn	Transmit History Status Register m	RSCANnTHLSTSm	<RSCFDn_base> + 0420_H + (04_H × m)
RSCANn	Transmit History Access Register m	RSCANnTHLPCTRm	<RSCFDn_base> + 0440_H + (04_H × m)
RSCANn	Transmit History Pointer Control Register m	RSCANnTHLACCm	<RSCFDn_base> + 1800_H + (04_H × m)
Test-related registers			
RSCANn	Global Test Configuration Register	RSCANnGTSTCFG	<RSCFDn_base> + 0468_H
RSCANn	Global Test Control Register	RSCANnGTSTCTR	<RSCFDn_base> + $046C_H$
RSCANn	Global Lock Key Register	RSCANnGLOCKK	<RSCFDn_base> + $047C_H$
RSCANn	RAM Test Page Access Register r	RSCANnRPGACCr	<RSCFDn_base> + 1900_H + (04_H × r)

Table 23.14 Transmit Buffer p Allocated to Each Channel

	CANm
Transmit buffer p	Transmit buffer 16 × m + 0
	Transmit buffer 16 × m + 1
	Transmit buffer 16 × m + 2
	Transmit buffer 16 × m + 3
	Transmit buffer 16 × m + 4
	Transmit buffer 16 × m + 5
	Transmit buffer 16 × m + 6
	Transmit buffer 16 × m + 7
	Transmit buffer 16 × m + 8
	Transmit buffer 16 × m + 9
	Transmit buffer 16 × m + 10
	Transmit buffer 16 × m + 11
	Transmit buffer 16 × m + 12
	Transmit buffer 16 × m + 13
	Transmit buffer 16 × m + 14
	Transmit buffer 16 × m + 15

Table 23.15 Transmit/Receive FIFO Buffer k Allocated to Each Channel

	CANm
Transmit/receive FIFO buffer k	Transmit/receive FIFO buffer 3 × m + 0
	Transmit/receive FIFO buffer 3 × m + 1
	Transmit/receive FIFO buffer 3 × m + 2

Table 23.16 Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer by the Setting of Bits CFTML[3:0]

Setting of Bits CFTML[3:0]	Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer
0000_B	Transmit buffer 16 × m + 0
0001_B	Transmit buffer 16 × m + 1
0010_B	Transmit buffer 16 × m + 2
0011_B	Transmit buffer 16 × m + 3
0100_B	Transmit buffer 16 × m + 4
0101_B	Transmit buffer 16 × m + 5
0110_B	Transmit buffer 16 × m + 6
0111_B	Transmit buffer 16 × m + 7
1000_B	Transmit buffer 16 × m + 8
1001_B	Transmit buffer 16 × m + 9
1010_B	Transmit buffer 16 × m + 10
1011_B	Transmit buffer 16 × m + 11
1100_B	Transmit buffer 16 × m + 12
1101_B	Transmit buffer 16 × m + 13
1110_B	Transmit buffer 16 × m + 14
1111_B	Transmit buffer 16 × m + 15

Table 23.17 Transmit Buffer p Allocated to the Transmit Queue of Each Channel

Setting of Bits TXQDC[3:0]	Transmit Buffer p Allocated to the Transmit Queue
0000_B	Setting prohibited
0001_B	Setting prohibited
0010_B	Transmit buffer 16 × m + 15 to 16 × m + 13
0011_B	Transmit buffer 16 × m + 15 to 16 × m + 12
0100_B	Transmit buffer 16 × m + 15 to 16 × m + 11
0101_B	Transmit buffer 16 × m + 15 to 16 × m + 10
0110_B	Transmit buffer 16 × m + 15 to 16 × m + 9
0111_B	Transmit buffer 16 × m + 15 to 16 × m + 8
1000_B	Transmit buffer 16 × m + 15 to 16 × m + 7
1001_B	Transmit buffer 16 × m + 15 to 16 × m + 6
1010_B	Transmit buffer 16 × m + 15 to 16 × m + 5
1011_B	Transmit buffer 16 × m + 15 to 16 × m + 4
1100_B	Transmit buffer 16 × m + 15 to 16 × m + 3
1101_B	Transmit buffer 16 × m + 15 to 16 × m + 2
1110_B	Transmit buffer 16 × m + 15 to 16 × m + 1
1111_B	Transmit buffer 16 × m + 15 to 16 × m + 0

23.3.2 Details of Channel-Related Registers

23.3.2.1 RSCANnGRMCFG — Global Interface Mode Select Register

Access: RSCANnGRMCFG register can be read/written in 32-bit units
RSCANnGRMCFGL register can be read/written in 16-bit units
RSCANnGRMCFGLL register can be read/written in 8-bit units

Address: RSCANnGRMCFG: <RSCFDn_base> + $04FC_H$

RSCANnGRMCFGL: <RSCFDn_base> + $04FC_H$

RSCANnGRMCFGLL: <RSCFDn_base> + $04FC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCMC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 23.18 RSCANnGRMCFG Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
0	RCMC	Interface Mode Select 0: Classical CAN mode 1: CAN FD mode

Note 1. The RSCANnGRMCFG register and the RSCFDnCFDGRMCFG register are the same register. Therefore, set either one of the registers.

Modify the RSCANnGRMCFG register only in global reset mode. Before setting other RS-CANFD registers, set this register.

RCMC Bit

Setting this bit to 0 makes classical CAN mode available. To switch CAN FD mode to classical CAN mode, set the value after reset to all registers and bits allocated to the register map of CAN FD mode and then modify the RSCANnGRMCFG register.

To switch the RS-CAN FD module from CAN FD mode to classical CAN mode, set the values after reset to all respective registers and bits allocated only to the register map in CAN FD mode, and then modify the value of RSCANnGRMCFG register.

23.3.2.2 RSCANnCmCFG — Channel Configuration Register (m = 0 to 2)

Access: RSCANnCmCFG register can be read/written in 32-bit units
RSCANnCmCFGL, RSCANnCmCFGH registers can be read/written in 16-bit units
RSCANnCmCFGLL, RSCANnCmCFGLH, RSCANnCmCFGHL, RSCANnCmCFGHH registers can be read/written in 8-bit units

Address: RSCANnCmCFG: <RSCFDn_base> + 0000_H + (10_H × m)

RSCANnCmCFGL: <RSCFDn_base> + 0000_H + (10_H × m),
RSCANnCmCFGH: <RSCFDn_base> + 0002_H + (10_H × m)

RSCANnCmCFGLL: <RSCFDn_base> + 0000_H + (10_H × m),
RSCANnCmCFGLH: <RSCFDn_base> + 0001_H + (10_H × m),
RSCANnCmCFGHL: <RSCFDn_base> + 0002_H + (10_H × m),
RSCANnCmCFGHH: <RSCFDn_base> + 0003_H + (10_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	SJW[1:0]		—	TSEG2[2:0]			TSEG1[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	BRP[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.19 RSCANnCmCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 26	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
25, 24	SJW[1:0]	Resynchronization Jump Width Control b25 b24 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq
23	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
22 to 20	TSEG2[2:0]	Time Segment 2 Control b22 b21 b20 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq

Table 23.19 RSCANnCmCFG Register Contents (2/2)

Bit Position	Bit Name	Function
19 to 16	TSEG1[3:0]	Time Segment 1 Control b19 b18 b17 b16 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq
15 to 10	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
9 to 0	BRP[9:0]	Prescaler Division Ratio Set When these bits are set to P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

Modify the RSCANnCmCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode before shifting to channel communication mode or channel halt mode. For a description of the bit timing parameters and settings, see Section 23.11.1, Initial Settings.

SJW[1:0] Bits

These bits are used to specify a Tq value for the resynchronization jump width. Allowed values are 1 Tq to 4 Tq, inclusive.
Set a value less than or equal to the value of the TSEG2[2:0] bits.

TSEG2[2:0] Bits

These bits are used to specify a Tq value for the length of phase segment 2 (PHASE_SEG2).
Allowed values are 2 Tq to 8 Tq, inclusive.
Set a value smaller than the value of the TSEG1[3:0] bits.

TSEG1[3:0] Bits

These bits are used to specify a Tq value for the total length of the propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1).
Allowed values are 4 Tq to 16 Tq, inclusive.

BRP[9:0] Bits

The CANmTq clock (fCANTQm) is calculated by dividing the CAN clock (fCAN) by the baud rate prescaler, ((BRP[9:0]) + 1). One clock cycle of the CANmTq clock is 1 Time Quantum (Tq).

23.3.2.3 RSCANnCmCTR — Channel Control Register (m = 0 to 2)

Access: RSCANnCmCTR register can be read/written in 32-bit units
RSCANnCmCTRL, RSCANnCmCTRH registers can be read/written in 16-bit units
RSCANnCmCTRLL, RSCANnCmCTRLH, RSCANnCmCTRHL, RSCANnCmCTRHH registers can be read/written in 8-bit units

Address: RSCANnCmCTR: <RSCFDn_base> + 0004_H + (10_H × m)

RSCANnCmCTRL: <RSCFDn_base> + 0004_H + (10_H × m),
RSCANnCmCTRH: <RSCFDn_base> + 0006_H + (10_H × m)

RSCANnCmCTRLL: <RSCFDn_base> + 0004_H + (10_H × m),
RSCANnCmCTRLH: <RSCFDn_base> + 0005_H + (10_H × m),
RSCANnCmCTRHL: <RSCFDn_base> + 0006_H + (10_H × m),
RSCANnCmCTRHH: <RSCFDn_base> + 0007_H + (10_H × m)

Value after reset: 0000 0005_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CRCT	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	—	—	—	TAIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 23.20 RSCANnCmCTR Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
30	CRCT	CRC Error Test Enable 0: The first bit of the reception ID field is not inverted. 1: The first bit of the reception ID field is inverted.
29 to 27	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
26, 25	CTMS[1:0]	Communication Test Mode Select b26 b25 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback mode) 1 1: Self-test mode 1 (internal loopback mode)
24	CTME	Communication Test Mode Enable 0: Communication test mode is disabled. 1: Communication test mode is enabled.
23	ERRD	Error Display Mode Select 0: Error flags are displayed only for the first error information after bits 14 to 8 in the RSCANnCmERFL register are all cleared. 1: Error flags for all error information are displayed.
22, 21	BOM[1:0]	Bus Off Recovery Mode Select b22 b21 0 0: ISO11898-1 compliant 0 1: Entry to channel halt mode automatically at bus-off entry 1 0: Entry to channel halt mode automatically at bus-off end 1 1: Entry to channel halt mode (in bus-off state) by program request
20 to 17	Reserved	These bits are read as the value after reset. The write value should be the value after reset.

Table 23.20 RSCANnCmCTR Register Contents (2/2)

Bit Position	Bit Name	Function
16	TAIE	Transmit Abort Interrupt Enable 0: Transmit abort interrupt is disabled. 1: Transmit abort interrupt is enabled.
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.
13	OLIE	Overload Frame Transmit Interrupt Enable 0: Overload frame transmit interrupt is disabled. 1: Overload frame transmit interrupt is enabled.
12	BORIE	Bus Off Recovery Interrupt Enable 0: Bus off recovery interrupt is disabled. 1: Bus off recovery interrupt is enabled.
11	BOEIE	Bus Off Entry Interrupt Enable 0: Bus off entry interrupt is disabled. 1: Bus off entry interrupt is enabled.
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.
7 to 4	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
3	RTBO	Forcible Return from Bus-off When this bit is set to 1, forcible return from the bus off state is made. This bit is always read as 0.
2	CSLPR	Channel Stop Mode 0: Other than channel stop mode 1: Channel stop mode
1, 0	CHMDC[1:0]	Mode Select b1 b0 0 0: Channel communication mode 0 1: Channel reset mode 1 0: Channel halt mode 1 1: Setting prohibited

CRCT Bit

This bit is used to test the CRC generation circuit in the RS-CANFD module. Setting this bit to 1 inverts the first bit of the ID field when a message is received. With this inversion of bit, the CRC calculation result does not match the normal CRC value of the received frame, which can detect a CRC error (the CERR bit in the RSCANnCmERFL register is 1). When using this function, note the following.

- This function is available while the CTME bit in the RSCANnCmCTR register is 1 (communication test mode enabled).
- This function cannot communicate with other CAN nodes. Use this function for inter-channel communication test (the CmICBCE bit in the RSCANnGTSTCFG register is 1).
- Bit inversion in the ID field may cause bit stuffing rule violation. In that case, no CRC error is

detected but a stuff error is detected.

Modify this bit only in channel halt mode. This bit is always 0 in channel reset mode.

CTMS[1:0] Bits

These bits are used to select a communication test mode. Modify these bits in channel halt mode only. These bits are set to 0 in channel reset mode.

CTME Bit

Setting this bit to 1 enables communication test mode. Modify these bits in channel halt mode. This bit is set to 0 in channel reset mode.

ERRD Bit

This bit is used to control the display mode of bits 14 to 8 in the RSCANnCmERFL register.
When this bit is clear to 0, if any error is detected while the flags of bits 14-8 in the RSCANnCmERFL register are all 0, only the flags of the first error event are set to 1. If two or more errors occur in the first error event, all the flags of the detected errors are set to 1.

When this bit is set to 1, all the flags of errors that have occurred are set to 1 regardless of the error occurrence order.

Modify this bit only in channel reset mode or channel halt mode.

BOM[1:0] Bits

These bits are used to select the bus off recovery mode of the RS-CANFD module.

When the BOM[1:0] bits are set to 00_B, return from the bus off state to the error active state is compliant with the CAN specifications. That is, the RS-CANFD module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CHMDC[1:0] bits are set to 10_B (channel halt mode) before recessive bits are detected 128 times, the RS-CANFD module does not transition to channel halt mode until recessive bits are detected 128 times.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 01_B, the CHMDC[1:0] bits in the RSCANnCmCTR register (m = 0 to 2) are set to 10_B and the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated and the TEC[7:0] and REC[7:0] bits in the RSCANnCmSTS register are cleared to 00_H.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 10_B, the CHMDC[1:0] bits are set to 10_B and the RS-CANFD module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H.

When the BOM[1:0] bits are set to 11_B and the CHMDC[1:0] bits are set to 10_B while the RS-CANFD module is in the bus off state, the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H. However, if 11 consecutive recessive bits are detected 128 times and the RS-CANFD module has recovered to the error active state from the bus off state before the CHMDC[1:0] bits are set to 10_B, a bus off recovery interrupt request is generated.

If a program writes to the CHMDC[1:0] bit at the same time as the RS-CANFD module transition to channel halt mode (at bus off entry when the BOM[1:0] bits are 01B or at bus off end when the BOM[1:0] bits are 10_B), the program's writing takes precedence. Modify the BOM[1:0] bits only in channel reset mode.

TAIE Bit

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

ALIE Bit

When the ALF flag in the RSCANnCmERFL register is set to 1 with the ALIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BLIE Bit

When the BLF flag in the RSCANnCmERFL register is set to 1 with the BLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

OLIE Bit

When the OVLF flag in the RSCANnCmERFL register is set to 1 with the OLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BORIE Bit

When the BORF flag in the RSCANnCmERFL register is set to 1 with the BORIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BOEIE Bit

When the BOEF flag in the RSCANnCmERFL register is set to 1 with the BOEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EPIE Bit

When the EPF flag in the RSCANnCmERFL register is set to 1 with the EPIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EWIE Bit

When the EWF flag in the RSCANnCmERFL register is set to 1 with the EWIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BEIE Bit

When the BEF flag in the RSCANnCmERFL register is set to 1 with the BEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

RTBO Bit

Setting this bit to 1 in the bus off state forcibly changes the state from the bus off state to the error active state. This bit is automatically cleared to 0. Setting this bit to 1 clears the TEC[7:0] and REC[7:0] bits in the RSCANnCmSTS register to 00_H and also clears the BOSTS flag in the RSCANnCmSTS register to 0 (not in bus off state). The other registers remain unchanged. No bus off

recovery interrupt request is generated upon return from the bus off state in this case. Use this bit only when the BOM[1:0] bits in the RSCANnCmCTR register are 00_B (ISO11898-1 compliant).

A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the RSCAN module transitions to the error active state. Set this bit to 1 in channel communication mode.

CSLPR Bit

Setting this bit to 1 places the channel into channel stop mode.
Clearing this bit to 0 makes the channel exit channel stop mode.
This bit should not be modified in channel communication mode or channel halt mode.

CHMDC[1:0] Bits

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see Section 23.6.2, Channel Modes. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to 11_B. When the CAN module has automatically transitioned to channel halt mode based on the setting of the BOM[1:0] bits, the CHMDC[1:0] bits automatically become 10_B.

23.3.2.4 RSCANnCmSTS — Channel Status Register (m = 0 to 2)

Access: RSCANnCmSTS register can be read only in 32-bit units
RSCANnCmSTSL, RSCANnCmSTSH registers can be read only in 16-bit units
RSCANnCmSTSLL, RSCANnCmSTSHL, RSCANnCmSTSHH registers can be read only in 8-bit units

Address: RSCANnCmSTS: <RSCFDn_base> + 0008_H + (10_H × m)

RSCANnCmSTSL: <RSCFDn_base> + 0008_H + (10_H × m),
RSCANnCmSTSH: <RSCFDn_base> + $000A_H$ + (10_H × m)

RSCANnCmSTSLL: <RSCFDn_base> + 0008_H + (10_H × m),
RSCANnCmSTSHL: <RSCFDn_base> + $000A_H$ + (10_H × m),
RSCANnCmSTSHH: <RSCFDn_base> + $000B_H$ + (10_H × m)

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEC[7:0]								REC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.21 RSCANnCmSTS Register Contents

Bit Position	Bit Name	Function
31 to 24	TEC[7:0]	The transmit error counter (TEC) can be read.
23 to 16	REC[7:0]	The receive error counter (REC) can be read.
15 to 8	Reserved	These bits are read as the value after reset.
7	COMSTS	Communication Status Flag 0: Communication is not ready. 1: Communication is ready.
6	RECSTS	Receive Status Flag 0: Bus idle, in transmission or bus off state 1: In reception
5	TRMSTS	Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus off state
4	BOSTS	Bus Off Status Flag 0: Not in bus off state 1: In bus off state
3	EPSTS	Error Passive Status Flag 0: Not in error passive state 1: In error passive state
2	CSLPSTS	Channel Stop Status Flag 0: Not in channel stop mode 1: In channel stop mode
1	CHLTSTS	Channel Halt Status Flag 0: Not in channel halt mode 1: In channel halt mode
0	CRSTSTS	Channel Reset Status Flag 0: Not in channel reset mode 1: In channel reset mode

TEC[7:0] Bits

These bits contain the transmit error counter value. For transmit error counter increment/decrement conditions, see the CAN specification (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

REC[7:0] Bits

These bits contain the receive error counter value. For receive error counter increment/decrement conditions, see the CAN specifications (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

COMSTS Flag

This bit indicates that communication is ready.

This flag becomes 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag is cleared to 0 in channel reset mode or channel halt mode.

RECSTS Flag

This flag is set to 1 when reception has started, and is cleared to 0 when the bus has become idle or transmission has started.

TRMSTS Flag

This flag is set to 1 when transmission has started, and is cleared to 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

BOSTS Flag

This flag is set to 1 when the bus off state (TEC[7:0] > 255) is entered. It is cleared to 0 when the CAN module has exited the bus off state.

EPSTS Flag

This flag is set to 1 when the RS-CANFD module has entered the error passive state ((128 ≤ TEC[7:0] ≤ 255) or (128 ≤ REC[7:0])), It is cleared to 0 when the RS-CANFD module has exited the error passive state or has entered channel reset mode.

CSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel stop mode, and is cleared to 0 when the CAN module has returned from channel stop mode.

CHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel halt mode, and is cleared to 0 when the CAN module has returned from channel halt mode.

CRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel reset mode, and is cleared to 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 when the CAN module transitions from channel reset mode to channel stop mode.

23.3.2.5 RSCANnCmERFL — Channel Error Flag Register (m = 0 to 2)

Access: RSCANnCmERFL register can be read/written in 32-bit units
RSCANnCmERFLL register can be read/written in 16-bit units
RSCANnCmERFLH register is a read-only register that can be read in 16-bit units
RSCANnCmERFLLL, RSCANnCmERFLLH registers can be read/written in 8-bit units
RSCANnCmERFLHL, RSCANnCmERFLHH registers are the read-only registers that can be read in 8-bit units

Address: RSCANnCmERFL: <RSCFDn_base> + $000C_H$ + (10_H × m)

RSCANnCmERFLL: <RSCFDn_base> + $000C_H$ + (10_H × m),
RSCANnCmERFLH: <RSCFDn_base> + $000E_H$ + (10_H × m)

RSCANnCmERFLLL: <RSCFDn_base> + $000C_H$ + (10_H × m),
RSCANnCmERFLLH: <RSCFDn_base> + $000D_H$ + (10_H × m),
RSCANnCmERFLHL: <RSCFDn_base> + $000E_H$ + (10_H × m),
RSCANnCmERFLHH: <RSCFDn_base> + $000F_H$ + (10_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CRCREG[14:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.22 RSCANnCmERFL Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
30 to 16	CRCREG[14:0]	CRC Calculation Data A CRC value calculated based on the transmit message or receive message is indicated.
15	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
14	ADERR	ACK Delimiter Error Flag 0: No ACK delimiter error is detected. 1: ACK delimiter error is detected.
13	B0ERR	Dominant Bit Error Flag 0: No dominant bit error is detected. 1: Dominant bit error is detected.
12	B1ERR	Recessive Bit Error Flag 0: No recessive bit error is detected. 1: Recessive bit error is detected.
11	CERR	CRC Error Flag 0: No CRC error is detected. 1: CRC error is detected.
10	AERR	ACK Error Flag 0: No ACK error is detected. 1: ACK error is detected.
9	FERR	Form Error Flag 0: No form error is detected. 1: Form error is detected.

Table 23.22 RSCANnCmERFL Register Contents (2/2)

Bit Position	Bit Name	Function
8	SERR	Stuff Error Flag 0: No stuff error is detected. 1: Stuff error is detected.
7	ALF	Arbitration-lost Flag 0: No arbitration-lost is detected. 1: Arbitration-lost is detected.
6	BLF	Bus Lock Flag 0: No channel bus lock is detected. 1: Channel bus lock is detected.
5	OVLF	Overload Flag 0: No overload is detected. 1: Overload is detected.
4	BORF	Bus Off Recovery Flag 0: No bus off recovery is detected. 1: Bus off recovery is detected.
3	BOEF	Bus Off Entry Flag 0: No bus off entry is detected. 1: Bus off entry is detected.
2	EPF	Error Passive Flag 0: No error passive is detected. 1: Error passive is detected.
1	EWF	Error Warning Flag 0: No error warning is detected. 1: Error warning is detected.
0	BEF	Bus Error Flag 0: No channel bus error is detected. 1: Channel bus error is detected.

See the CAN specification (ISO11898-1) for a description of error occurrence conditions. To clear each flag of this register, the program must write a 0 to the corresponding bit. These flags cannot be set to 1 by the program. If any of these flags is set to 1 at the same time that the program writes 0 to the flag, the flag is still set to 1. The channel reset mode transition clears all of these flags to 0.

If the ERRD bit in the RSCANnCmCTR register is set to 0 (ie, only the flags for the first error event are displayed) and an error related to bits 14 to 8 of RSCANnCmERFL is detected, the flag bits are only set to 1 by the error event if bits 14 to 8 were all 0 at the time when the error occurred.

CRCREG[14:0] Flag

When the CTME bit in the RSCANnCmCTR register is set to 1 (communication test mode is enabled), the CRC value calculated based on the transmit or receive message can be read. When the CTME bit is set to 0 (communication test mode is disabled), these bits are always read as 0. This bit is always 0 in channel reset mode.

ADERR Flag

This flag is set to 1 when a form error has been detected in the ACK delimiter during transmission.

B0ERR Flag

This flag is set to 1 when a recessive bit has been detected though a dominant bit was transmitted.

B1ERR Flag

This flag is set to 1 when a dominant bit has been detected though a recessive bit was transmitted.

CERR Flag

This flag is set to 1 when a CRC error has been detected.

AERR Flag

This flag is set to 1 when an ACK error has been detected.

FERR Flag

This flag is set to 1 when a form error has been detected.

SERR Flag

This flag is set to 1 when a stuff error has been detected.

ALF Flag

This flag is set to 1 when an arbitration-lost has been detected.

BLF Flag

This flag is set to 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, detection of a dominant lock is restarted when either of the following conditions is met.

- A recessive bit is detected after the BLF bit has been cleared from 1 to 0.
- The CAN module transitions to channel reset mode and returns to channel communication mode after the BLF bit has been cleared from 1 to 0.

OVLF Flag

This flag is set to 1 when the overload frame transmit condition has been detected when performing reception or transmission.

BORF Flag

This flag is set to 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus off state. However, this flag is not set to 1 if the CAN module returns from the bus off state in any of the following ways before 11 consecutive recessive bits are detected 128 times.

- The CHMDC[1:0] bits in the RSCANnCmCTR register are set to 01_B (channel reset mode).
- The RTBO bit in the RSCANnCmCTR register is set to 1 (forcible return from the bus off state is made).
- The BOM[1:0] bits in the RSCANnCmCTR register are set to 01_B (transition to channel halt mode at bus off entry).
- The CHMDC[1:0] bits in the RSCANnCmCTR register are set to 10_B (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the BOM[1:0] bits set to 11_B (transition to channel halt mode upon a request from the program during bus off).

BOEF Flag

This flag is set to 1 when the bus off state is entered (TEC[7:0] value > 255). This flag is also set to 1 if the bus off state is entered when the BOM[1:0] bits in the RSCANnCmCTR register (m = 0 to 2) set to 01_B (transition to channel halt mode at bus off entry).

EPF Flag

This flag becomes 1 when the error passive state is reached ((128 ≤ TEC[7:0] ≤ 255) or (128 ≤ REC[7:0])).
This flag becomes 1 only when the REC[7:0] or TEC[7:0] value first exceeds 127. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 127, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 127 or less and then the REC[7:0] or TEC[7:0] value exceeds 127 again.

EWF Flag

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 95. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 95, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 95 or less and then the REC[7:0] or TEC[7:0] value exceeds 95 again.

BEF Flag

This flag is set to 1 when any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags in the RSCANnCmERFL register is set to 1.

NOTE

To clear the flag of this register to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

23.3.3 Details of Global-Related Registers

23.3.3.1 RSCANnGCFG — Global Configuration Register

Access: RSCANnGCFG register can be read/written in 32-bit units
RSCANnGCFGL, RSCANnGCFGH registers can be read/written in 16-bit units
RSCANnGCFGLL, RSCANnGCFGLH, RSCANnGCFGHL, RSCANnGCFGHH registers can be read/written in 8-bit units

Address: RSCANnGCFG: <RSCFDn_base> + 0084_H

RSCANnGCFGL: <RSCFDn_base> + 0084_H,
RSCANnGCFGH: <RSCFDn_base> + 0086_H

RSCANnGCFGLL: <RSCFDn_base> + 0084_H,
RSCANnGCFGLH: <RSCFDn_base> + 0085_H,
RSCANnGCFGHL: <RSCFDn_base> + 0086_H,
RSCANnGCFGHH: <RSCFDn_base> + 0087_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITRCP[15:0}															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSBTCS[2:0]			TSSS	TSP[3:0]				TMTSCE	EEFE	—	DCS	MME	DRE	DCE	TPRI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 23.23 RSCANnGCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	ITRCP[15:0]	Interval Timer Prescaler Set When these bits are set to M, the pclk is divided by M. Setting 0000_H is prohibited when the interval timer is in use.
15 to 13	TSBTCS[2:0]	Timestamp Clock Source Select b15 b14 b13 0 0 0: Channel 0 bit time clock 0 0 1: Channel 1 bit time clock 0 1 0: Channel 2 bit time clock 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited
12	TSSS	Timestamp Source Select 0: pclk/2*1 1: Bit time clock

Table 23.23 RSCANnGCFG Register Contents (2/2)

Bit Position	Bit Name	Function
11 to 8	TSP[3:0]	Timestamp Clock Source Division b11 b10 b9 b8 0 0 0 0: Not divided 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 256 1 0 0 1: Divided by 512 1 0 1 0: Divided by 1024 1 0 1 1: Divided by 2048 1 1 0 0: Divided by 4096 1 1 0 1: Divided by 8192 1 1 1 0: Divided by 16384 1 1 1 1: Divided by 32768
7	TMTSCE	Transmission Timestamp Enable 0: Transmission timestamp is disabled. 1: Transmission timestamp is enabled.
6	EEFE	ECC Error Flag Enable 0: The ECC error flag is disabled. 1: The ECC error flag is enabled.
5	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
4	DCS	CAN Clock Source Select*2 0: clkc 1: clk_xincan
3	MME	Mirror Function Enable 0: Mirror function is disabled. 1: Mirror function is enabled.
2	DRE	DLC Replacement Enable 0: DLC replacement is disabled. 1: DLC replacement is enabled.
1	DCE	DLC Check Enable 0: DLC check is disabled. 1: DLC check is enabled.
0	TPRI	Transmit Priority Select 0: ID priority 1: Transmit buffer number priority

Note 1. When specifying pclk/2 as the timestamp counter count source, set bits TSBTCS[2:0] to 000_B.

Note 2. For the CAN clock frequency settings, see Table 23.8, Setting Range example of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/D1L/D1M.

Modify the RSCANnGCFG register only in global reset mode.

ITRCP[15:0] Bits

These bits are used to set a clock source division value of the interval timer for FIFO buffers. See Section 23.8.3.1, Interval Transmission Function.

TSBTCS[2:0] Bits

When the TSSS bit is 1, these bits are used to select the channel of the bit time clock that will be the clock source of the timestamp counter.

TSSS Bit

This bit is used to select a clock source of the timestamp counter.

TSP[3:0] Bits

A clock obtained by dividing the clock source selected with the TSBTCS[2:0] bits and TSSS bit according to the TSP[3:0] bits is used as the timestamp counter count source.

TMTSCE Bit

Setting this bit to 1 makes it possible to store the timestamp of a transmitted message in the transmit history buffer. The timestamp is stored in TMTS[15:0] bits in the RSCANnTHLACCm register.

EEFE Bit

Setting this bit to 1 sets the EEFm bit in the RSCANnGERFL register to 1 when a 2-bit ECC error is detected during the transmission priority determination. At this time, the message in which a 2-bit ECC error was detected is not transmitted.

DCS Bit

When this bit is set to 0, clkc is used as the clock source of the CAN clock (fCAN).

When this bit is set to 1, clk_xincan is used as the clock source of the CAN clock (fCAN).

For the CAN clock frequency settings, see Table 23.8, Setting Range example of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/D1L/D1M.

MME Bit

Setting this bit to 1 makes the mirror function available.

DRE Bit

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of 00_H is stored in each data byte beyond the DLC value of the receive rule.

The DLC replacement function is only available when the DCE bit is set to 1 (DLC check is enabled).

DCE Bit

Setting this bit to 1 makes the DLC check function available. When disabling the DLC check function, set the GAFLDLC[3:0] bits in the RSCANnGAFLP0_j register to 0000_B before clearing the DCE bit in the RSCANnGCFG register to 0.

TPRI Bit

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO11898-1 specifications). When this bit is set to 1, transmit buffer number priority is selected and the transmit buffer with the smallest number of those has the highest priority.

While the transmit queue is in use, this bit should be set to 0.

23.3.3.2 RSCANnGCTR — Global Control Register

Access: RSCANnGCTR register can be read/written in 32-bit units
RSCANnGCTRL, RSCANnGCTRH registers can be read/written in 16-bit units
RSCANnGCTRLL, RSCANnGCTRLH, RSCANnGCTRHL registers can be read/written in 8-bit units

Address: RSCANnGCTR: <RSCFDn_base> + 0088_H

RSCANnGCTRL: <RSCFDn_base> + 0088_H,
RSCANnGCTRH: <RSCFDn_base> + $008A_H$

RSCANnGCTRLL: <RSCFDn_base> + 0088_H,
RSCANnGCTRLH: <RSCFDn_base> + 0089_H,
RSCANnGCTRHL: <RSCFDn_base> + $008A_H$

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 23.24 RSCANnGCTR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
16	TSRST	Timestamp Counter Reset Setting the TSRST bit to 1 resets the timestamp counter. This bit is always read as 0.
15 to 11	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
10	THLEIE	Transmit History Buffer Overflow Interrupt Enable 0: Transmit history buffer overflow interrupt is disabled. 1: Transmit history buffer overflow interrupt is enabled.
9	MEIE	FIFO Message Lost Interrupt Enable 0: FIFO message lost interrupt is disabled. 1: FIFO message lost interrupt is enabled.
8	DEIE	DLC Error Interrupt Enable 0: DLC error interrupt is disabled. 1: DLC error interrupt is enabled.
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	GSLPR	Global Stop Mode 0: Other than global stop mode 1: Global stop mode
1, 0	GMDC[1:0]	Global Mode Select b1 b0 0 0: Global operating mode 0 1: Global reset mode 1 0: Global test mode 1 1: Setting prohibited

TSRST Bit

This bit is used to reset the timestamp counter. When this bit is set to 1, the RSCANnGTSC register is cleared to 0000_H.

THLEIE Bit

When the THLEIE bit is set to 1 and the THLES flag in the RSCANnGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

MEIE Bit

When the MEIE bit is set to 1 and the MES flag in the RSCANnGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

DEIE Bit

When the DEIE bit is set to 1 and the DEF flag in the RSCANnGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

GSLPR Bit

Setting this bit to 1 places the RSCAN module into global stop mode.
Clearing this bit to 0 makes the RSCAN module leave from global stop mode.
This bit should not be modified in global operating mode or global test mode.

GMDC[1:0] Bits

These bits are used to select the mode of entire RS-CANFD module (global operating mode, global reset mode, or global test mode). For details, see Section 23.6.1, Global Modes. Setting the GSLPR bit to 1 when in global reset mode places the RS-CANFD module into global stop mode.

23.3.3.3 RSCANnGSTS — Global Status Register

Access: RSCANnGSTS register can be read only in 32-bit units
RSCANnGSTSL register is a read-only register that can be read only in 16-bit units
RSCANnGSTSLL register is a read-only register that can be read only in 8-bit units

Address: RSCANnGSTS: <RSCFDn_base> + $008C_H$

RSCANnGSTSL: <RSCFDn_base> + $008C_H$

RSCANnGSTSLL: <RSCFDn_base> + $008C_H$

Value after reset: $0000\ 000D_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	GRAMINIT	GSLPSTS	GHLTSTS	GRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.25 RSCANnGSTS Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	GRAMINIT	CAN RAM Initialization Status Flag 0: CAN RAM initialization is completed. 1: CAN RAM initialization is ongoing.
2	GSLPSTS	Global Stop Status Flag 0: Not in global stop mode 1: In global stop mode
1	GHLTSTS	Global Test Status Flag 0: Not in global test mode 1: In global test mode
0	GRSTSTS	Global Reset Status Flag 0: Not in global reset mode 1: In global reset mode

GRAMINIT Flag

This flag indicates the initialization status of the CAN RAM.

This flag is set to 1 after the MCU has been reset, and is cleared to 0 when CAN RAM initialization is completed.

GSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to global stop mode, and is cleared to 0 when the CAN module has returned from global stop mode.

GHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global test mode, and is cleared to 0 when the CAN module has exited global test mode.

GRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global reset mode, and is cleared to 0 when the CAN module has exited global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

23.3.3.4 RSCANnGERFL — Global Error Flag Register

Access: RSCANnGERFL register can be read/written in 32-bit units
RSCANnGERFLL, RSCANnGERFLH registers can be read/written in 16-bit units
RSCANnGERFLLL, RSCANnGERFLHL registers can be read/written in 8-bit units

Address: RSCANnGERFL: <RSCFDn_base> + 0090_H

RSCANnGERFLL: <RSCFDn_base> + 0090_H,
RSCANnGERFLH: <RSCFDn_base> + 0092_H

RSCANnGERFLLL: <RSCFDn_base> + 0090_H,
RSCANnGERFLHL: <RSCFDn_base> + 0092_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	EEF2	EEF1	EEF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	THLES	MES	DEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.26 RSCANnGERFL Register Contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When read, the value after reset is returned. When writing these bits, write the value after reset.
18	EEF2	ECC Error Flag for Channel 2 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
17	EEF1	ECC Error Flag for Channel 1 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
16	EEF0	ECC Error Flag for Channel 0 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
15, 14	Reserved	When read, the value after reset is returned. When writing these bits, write the value after reset.
13 to 8	Reserved	When read, the undefined value is returned. When writing these bits, write the value after reset.
7, 6	Reserved	When read, the value after reset is returned. When writing these bits, write the value after reset.
5	Reserved	When read, the undefined value is returned. When writing these bits, write the value after reset.
4, 3	Reserved	When read, the value after reset is returned. When writing these bits, write the value after reset.
2	THLES	Transmit History Buffer Overflow Status Flag 0: No transmit history buffer overflow has occurred. 1: A transmit history buffer overflow has occurred.
1	MES	FIFO Message Lost Status Flag 0: No FIFO message lost error has occurred. 1: A FIFO message lost error has occurred.

Table 23.26 RSCANnGERFL Register Contents

Bit Position	Bit Name	Function
0	DEF	DLC Error Flag 0: No DLC error has occurred. 1: A DLC error has occurred.

All flags in the RSCANnGERFL register are cleared to 0 in global reset mode.

EEFm Flag

While the EEFE bit in the RSCANnGCFG register is 1, when a 2-bit ECC error is detected during the transmission priority determination of channel m (m = 0 to 2), the EEFm flag is set to 1, disabling message transmission. This flag can be cleared to 0 by writing 0 by the program.

THLES Flag

The THLES flag is set to 1 when any one of the THLELT flags in the RSCANnTHLSTSm register (m = 0 to 2) is set to 1.

This flag is cleared to 0 when the THLELT flags of all channels are set to 0.

MES Flag

The MES flag is set to 1 when any one of the RFMLT flags in the RSCANnRFSTSx register (x = 0 to 7) or the CFMLT flags in the RSCANnCFSTSk register (k = 0 to 8) is set to 1.

This flag is cleared to 0 when all RFMLT flags and CFMLT flags are set to 0.

DEF Flag

The DEF flag is set to 1 when an error has been detected during the DLC check. The program can clear this flag by writing 0 to this bit.

NOTE

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

23.3.3.5 RSCANnGTSC — Global Timestamp Counter Register

Access: RSCANnGTSC register can be read only in 32-bit units.
RSCANnGTSCL register is a read-only register that can be read only in 16-bit units.

Address: RSCANnGTSC: <RSCFDn_base> + 0094_H

RSCANnGTSCL: <RSCFDn_base> + 0094_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.27 RSCANnGTSC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset.
15 to 0	TS[15:0]	Timestamp Value The timestamp counter value can be read. Counter Value: 0000_H to $FFFF_H$

TS[15:0] Bits

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. When the SOF is detected, the TS[15:0] value is captured and later stored in the receive buffer or the FIFO buffer.

Furthermore, while the TMTSCE bit in the RSCANnGCFG register is 1, the TS[15: 0] value is stored in the transmit history buffer. The timestamp counter is initialized in global reset mode.

The timestamp counter start timing and stop timing depend on the count source.

- When the TSSS bit in the RSCANnGCFG register is 0 (pclk):
  The timestamp counter starts counting when the RSCAN module has transitioned to global operating mode.
  This counter stops counting when the RSCAN module has transitioned to global stop mode or global test mode.
- When the TSSS bit is 1 (CANm bit time clock):
  The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.
  This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

23.3.3.6 RSCANnGTINTSTS0 — Global TX Interrupt Status Register 0

Access: RSCANnGTINTSTS0 register can be read only in 32-bit units
RSCANnGTINTSTS0L, RSCANnGTINTSTS0H registers can be read only in 16-bit units
RSCANnGTINTSTS0LL, RSCANnGTINTSTS0LH, RSCANnGTINTSTS0HL registers can be read only in 8-bit units

Address: RSCANnGTINTSTS0: <RSCFDn_base> + 0460_H

RSCANnGTINTSTS0L: <RSCFDn_base> + 0460_H,
RSCANnGTINTSTS0H: <RSCFDn_base> + 0462_H

RSCANnGTINTSTS0LL: <RSCFDn_base> + 0460_H,
RSCANnGTINTSTS0LH: <RSCFDn_base> + 0461_H,
RSCANnGTINTSTS0HL: <RSCFDn_base> + 0462_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	THIF2	CFTIF2	TQIF2	TAIF2	TSIF2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R*1	R*1	R*1	R*1	R*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	—	—	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R*1	R*1	R*1	R*1	R*1	R	R	R	R*1	R*1	R*1	R*1	R*1

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 23.28 RSCANnGTINTSTS0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 21	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
20	THIF2	Channel 2 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
19	CFTIF2	Channel 2 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
18	TQIF2	Channel 2 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
17	TAIF2	Channel 2 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
16	TSIF2	Channel 2 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
15 to 13	Reserved	These bits are read as the value after reset.
12	THIF1	Channel 1 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
11	CFTIF1	Channel 1 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
10	TQIF1	Channel 1 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.

Table 23.28 RSCANnGTINTSTS0 Register Contents (2/2)

Bit Position	Bit Name	Function
9	TAIF1	Channel 1 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
8	TSIF1	Channel 1 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
7 to 5	Reserved	These bits are read as the value after reset.
4	THIF0	Channel 0 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF0	Channel 0 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF0	Channel 0 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF0	Channel 0 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
0	TSIF0	Channel 0 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

TSIFm Bits

The TSIFm bit is set to 1 when the TMIEp bit in the RSCANnTMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the RSCANnTMSTSp register are set to 10_B (transmit completed without abort request) or 11_B (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to 00_B under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIEp bit to 0 also clears this flag to 0.

TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RSCANnCmCTR register is 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RSCANnTMSTSp register are set to 01_B (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to 00_B after the transmit abort is completed.

TQIFm Bits

When the TXQIE bit in the RSCANnTXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQIF bit in the RSCANnTXQSTSm register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQIF bit (transmit queue interrupt request) in the RSCANnTXQSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the TXQIE bit is cleared to 0.

CFTIFm Bits

When the CFTXIE bit in the RSCANnCFCCk register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RSCANnCFSTSk register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

THIFm Bits

When the THLIE bit in the RSCANnTHLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RSCANnTHLSTSm register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RSCANnTHLSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the THLIE bit is cleared to 0.

23.3.4 Details of Receive Rule-related Registers

23.3.4.1 RSCANnGAFLECTR — Receive Rule Entry Control Register

Access: RSCANnGAFLECTR register can be read/written in 32-bit units
RSCANnGAFLECTRL register can be read/written in 16-bit units
RSCANnGAFLECTRLL, RSCANnGAFLECTRLH registers can be read/written in 8-bit units

Address: RSCANnGAFLECTR: <RSCFDn_base> + 0098_H

RSCANnGAFLECTRL: <RSCFDn_base> + 0098_H

RSCANnGAFLECTRLL: <RSCFDn_base> + 0098_H,
RSCANnGAFLECTRLH: <RSCFDn_base> + 0099_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	AFLDAE	—	—	—	AFLPN[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 23.29 RSCANnGAFLECTR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
8	AFLDAE	Receive Rule Table Write Enable 0: Receive rule table write is disabled. 1: Receive rule table write is enabled.
7 to 5	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
4 to 0	AFLPN[4:0]	Receive Rule Table Page Number Configuration A page number can be selected from a range of page 0 (00000_B) to page 11 (01011_B).

AFLDAE Bit

Setting this bit to 0 disables the write to the receive rule table. After writes to the receive rule table are completed, set this bit to 0 to disable the write to the table. The receive rule table can be read regardless of the value of this bit.

Set the AFLDAE bit to 1 only in global reset mode.

AFLPN[4:0] Bits

These bits are used to set the page number of the receive rule table. Sixteen receive rules can be set per page.

Set these bits to a value within the range of 00000_B to 01011_B.

23.3.4.2 RSCANnGAFLCFG0 — Receive Rule Configuration Register 0

Access: RSCANnGAFLCFG0 register can be read/written in 32-bit units
RSCANnGAFLCFG0L, RSCANnGAFLCFG0H registers can be read/written in 16-bit units
RSCANnGAFLCFG0LH, RSCANnGAFLCFG0HL, RSCANnGAFLCFG0HH registers can be read/written in 8-bit units

Address: RSCANnGAFLCFG0: <RSCFDn_base> + $009C_H$

RSCANnGAFLCFG0L: <RSCFDn_base> + $009C_H$,
RSCANnGAFLCFG0H: <RSCFDn_base> + $009E_H$

RSCANnGAFLCFG0LH: <RSCFDn_base> + $009D_H$,
RSCANnGAFLCFG0HL: <RSCFDn_base> + $009E_H$,
RSCANnGAFLCFG0HH: <RSCFDn_base> + $009F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC0[7:0]								RNC1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RNC2[7:0]								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 23.30 RSCANnGAFLCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC0[7:0]	Number of Rules for Channel 0 Set the number of receive rules exclusively used for channel 0.
23 to 16	RNC1[7:0]	Number of Rules for Channel 1 Set the number of receive rules exclusively used for channel 1.
15 to 8	RNC2[7:0]	Number of Rules for Channel 2 Set the number of receive rules exclusively used for channel 2.
7 to 0	Reserved	These bits are read as the value after reset. The write value should be the value after reset.

Modify the RSCANnGAFLCFG0 register only in global reset mode.

Up to 64 × (number of channels) rules can be registered in the receive rule table for the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total of the number of rules allocated to each channel is not larger than the number of rules that can be registered for the entire unit.

RNC0[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 0 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC1[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 1 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC2[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 2 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

23.3.4.3 RSCANnGAFLIDj — Receive Rule ID Register (j = 0 to 15)

Access: RSCANnGAFLIDj register can be read/written in 32-bit units
RSCANnGAFLIDjL, RSCANnGAFLIDjH registers can be read/written in 16-bit units
RSCANnGAFLIDjLL, RSCANnGAFLIDjLH, RSCANnGAFLIDjHL, RSCANnGAFLIDjHH registers can be read/written in 8-bit units

Address: RSCANnGAFLIDj: <RSCFDn_base> + 0500_H + (10_H × j)

RSCANnGAFLIDjL: <RSCFDn_base> + 0500_H + (10_H × j),
RSCANnGAFLIDjH: <RSCFDn_base> + 0502_H + (10_H × j)

RSCANnGAFLIDjLL: <RSCFDn_base> + 0500_H + (10_H × j),
RSCANnGAFLIDjLH: <RSCFDn_base> + 0501_H + (10_H × j),
RSCANnGAFLIDjHL: <RSCFDn_base> + 0502_H + (10_H × j),
RSCANnGAFLIDjHH: <RSCFDn_base> + 0503_H + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.31 RSCANnGAFLIDj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDE	IDE Select 0: Standard ID 1: Extended ID
30	GAFLRTR	RTR Select 0: Data frame 1: Remote frame
29	GAFLLB	Receive Rule Target Message Select 0: When a message transmitted from another CAN node is received 1: When the own transmitted message is received
28 to 0	GAFLID[28:0]	ID Set the ID of the receive rule. For the standard ID, set the ID in bits b10 to b0 and set bits b28 to b11 to 0.

Modify the RSCANnGAFLIDj register when the AFLDAE bit in the RSCANnGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDE Bit

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

GAFLRTR Bit

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

GAFLLB Bit

When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.

When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when the CAN node is receiving its own transmitted messages.

GAFLID[28:0] Bits

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID of the received message during the acceptance filter processing.

23.3.4.4 RSCANnGAFLMj — Receive Rule Mask Register (j = 0 to 15)

Access: RSCANnGAFLMj register can be read/written in 32-bit units
RSCANnGAFLMjL, RSCANnGAFLMjH registers can be read/written in 16-bit units
RSCANnGAFLMjLL, RSCANnGAFLMjLH, RSCANnGAFLMjHL, RSCANnGAFLMjHH registers can be read/written in 8-bit units

Address: RSCANnGAFLMj: <RSCFDn_base> + 0504_H + (10_H × j)

RSCANnGAFLMjL: <RSCFDn_base> + 0504_H + (10_H × j),
RSCANnGAFLMjH: <RSCFDn_base> + 0506_H + (10_H × j)

RSCANnGAFLMjLL: <RSCFDn_base> + 0504_H + (10_H × j),
RSCANnGAFLMjLH: <RSCFDn_base> + 0505_H + (10_H × j),
RSCANnGAFLMjHL: <RSCFDn_base> + 0506_H + (10_H × j),
RSCANnGAFLMjHH: <RSCFDn_base> + 0507_H + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDEM	GAFLRTRM	—	GAFLIDM[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLIDM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.32 RSCANnGAFLMj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDEM	IDE Mask 0: The IDE bit is not compared. 1: The IDE bit is compared.
30	GAFLRTRM	RTR Mask 0: The RTR bit is not compared. 1: The RTR bit is compared
29	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
28 to 0	GAFLIDM[28:0]	ID Mask 0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.

Modify the RSCANnGAFLMj register when the AFLDAE bit in the RSCANnGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDEM Bit

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDE bit in the RSCANnGAFLIDj register.

When this bit is cleared to 0, the IDs of all the receive messages and the specified IDs are regarded as matched. To set the GAFLIDEM bit to 0, set the GAFLIDM[28:0] bits to all 0 at the same time.

GAFLRTRM Bit

This bit is used to mask the RTR bit of the receive rule.

GAFLIDM[28:0] Bits

These bits are used to mask the corresponding ID bit of the receive rule.

23.3.4.5 RSCANnGAFLP0_j — Receive Rule Pointer 0 Register (j = 0 to 15)

Access: RSCANnGAFLP0_j register can be read/written in 32-bit units
RSCANnGAFLP0_jL, RSCANnGAFLP0_jH registers can be read/written in 16-bit units
RSCANnGAFLP0_jLH, RSCANnGAFLP0_jHL, RSCANnGAFLP0_jHH registers can be read/written in 8-bit units

Address: RSCANnGAFLP0_j: <RSCFDn_base> + 0508_H + (10_H × j)

RSCANnGAFLP0_jL: <RSCFDn_base> + 0508_H + (10_H × i),
RSCANnGAFLP0_jH: <RSCFDn_base> + $050A_H$ + (10_H × j)

RSCANnGAFLP0_jLH: <RSCFDn_base> + 0509_H + (10_H × j),
RSCANnGAFLP0_jHL: <RSCFDn_base> + $050A_H$ + (10_H × j),
RSCANnGAFLP0_jHH: <RSCFDn_base> + $050B_H$ + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLDLC[3:0]				GAFLPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLRMV	GAFLRMDP[6:0]							—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 23.33 RSCANnGAFLP0_j Register Contents

Bit Position	Bit Name	Function
31 to 28	GAFLDLC[3:0]	Receive Rule DLC b31 b30 b29 b28 0 0 0 0: DLC check is disabled. 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	GAFLPTR[11:0]	Receive Rule Label Set the 12-bit label information.
15	GAFLRMV	Receive Buffer Enable 0: No receive buffer is used. 1: A receive buffer is used.
14 to 8	GAFLRMDP[6:0]	Receive Buffer Number Select Set the receive buffer number to store receive messages.
7 to 0	Reserved	These bits are read as the value after reset. The write value should be the value after reset.

Modify the RSCANnGAFLP0_j register when the AFLDAE bit in the RSCANnGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLDLC[3:0] Bits

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000_B disables the DLC check function allowing messages with any data length to pass the DLC check.

GAFLPTR[11:0] Bits

These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

GAFLRMV Bit

When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

GAFLRMDP[6:0] Bits

These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the NRXMB[7:0] bits in the RSCANnRMNB register.

23.3.4.6 RSCANnGAFLP1_j — Receive Rule Pointer 1 Register (j = 0 to 15)

Access: RSCANnGAFLP1_j register can be read/written in 32-bit units
RSCANnGAFLP1_jL, RSCANnGAFLP1_jH registers can be read/written in 16-bit units
RSCANnGAFLP1_jLL, RSCANnGAFLP1_jLH, RSCANnGAFLP1_jHL, RSCANnGAFLP1_jHH registers can be read/written in 8-bit units

Address: RSCANnGAFLP1_j: <RSCFDn_base> + $050C_H$ + (10_H × j)

RSCANnGAFLP1_jL: <RSCFDn_base> + $050C_H$ + (10_H × j),
RSCANnGAFLP1_jH: <RSCFDn_base> + $050E_H$ + (10_H × j)

RSCANnGAFLP1_jLL: <RSCFDn_base> + $050C_H$ + (10_H × j),
RSCANnGAFLP1_jLH: <RSCFDn_base> + $050D_H$ + (10_H × j),
RSCANnGAFLP1_jHL: <RSCFDn_base> + $050E_H$ + (10_H × j),
RSCANnGAFLP1_jHH: <RSCFDn_base> + $050F_H$ + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GAFLFDP[25:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLFDP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.34 RSCANnGAFLP1_j Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
25 to 8	GAFLFDP[25:8]	Transmit/Receive FIFO Buffer k Select (Bit position –8 = target transmit/receive FIFO buffer number k) 0: Transmit/receive FIFO buffer is not selected. 1: Transmit/receive FIFO buffer is selected.
7 to 0	GAFLFDP[7:0]	Receive FIFO Buffer x Select (Bit position = target receive FIFO buffer number x) 0: Receive FIFO buffer is not selected. 1: Receive FIFO buffer is selected.

Modify the RSCANnGAFLP1_j register when the AFLDAE bit in the RSCANnGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLFDP [25:0] Bits

These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to eight FIFO buffers are selectable. However, when the GAFLRMV bit in the RSCANnGAFLP0_j register is set to 1 (a message is stored in the receive buffer), up to seven FIFO buffers can be selected. Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFM[1:0] bits in the RSCANnCFCCk register are set to 00_B (receive mode) or 10_B (gateway mode) are selectable.

23.3.5 Details of Receive Buffer-Related Registers

23.3.5.1 RSCANnRMNB — Receive Buffer Number Register

Access: RSCANnRMNB register can be read/written in 32-bit units
RSCANnRMNBL register can be read/written in 16-bit units
RSCANnRMNBLL register can be read/written in 8-bit units

Address: RSCANnRMNB: <RSCFDn_base> + $00A4_H$

RSCANnRMNBL: <RSCFDn_base> + $00A4_H$

RSCANnRMNBLL: <RSCFDn_base> + $00A4_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	NRXMB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.35 RSCANnRMNB Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
7 to 0	NRXMB[7:0]	Receive Buffer Number Configuration Set the number of receive buffers. Set a value of 0 to 48.

Modify the RSCANnRMNB register only in global reset mode.

NRXMB[7:0] Bits

These bits are used to set the total number of receive buffers of the RS-CANFD module. The maximum value is 16 × (number of channels).

Setting these bits all to 0 makes receive buffers unavailable.

23.3.5.2 RSCANnRMNDy — Receive Buffer New Data Register (y = 0, 1)

Access: RSCANnRMNDy register can be read/written in 32-bit units
RSCANnRMNDyL, RSCANnRMNDyH registers can be read/written in 16-bit units
RSCANnRMNDyLL, RSCANnRMNDyLH, RSCANnRMNDyHL, RSCANnRMNDyHH registers can be read/written in 8-bit units

Address: RSCANnRMNDy: <RSCFDn_base> + $00A8_H$ + (04_H × y)

RSCANnRMNDyL: <RSCFDn_base> + $00A8_H$ + (04_H × y),
RSCANnRMNDyH: <RSCFDn_base> + $00AA_H$ + (04_H × y)

RSCANnRMNDyLL: <RSCFDn_base> + $00A8_H$ + (04_H × y),
RSCANnRMNDyLH: <RSCFDn_base> + $00A9_H$ + (04_H × y),
RSCANnRMNDyHL: <RSCFDn_base> + $00AA_H$ + (04_H × y),
RSCANnRMNDyHH: <RSCFDn_base> + $00AB_H$ + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMNSq (q = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMNSq (q = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.36 RSCANnRMNDy Register Contents

Bit Position	Bit Name	Function
31 to 16	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 31 to y × 32 + 16) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.
15 to 0	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 15 to y × 32 + 0) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.

Write 0 to the RSCANnRMNDy register in global operating mode or global test mode.

RMNSq Flags (q = 0 to 47)

Each RMNSq flag is set to 1 when the processing for storing a message in the corresponding receive buffer starts.

To clear a flag to 0, the program must write 0 to the flag. Use a store instruction to write “0” to the flag and “1” to other flags. These bits cannot be set to 0 while a message is being stored. It takes ten clock cycles of pclk to store a message.

These flags are cleared to 0 in global reset mode.

23.3.5.3 RSCANnRMIDq — Receive Buffer ID Register (q = 0 to 47)

Access: RSCANnRMIDq register can be read only in 32-bit units
RSCANnRMIDqL, RSCANnRMIDqH registers can be read only in 16-bit units
RSCANnRMIDqLL, RSCANnRMIDqLH, RSCANnRMIDqHL, RSCANnRMIDqHH registers can be read only in 8-bit units

Address: RSCANnRMIDq: <RSCFDn_base> + 0600_H + (10_H × q)

RSCANnRMIDqL: <RSCFDn_base> + 0600_H + (10_H × q),
RSCANnRMIDqH: <RSCFDn_base> + 0602_H + (10_H × q)

RSCANnRMIDqLL: <RSCFDn_base> + 0600_H + (10_H × q),
RSCANnRMIDqLH: <RSCFDn_base> + 0601_H + (10_H × q),
RSCANnRMIDqHL: <RSCFDn_base> + 0602_H + (10_H × q),
RSCANnRMIDqHH: <RSCFDn_base> + 0603_H + (10_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMIDE	RMRTR	—	RMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.37 RSCANnRMIDq Register Contents

Bit Position	Bit Name	Function
31	RMIDE	Receive Buffer IDE 0: Standard ID 1: Extended ID
30	RMRTR	Receive Buffer RTR 0: Data frame 1: Remote frame
29	Reserved	These bits are read as the value after reset.
28 to 0	RMID[28:0]	Receive Buffer ID Data These bits contain the standard ID or extended ID of the received message. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RMIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

RMRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer.

RMID[28:0] Bits

These bits contain the ID of the message stored in the receive buffer.

23.3.5.4 RSCANnRMPTRq — Receive Buffer Pointer Register (q = 0 to 47)

Access: RSCANnRMPTRq register can be read only in 32-bit units
RSCANnRMPTRqL, RSCANnRMPTRqH registers can be read only in 16-bit units
RSCANnRMPTRqLL, RSCANnRMPTRqLH, RSCANnRMPTRqHL, RSCANnRMPTRqHH registers can be read only in 8-bit units

Address: RSCANnRMPTRq: <RSCFDn_base> + 0604_H + (10_H × q)

RSCANnRMPTRqL: <RSCFDn_base> + 0604_H + (10_H × q),
RSCANnRMPTRqH: <RSCFDn_base> + 0606_H + (10_H × q)

RSCANnRMPTRqLL: <RSCFDn_base> + 0604_H + (10_H × q),
RSCANnRMPTRqLH: <RSCFDn_base> + 0605_H + (10_H × q),
RSCANnRMPTRqHL: <RSCFDn_base> + 0606_H + (10_H × q),
RSCANnRMPTRqHH: <RSCFDn_base> + 0607_H + (10_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDLC[3:0]				RMPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.38 RSCANnRMPTRq Register Contents

Bit Position	Bit Name	Function
31 to 28	RMDLC[3:0]	Receive Buffer DLC Data b31 b30 b29 b28 0 0 0 0: No data byte 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	RMPTR[11:0]	Receive Buffer Label Data Label information of the received message.
15 to 0	RMTS[15:0]	Receive Buffer Timestamp Data Timestamp value of the received message.

RMDLC[3:0] Bits

These bits indicate the data length of the message stored in the receive buffer.

RMPTR[11:0] Bits

These bits indicate the label information of the message stored in the receive buffer.

RMTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the receive buffer.

23.3.5.5 RSCANnRMDF0_q — Receive Buffer Data Field 0 Register (q = 0 to 47)

Access: RSCANnRMDF0_q register can be read only in 32-bit units
RSCANnRMDF0_qL, RSCANnRMDF0_qH registers can be read only in 16-bit units
RSCANnRMDF0_qLL, RSCANnRMDF0_qLH, RSCANnRMDF0_qHL, RSCANnRMDF0_qHH registers can be read only in 8-bit units

Address: RSCANnRMDF0_q: <RSCFDn_base> + 0608_H + (10_H × q)

RSCANnRMDF0_qL: <RSCFDn_base> + 0608_H + (10_H × q),
RSCANnRMDF0_qH: <RSCFDn_base> + $060A_H$ + (10_H × q)

RSCANnRMDF0_qLL: <RSCFDn_base> + 0608_H + (10_H × q),
RSCANnRMDF0_qLH: <RSCFDn_base> + 0609_H + (10_H × q),
RSCANnRMDF0_qHL: <RSCFDn_base> + $060A_H$ + (10_H × q),
RSCANnRMDF0_qHH: <RSCFDn_base> + $060B_H$ + (10_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB3[7:0]								RMDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB1[7:0]								RMDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.39 RSCANnRMDF0_q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB3[7:0]	Receive Buffer Data Byte 3
23 to 16	RMDB2[7:0]	Receive Buffer Data Byte 2
15 to 8	RMDB1[7:0]	Receive Buffer Data Byte 1
7 to 0	RMDB0[7:0]	Receive Buffer Data Byte 0 Data for a message stored in the receive buffer can be read.

When the RMDLC[3:0] value in the RSCANnRMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

23.3.5.6 RSCANnRMDF1_q — Receive Buffer Data Field 1 Register (q = 0 to 47)

Access: RSCANnRMDF1_q register can be read only in 32-bit units
RSCANnRMDF1_qL, RSCANnRMDF1_qH register can be read only in 16-bit units
RSCANnRMDF1_qLL, RSCANnRMDF1_qLH, RSCANnRMDF1_qHL, RSCANnRMDF1_qHH registers can be read only in 8-bit units

Address: RSCANnRMDF1_q: <RSCFDn_base> + $060C_H$ + (10_H × q)

RSCANnRMDF1_qL: <RSCFDn_base> + $060C_H$ + (10_H × q),
RSCANnRMDF1_qH: <RSCFDn_base> + $060E_H$ + (10_H × q)

RSCANnRMDF1_qLL: <RSCFDn_base> + $060C_H$ + (10_H × q),
RSCANnRMDF1_qLH: <RSCFDn_base> + $060D_H$ + (10_H × q),
RSCANnRMDF1_qHL: <RSCFDn_base> + $060E_H$ + (10_H × q),
RSCANnRMDF1_qHH: <RSCFDn_base> + $060F_H$ + (10_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB7[7:0]								RMDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB5[7:0]								RMDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.40 RSCANnRMDF1_q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB7[7:0]	Receive Buffer Data Byte 7
23 to 16	RMDB6[7:0]	Receive Buffer Data Byte 6
15 to 8	RMDB5[7:0]	Receive Buffer Data Byte 5
7 to 0	RMDB4[7:0]	Receive Buffer Data Byte 4 Data for a message stored in the receive buffer can be read.

When the RMDLC[3:0] value in the RSCANnRMPTRq register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

23.3.6 Details of Receive FIFO Buffer-Related Registers

23.3.6.1 RSCANnRFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7)

Access: RSCANnRFCCx register can be read/written in 32-bit units
RSCANnRFCCxL register can be read/written in 16-bit units
RSCANnRFCCxLL, RSCANnRFCCxLH registers can be read/written in 8-bit units

Address: RSCANnRFCCx: <RSCFDn_base> + $00B8_H$ + (04_H × x)

RSCANnRFCCxL: <RSCFDn_base> + $00B8_H$ + (04_H × x)

RSCANnRFCCxLL: <RSCFDn_base> + $00B8_H$ + (04_H × x),
RSCANnRFCCxLH: <RSCFDn_base> + $00B9_H$ + (04_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	—	—	—	—	—	RFIE	RFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 23.41 RSCANnRFCCx Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
15 to 13	RFIGCV[2:0]	Receive FIFO Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.
12	RFIM	Receive FIFO Interrupt Source Select 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met. 1: An interrupt occurs each time a message has been received.
11	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
10 to 8	RFDC[2:0]	Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7 to 2	Reserved	These bits are read as the value after reset. The write value should be the value after reset.

Table 23.41 RSCANnRFCCx Register Contents (2/2)

Bit Position	Bit Name	Function
1	RFIE	Receive FIFO Interrupt Enable 0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled.
0	RFE	Receive FIFO Buffer Enable 0: No receive FIFO buffer is used. 1: Receive FIFO buffers are used.

RFIGCV[2:0] Bits

These bits are used to specify the number of received messages for generating a receive FIFO interrupt request when the RFIM bit is set to 0 with a fraction for the total number of buffers (the setting of RFDC[2:0]).

When the RFDC[2:0] bits are set to 001_B (4 messages), set the RFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B. Modify these bits only in global reset mode.

RFIM Bit

This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

RFDC[2:0] Bits

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. When these bits are set to 000_B, no receive FIFO buffer should be used. Modify these bits only in global reset mode.

RFIE Bit

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit set to 0 (no receive FIFO buffer is used).

RFE Bit

Setting the RFE bit to 1 makes receive FIFO buffers available. Clearing this bit to 0 sets the RFEMP flag in the RSCANnRFSTSx register to 1 (buffer empty). Modify this bit in global operating mode or global test mode.

Set this bit to 1 with another instruction after the settings to all bits in the RSCANnRFCCx register have been done.

This bit is cleared to 0 in global reset mode.

23.3.6.2 RSCANnRFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7)

Access: RSCANnRFSTSx register can be read/written in 32-bit units
RSCANnRFSTSxL register can be read/written in 16-bit units
RSCANnRFSTSxLL register can be read/written in 8-bit units
RSCANnRFSTSxLH register is a read-only register that can be read in 8-bit units

Address: RSCANnRFSTSx: <RSCFDn_base> + $00D8_H$ + (04_H × x)

RSCANnRFSTSxL: <RSCFDn_base> + $00D8_H$ + (04_H × x)

RSCANnRFSTSxLL: <RSCFDn_base> + $00D8_H$ + (04_H × x),
RSCANnRFSTSxLH: <RSCFDn_base> + $00D9_H$ + (04_H × x)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMC[7:0]								—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.42 RSCANnRFSTSx Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
15 to 8	RFMC[7:0]	Receive FIFO Unread Message Counter The number of unread messages stored in the receive FIFO buffer is displayed.
7 to 4	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
3	RFIF	Receive FIFO Interrupt Request Flag 0: No receive FIFO interrupt request is present. 1: A receive FIFO interrupt request is present.
2	RFMLT	Receive FIFO Message Lost Flag 0: No receive FIFO message is lost. 1: A receive FIFO message is lost.
1	RFFLL	Receive FIFO Buffer Full Status Flag 0: The receive FIFO buffer is not full. 1: The receive FIFO buffer is full.
0	RFEMP	Receive FIFO Buffer Empty Status Flag 0: The receive FIFO buffer contains unread message. 1: The receive FIFO buffer contains no unread message (buffer empty).

RFMC[7:0] Flag

This flag indicates the number of unread messages in the receive FIFO buffer. This flag becomes 00_H when the RFE bit in the RSCANnRFCCx register is set to 0.
This flag is 00_H in global reset mode.

RFIF Flag

This flag is set to 1 when the receive FIFO interrupt request generation conditions set by the RFIGCV[2:0] bits and the RFIM bit in the RSCANnRFCCx register are met. This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit in global operating mode or global test mode.
To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFMLT Flag

This flag is set to 1 when an attempt is made to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.

This flag is cleared to 0 in global reset mode or by writing 0 to this flag.

Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFFLL Flag

This flag is set to 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFDC[2:0] bits in the RSCANnRFCCx register.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFDC[2:0] bits, this flag is cleared to 0. This flag is also cleared to 0 when the RFE bit in the RSCANnRFCCx register is set to 0 (no receive FIFO buffer is used) or in global reset mode.

RFEMP Flag

This flag is set to 1 when all messages in the receive FIFO buffer have been read. This flag is also set to 1 when the RFE bit in the RSCANnRFCCx register is 0 or in global reset mode.

This flag is cleared to 0 when even a single received message has been stored in the receive FIFO buffer.

NOTE

To clear the RFMLT or RFIF flag to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

23.3.6.3 RSCANnRFPCTRx — Receive FIFO Buffer Pointer Control Register (x = 0 to 7)

Access: RSCANnRFPCTRx register can only be written in 32-bit units
RSCANnRFPCTRxL register is a write-only register that can only be written in 16-bit units
RSCANnRFPCTRxLL register is a write-only register that can only be written in 8-bit units

Address: RSCANnRFPCTRx: <RSCFDn_base> + $00F8_H$ + (04_H × x)

RSCANnRFPCTRxL: <RSCFDn_base> + $00F8_H$ + (04_H × x)

RSCANnRFPCTRxLL: <RSCFDn_base> + $00F8_H$ + (04_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.43 RSCANnRFPCTRx Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be the value after reset.
7 to 0	RFPC[7:0]	Receive FIFO Pointer Control When these bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer.

RFPC[7:0] Bits

When the RFPC[7:0] bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFMC[7:0] (receive FIFO unread message counter) value in the RSCANnRFSTSx register is decremented by 1. Read the RSCANnRFIDx, RSCANnRFPTRx, RSCANnRFDF0_x, and RSCANnRFDF1_x registers to read messages in the receive FIFO buffer, and then write FF_H to the RFPC[7:0] bits.

When writing FF_H to these bits, make sure that the RFE bit in the RSCANnRFCCx register is set to 1 (receive FIFO buffers are used) and the RFEMP flag in the RSCANnRFSTSx register is 0 (the receive FIFO buffer contains unread messages).

23.3.6.4 RSCANnRFIDx — Receive FIFO Buffer Access ID Register (x = 0 to 7)

Access: RSCANnRFIDx register can be read only in 32-bit units
RSCANnRFIDxL, RSCANnRFIDxH registers can be read only in 16-bit units
RSCANnRFIDxLL, RSCANnRFIDxLH, RSCANnRFIDxHL, RSCANnRFIDxHH registers can be read only in 8-bit units

Address: RSCANnRFIDx: <RSCFDn_base> + $0E00_H$ + (10_H × x)

RSCANnRFIDxL: <RSCFDn_base> + $0E00_H$ + (10_H × x),
RSCANnRFIDxH: <RSCFDn_base> + $0E02_H$ + (10_H × x)

RSCANnRFIDxLL: <RSCFDn_base> + $0E00_H$ + (10_H × x),
RSCANnRFIDxLH: <RSCFDn_base> + $0E01_H$ + (10_H × x),
RSCANnRFIDxHL: <RSCFDn_base> + $0E02_H$ + (10_H × x),
RSCANnRFIDxHH: <RSCFDn_base> + $0E03_H$ + (10_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFIDE	RFRTR	—	RFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.44 RSCANnRFIDx Register Contents

Bit Position	Bit Name	Function
31	RFIDE	Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	RFRTR	Receive FIFO Buffer RTR 0: Data frame 1: Remote frame
29	Reserved	These bits are read as the value after reset.
28 to 0	RFID[28:0]	Receive FIFO Buffer ID Data The standard ID or extended ID of received message can be read. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

RFRTR Bit

This bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer.

RFID[28:0] Bits

These bits indicate the ID of the message stored in the receive FIFO buffer.

23.3.6.5 RSCANnRFPTRx — Receive FIFO Buffer Access Pointer Register (x = 0 to 7)

Access: RSCANnRFPTRx register can be read only in 32-bit units
RSCANnRFPTRxL, RSCANnRFPTRxH registers can be read only in 16-bit units
RSCANnRFPTRxLL, RSCANnRFPTRxLH, RSCANnRFPTRxHL, RSCANnRFPTRxHH registers can be read only in 8-bit units

Address: RSCANnRFPTRx: <RSCFDn_base> + $0E04_H$ + (10_H × x)

RSCANnRFPTRxL: <RSCFDn_base> + $0E04_H$ + (10_H × x),
RSCANnRFPTRxH: <RSCFDn_base> + $0E06_H$ + (10_H × x)

RSCANnRFPTRxLL: <RSCFDn_base> + $0E04_H$ + (10_H × x),
RSCANnRFPTRxLH: <RSCFDn_base> + $0E05_H$ + (10_H × x),
RSCANnRFPTRxHL: <RSCFDn_base> + $0E06_H$ + (10_H × x),
RSCANnRFPTRxHH: <RSCFDn_base> + $0E07_H$ + (10_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDLC[3:0]				RFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.45 RSCANnRFPTRx Register Contents

Bit Position	Bit Name	Function
31 to 28	RFDLC[3:0]	Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	RFPTR[11:0]	Receive FIFO Buffer Label Data Label information of the received message can be read.
15 to 0	RFTS[15:0]	Receive FIFO Buffer Timestamp Data Timestamp value of the received message can be read.

RFDLC[3:0] Bits

These bits contain the data length of the message stored in the receive FIFO buffer.

RFPTR[11:0] Bits

These bits contain the label information of the message stored in the receive FIFO buffer.

RFTS[15:0] Bits

These bits contain the timestamp value of the message stored in the receive FIFO buffer.

23.3.6.6 RSCANnRFDF0_x — Receive FIFO Buffer Access Data Field 0 Register (x = 0 to 7)

Access: RSCANnRFDF0_x register can be read-only in 32-bit units
RSCANnRFDF0_xL, RSCANnRFDF0_xH registers can be read only in 16-bit units
RSCANnRFDF0_xLL, RSCANnRFDF0_xLH, RSCANnRFDF0_xHL, RSCANnRFDF0_xHH registers can be read only in 8-bit units

Address: RSCANnRFDF0_x: <RSCFDn_base> + $0E08_H$ + (10_H × x)

RSCANnRFDF0_xL: <RSCFDn_base> + $0E08_H$ + (10_H × x),
RSCANnRFDF0_xH: <RSCFDn_base> + $0E0A_H$ + (10_H × x)

RSCANnRFDF0_xLL: <RSCFDn_base> + $0E08_H$ + (10_H × x),
RSCANnRFDF0_xLH: <RSCFDn_base> + $0E09_H$ + (10_H × x),
RSCANnRFDF0_xHL: <RSCFDn_base> + $0E0A_H$ + (10_H × x),
RSCANnRFDF0_xHH: <RSCFDn_base> + $0E0B_H$ + (10_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB3[7:0]								RFDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB1[7:0]								RFDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.46 RSCANnRFDF0_x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB3[7:0]	Receive FIFO Buffer Data Byte 3
23 to 16	RFDB2[7:0]	Receive FIFO Buffer Data Byte 2
15 to 8	RFDB1[7:0]	Receive FIFO Buffer Data Byte 1
7 to 0	RFDB0[7:0]	Receive FIFO Buffer Data Byte 0 Data for a message stored in the receive FIFO buffer can be read.

When the RFDLC[3:0] value in the RSCANnRFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

23.3.6.7 RSCANnRFDF1_x — Receive FIFO Buffer Access Data Field 1 Register (x = 0 to 7)

Access: RSCANnRFDF1_x register can be read only in 32-bit units
RSCANnRFDF1_xL, RSCANnRFDF1_xH registers can be read only in 16-bit units
RSCANnRFDF1_xLL, RSCANnRFDF1_xLH, RSCANnRFDF1_xHL, RSCANnRFDF1_xHH registers can be read only in 8-bit units

Address: RSCANnRFDF1_x: <RSCFDn_base> + $0E0C_H$ + (10_H × x)

RSCANnRFDF1_xL: <RSCFDn_base> + $0E0C_H$ + (10_H × x),
RSCANnRFDF1_xH: <RSCFDn_base> + $0E0E_H$ + (10_H × x)

RSCANnRFDF1_xLL: <RSCFDn_base> + $0E0C_H$ + (10_H × x),
RSCANnRFDF1_xLH: <RSCFDn_base> + $0E0D_H$ + (10_H × x),
RSCANnRFDF1_xHL: <RSCFDn_base> + $0E0E_H$ + (10_H × x),
RSCANnRFDF1_xHH: <RSCFDn_base> + $0E0F_H$ + (10_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB7[7:0]								RFDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB5[7:0]								RFDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.47 RSCANnRFDF1_x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB7[7:0]	Receive FIFO Buffer Data Byte 7
23 to 16	RFDB6[7:0]	Receive FIFO Buffer Data Byte 6
15 to 8	RFDB5[7:0]	Receive FIFO Buffer Data Byte 5
7 to 0	RFDB4[7:0]	Receive FIFO Buffer Data Byte 4 Data for a message stored in the receive FIFO buffer can be read.

When the RFDLC[3:0] value in the RSCANnRFPTRx register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

23.3.7 Details of Transmit/Receive FIFO Buffer-Related Registers

23.3.7.1 RSCANnCFCCk — Transmit/receive FIFO Buffer Configuration and Control Register k (k = 0 to 8)

Access: RSCANnCFCCk register can be read/written in 32-bit units
RSCANnCFCCkL, RSCANnCFCCkH registers can be read/written in 16-bit units
RSCANnCFCCkLL, RSCANnCFCCkLH, RSCANnCFCCkHL, RSCANnCFCCkHH registers can be read/written in 8-bit units

Address: RSCANnCFCCk: <RSCFDn_base> + 0118_H + (04_H × k)

RSCANnCFCCkL: <RSCFDn_base> + 0118_H + (04_H × k),
RSCANnCFCCkH: <RSCFDn_base> + $011A_H$ + (04_H × k)

RSCANnCFCCkLL: <RSCFDn_base> + 0118_H + (04_H × k),
RSCANnCFCCkLH: <RSCFDn_base> + 0119_H + (04_H × k),
RSCANnCFCCkHL: <RSCFDn_base> + $011A_H$ + (04_H × k),
RSCANnCFCCkHH: <RSCFDn_base> + $011B_H$ + (04_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFITT[7:0]								CFTML[3:0]				CFITR	CFITSS	CFM[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFIGCV[2:0]			CFIM	—	CFDC[2:0]			—	—	—	—	—	CFTXIE	CFRXIE	CFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 23.48 RSCANnCFCCk Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	CFITT[7:0]	Set a message transmission interval. Set Value: 00_H to FF_H
23 to 20	CFTML[3:0]	Transmit Buffer Link Configuration Set the transmit buffer number to be linked to the transmit/receive FIFO buffer.
19	CFITR	Transmit/Receive FIFO Interval Timer Resolution 0: Clock dividing pclk/2 by (ITRCP [15:0] bits) 1: Clock dividing pclk/2 by (ITRCP [15:0] bits × 10)
18	CFITSS	Transmit/Receive FIFO Interval Timer Clock Source Select 0: Interval timer clock source selected by the CFITR bit 1: Interval timer clock source is the bit time clock for the channel to which the FIFO is linked.
17, 16	CFM[1:0]	Transmit/Receive FIFO Mode Select b17 b16 0 0: Receive mode 0 1: Transmit mode 1 0: Gateway mode 1 1: Setting prohibited
15 to 13	CFIGCV[2:0]	Transmit/Receive FIFO Receive Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.

Table 23.48 RSCANnCFCCk Register Contents (2/2)

Bit Position	Bit Name	Function
12	CFIM	Transmit/Receive FIFO Interrupt Source Select 0: • Receive mode/gateway mode When the number of received messages has met the condition set by the CFIGCV[2:0] bits, a FIFO receive interrupt request is generated. • Transmit mode/gateway mode When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated. 1: • Receive mode/gateway mode A FIFO receive interrupt request is generated each time a message has been received. • Transmit mode/gateway mode A FIFO transmit interrupt request is generated each time a message has been transmitted.
11	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
10 to 8	CFDC[2:0]	Transmit/Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7 to 3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2	CFTXIE	Transmit/Receive FIFO Transmit Interrupt Enable 0: Transmit/receive FIFO transmit interrupt is disabled. 1: Transmit/receive FIFO transmit interrupt is enabled.
1	CFRXIE	Transmit/Receive FIFO Receive Interrupt Enable 0: Transmit/receive FIFO receive interrupt is disabled. 1: Transmit/receive FIFO receive interrupt is enabled.
0	CFE	Transmit/Receive FIFO Buffer Enable 0: No transmit/receive FIFO buffer is used. 1: Transmit/receive FIFO buffers are used.

CFITT[7:0] Bits

These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode).

Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITT[7:0] bits.

CFTML[3:0] Bits

These bits are used to set the number of transmit buffer on the channel which will be linked to transmit/receive FIFO buffer k when the CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode). There are three transmit/receive FIFO buffers per channel, so channel number m of FIFO buffer k is calculated as m = k/3 (integer division). The actual assigned transmit buffer number p linked to FIFO buffer k will be ((16 × m) + CFTML[3:0]) (see **Table 23.16**).

See **Table 23.14** and **Table 23.15**, as for the relationship between transmit/receive FIFO buffer k and transmit buffer p.

Setting the CFDC[2:0] bits to 001_B or more enables the setting of the CFTML[3:0] bits.

Do not link to any transmit buffer which is already allocated to a transmit queue on the same channel or to another transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFITR Bit

This bit is enabled when the CFITSS bit is 0.

When this bit is 0, the interval timer clock source is the pclk/2 clock divided by the value of the ITRCP[15:0] bits in the RSCANnGCFG register.

When this bit is 1, the interval timer clock source is the pclk/2 clock divided by (the value of the ITRCP[15:0] bits in the RSCANnGCFG register × 10).

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFITSS Bit

When this bit is 0, the clock selected by the CFITR bit is the count source of the interval timer.

When this bit is 1, the bit time clock of the channel to which the FIFO is linked is the count source of the interval timer.

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFM[1:0] Bits

These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

CFIGCV[2:0] Bits

These bits are used to specify the number of received messages for generating a transmit/receive FIFO receive interrupt request when the CFM[1:0] bits are set to 00_B (receive mode) or 10_B (gateway mode) and the CFIM bit is set to 0 with a fraction for the total number of buffers (the setting of CFDC[2:0]).

When the CFDC[2:0] bits are set to 001_B (4 messages), set the CFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B.

Modify these bits only in global reset mode.

CFIM Bit

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

CFDC[2:0] Bits

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer. When these bits are set to 000_B, do not use a transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFTXIE Bit

When this bit is set to 1 and the CFTXIF flag in the RSCANnCFSTSk register is set to 1, a transmit/receive FIFO transmit interrupt request is generated.

Modify this bit with the CFE bit set to 0 (no transmit/receive FIFO buffer is used).

CFRXIE Bit

When this bit is set to 1 and the CFRXIF flag in the RSCANnCFSTSk register is set to 1, a transmit/receive FIFO receive interrupt request is generated.

Modify this bit with the CFE bit set to 0.

CFE Bit

Setting this bit to 1 makes transmit/receive FIFO buffers available.

When this bit is set to 0 in transmit mode or gateway mode, if a message in the transmit/receive FIFO buffer is being transmitted or will be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission of that message, or upon detection of a CAN bus error, or arbitration-lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

This bit is cleared to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode or gateway mode: Channel reset mode

Modify this bit in the following mode.

- Receive mode: Global operating mode or global test mode
- Transmit mode or gateway mode: Channel communication mode or channel halt mode

After all other bits in the RSCANnCFCCk register have been set, set this bit to 1 by using another instruction.

23.3.7.2 RSCANnCFSTSk — Transmit/receive FIFO Buffer Status Register (k = 0 to 8)

Access: RSCANnCFSTSk register can be read/written in 32-bit units
RSCANnCFSTSkL register can be read/written in 16-bit units
RSCANnCFSTSkLL register can be read/written in 8-bit units
RSCANnCFSTSkLH register is a read-only register that can be read in 8-bit units.

Address: RSCANnCFSTSk: <RSCFDn_base> + 0178_H + (04_H × k)

RSCANnCFSTSkL: <RSCFDn_base> + 0178_H + (04_H × k)

RSCANnCFSTSkLL: <RSCFDn_base> + 0178_H + (04_H × k),
RSCANnCFSTSkLH: <RSCFDn_base> + 0179_H + (04_H × k)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFMC[7:0]								—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.49 RSCANnCFSTSk Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
15 to 8	CFMC[7:0]	Transmit/Receive FIFO Message Counter The number of messages stored in the transmit/receive FIFO buffer.
7 to 5	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
4	CFTXIF	Transmit/Receive FIFO Transmit Interrupt Request Flag 0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present.
3	CFRXIF	Transmit/Receive FIFO Receive Interrupt Request Flag 0: No transmit/receive FIFO receive interrupt request is present. 1: A transmit/receive FIFO receive interrupt request is present.
2	CFMLT	Transmit/Receive FIFO Message Lost Flag 0: No transmit/receive FIFO message is lost. 1: A transmit/receive FIFO message is lost.
1	CFFLL	Transmit/Receive FIFO Buffer Full Status Flag 0: The transmit/receive FIFO buffer is not full. 1: The transmit/receive FIFO buffer is full.
0	CFEMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: The transmit/receive FIFO buffer contains messages. 1: The transmit/receive FIFO buffer contains no message (buffer empty).

CFMC[7:0] Bits

The CFMC[7:0] bits indicate the following values that depend on the setting of the CFM[1:0] bits in the RSCANnCFCCk register.

- When CFM[1:0] value is 01_B (transmit mode): Number of untransmitted messages in the buffer

- When CFM[1:0] value is 00_B (receive mode): Number of unread received messages in the buffer
- When CFM[1:0] value is 10_B (gateway mode): Number of untransmitted received messages in the buffer

These bits are cleared to 0 when any of the following conditions is met.

- When CFM[1:0] value is 00_B: In global reset mode
- When CFM[1:0] value is 01_B or 10_B: In channel reset mode
- When the CFE bit in the RSCANnCFCCk register is cleared to 0.

CFTXIF Flag

The CFTXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 01_B or 10_B, and the factor selected by the CFIM bit in the RSCANnCFCCk register occurs

The CFTXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFTXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFRXIF Flag

The CFRXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B, and the factor selected by the CFIM bit in the RSCANnCFCCk register occurs

The CFRXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFRXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFMLT Flag

The CFMLT flag is set to 1 when any of the following conditions is met.

- When an attempt is made to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

The CFMLT flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFMLT flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFFLL Flag

The CFFLL flag is set to 1 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFDC[2:0] bits in the RSCANnCFCCk register.

The CFFLL flag is cleared to 0 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFDC[2:0] bits.
- When the CFE bit in the RSCANnCFCCk register is 0 (no transmit/receive FIFO buffer is used): When not in the transmit abort
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

CFEMP Flag

The CFEMP flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B: All messages have been read, or in global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: All messages have been transmitted, or in channel reset mode
- When the CFE bit is 0 (no transmit/receive FIFO buffer is used): Not in the transmit abort

The CFEMP flag is cleared to 0 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B: At least one received message has been stored in the transmit/receive FIFO buffer.
- When the CFM[1:0] bits are set to 01_B: A value of FF_H has been written to the RSCANnCFPCTRk register after data was written to the RSCANnCFIDk, RSCANnCFPTRk, RSCANnCFDF0_k, and RSCANnCFDF1_k registers.

NOTE

To clear CFTXIF, CFRXIF, or CFMLT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

23.3.7.3 RSCANnCFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register (k = 0 to 8)

Access: RSCANnCFPCTRk register can only be written in 32-bit units
RSCANnCFPCTRkL register can only be written in 16-bit units
RSCANnCFPCTRkLL register can only be written in 8-bit units

Address: RSCANnCFPCTRk: <RSCFDn_base> + $01D8_H$ + (04_H × k)

RSCANnCFPCTRkL: <RSCFDn_base> + $01D8_H$ + (04_H × k)

RSCANnCFPCTRkLL: <RSCFDn_base> + $01D8_H$ + (04_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.50 RSCANnCFPCTRk Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be the value after reset.
7 to 0	CFPC[7:0]	Transmit/Receive FIFO Pointer Control • Receive mode: Writing FF_H to these bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. • Transmit mode: Writing FF_H to these bits moves the write pointer to the next stage of the transmit/receive FIFO buffer. • Gateway mode: Setting prohibited

CFPC[7:0] Bits

- Receive mode (CFM[1:0] value in the RSCANnCFCCk register is 00_B):
  Writing FF_H to the CFPC[7:0] bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value (transmit/receive FIFO message counter) in the RSCANnCFSTSk register is decremented by 1. Read the RSCANnCFIDk, RSCANnCFPTRk, RSCANnCFDF0_k, and RSCANnCFDF1_k registers to read messages from the transmit/receive FIFO buffer, and then write FF_H to the CFPC[7:0] bits.
  When writing FF_H to these bits, make sure that the CFE bit in the RSCANnCFCCk register is set to 1 (transmit/receive FIFO buffers are used) and the CFEMP flag in the RSCANnCFSTSk register is 0 (the transmit/receive FIFO buffer contains messages).
- Transmit mode (CFM[1:0] value in the RSCANnCFCCk register is 01_B):
  Writing FF_H to the CFPC[7:0] bits stores the data written to the RSCANnCFIDk, RSCANnCFPTRk, RSCANnCFDF0_k, and RSCANnCFDF1_k registers in the transmit/receive FIFO buffer and moves the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value is incremented by 1 Write transmit messages to the RSCANnCFIDk, RSCANnCFPTRk, RSCANnCFDF0_k, and RSCANnCFDF1_k registers

before writing FF_H to the CFPC[7:0] bits.
When writing FF_H to these bits, make sure that the CFE bit in the RSCANnCFCCk register is set to 1 and the CFFLL flag in the RSCANnCFSTSk register is 0 (the transmit/receive FIFO buffer is not full).

- Gateway mode (CFM[1:0] value in the RSCANnCFCCk register is 10_B):
  Setting prohibited

23.3.7.4 RSCANnCFIDk — Transmit/receive FIFO Buffer Access ID Register (k = 0 to 8)

Access: RSCANnCFIDk register can be read/written in 32-bit units
RSCANnCFIDkL, RSCANnCFIDkH registers can be read/written in 16-bit units
RSCANnCFIDkLL, RSCANnCFIDkLH, RSCANnCFIDkHL, RSCANnCFIDkHH registers can be read/written in 8-bit units

Address: RSCANnCFIDk: <RSCFDn_base> + $0E80_H$ + (10_H × k)

RSCANnCFIDkL: <RSCFDn_base> + $0E80_H$ + (10_H × k),
RSCANnCFIDkH: <RSCFDn_base> + $0E82_H$ + (10_H × k)

RSCANnCFIDkLL: <RSCFDn_base> + $0E80_H$ + (10_H × k),
RSCANnCFIDkLH: <RSCFDn_base> + $0E81_H$ + (10_H × k),
RSCANnCFIDkHL: <RSCFDn_base> + $0E82_H$ + (10_H × k),
RSCANnCFIDkHH: <RSCFDn_base> + $0E83_H$ + (10_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFIDE	CFRTR	THLEN	CFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.51 RSCANnCFIDk Register Contents

Bit Position	Bit Name	Function
31	CFIDE	Transmit/Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	CFRTR	Transmit/Receive FIFO Buffer RTR 0: Data frame 1: Remote frame
29	THLEN	Transmit History Data Store Enable This bit is valid only when the CFM[1:0] value is 01_B (transmit mode). 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	CFID[28:0]	Transmit/Receive FIFO Buffer ID Data • When CFM[1:0] value is 01_B (transmit mode): Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11. • When CFM[1:0] value is 00_B (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, read bits 10 to 0. Bits 28 to 11 are read as 0.

This register is writable only when the CFM[1:0] value in the RSCANnCFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). This RSCANnCFIDk register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, these bits are used to set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

CFRTR Bit

This bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, this bit is used to set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, and buffer type) of transmit messages is stored in the transmit history buffer after transmission is completed.

This bit is enabled when the CFM[1:0] value is 01_B (transmit mode).

CFID[28:0] Bits

These bits contain the ID of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

When the CFM[1:0] value is 01_B, this bit is used to set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

23.3.7.5 RSCANnCFPTRk — Transmit/receive FIFO Buffer Access Pointer Register (k = 0 to 8)

Access: RSCANnCFPTRk register can be read/written in 32-bit units
RSCANnCFPTRkL, RSCANnCFPTRkH registers can be read/written in 16-bit units
RSCANnCFPTRkLL, RSCANnCFPTRkLH, RSCANnCFPTRkHL, RSCANnCFPTRkHH registers can be read/written in 8-bit units

Address: RSCANnCFPTRk: <RSCFDn_base> + $0E84_H$ + (10_H × k)

RSCANnCFPTRkL: <RSCFDn_base> + $0E84_H$ + (10_H × k),
RSCANnCFPTRkH: <RSCFDn_base> + $0E86_H$ + (10_H × k)

RSCANnCFPTRkLL: <RSCFDn_base> + $0E84_H$ + (10_H × k),
RSCANnCFPTRkLH: <RSCFDn_base> + $0E85_H$ + (10_H × k),
RSCANnCFPTRkHL: <RSCFDn_base> + $0E86_H$ + (10_H × k),
RSCANnCFPTRkHH: <RSCFDn_base> + $0E87_H$ + (10_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDLC[3:0]				CFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.52 RSCANnCFPTRk Register Contents

Bit Position	Bit Name	Function
31 to 28	CFDLC[3:0]	Transmit/Receive FIFO Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 16	CFPTR[11:0]	Transmit/Receive FIFO Buffer Label Data • When CFM[1:0] value is 01_B (transmit mode): Set the label information to be stored in the transmit history buffer. Only bits CFPTR[7:0] are valid. • When CFM[1:0] value is 00_B (receive mode): The label information of the received message can be read.
15 to 0	CFTS[15:0]	Transmit/Receive FIFO Buffer Timestamp Data These bits are valid only when the CFM[1:0] value is 00_B (receive mode). The timestamp value of the received message can be read.

This register is writable only when the CFM[1:0] value in the RSCANnCFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFDLC[3:0] Bits

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, these bits are used to set the data length of the message to be transmitted from the transmit/receive FIFO buffer. If the data length is set to 1001_B or more, the actual transmit data defaults to 8 bytes.

CFPTR[11:0] Bits

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, the CFPTR[7:0] value is stored in the transmit history buffer when message transmission has been completed.

CFTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.

These bits are valid when the CFM[1:0] value is 00_B.

23.3.7.6 RSCANnCFDF0_k — Transmit/receive FIFO Buffer Access Data Field 0 Register (k = 0 to 8)

Access: RSCANnCFDF0_k register can be read/written in 32-bit units
RSCANnCFDF0_kL, RSCANnCFDF0_kH registers can be read/written in 16-bit units
RSCANnCFDF0_kLL, RSCANnCFDF0_kLH, RSCANnCFDF0_kHL, RSCANnCFDF0_kHH registers can be read/written in 8-bit units

Address: RSCANnCFDF0_k: <RSCFDn_base> + $0E88_H$ + (10_H × k)

RSCANnCFDF0_kL: <RSCFDn_base> + $0E88_H$ + (10_H × k),
RSCANnCFDF0_kH: <RSCFDn_base> + $0E8A_H$ + (10_H × k)

RSCANnCFDF0_kLL: <RSCFDn_base> + $0E88_H$ + (10_H × k),
RSCANnCFDF0_kLH: <RSCFDn_base> + $0E89_H$ + (10_H × k),
RSCANnCFDF0_kHL: <RSCFDn_base> + $0E8A_H$ + (10_H × k),
RSCANnCFDF0_kHH: <RSCFDn_base> + $0E8B_H$ + (10_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB3[7:0]								CFDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB1[7:0]								CFDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.53 RSCANnCFDF0_k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB3[7:0]	Transmit/Receive FIFO Buffer Data Byte 3
23 to 16	CFDB2[7:0]	Transmit/Receive FIFO Buffer Data Byte 2
15 to8	CFDB1[7:0]	Transmit/Receive FIFO Buffer Data Byte 1
7 to 0	CFDB0[7:0]	Transmit/Receive FIFO Buffer Data Byte 0 • When CFM[1:0] value is 01_B (transmit mode): Set the transmit/receive FIFO buffer data. • When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.

This register is writable only when the CFM[1:0] value in the RSCANnCFCCk register is 01_B (transmit mode).

This register is readable only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] value in the RSCANnCFPTRk register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

23.3.7.7 RSCANnCFDF1_k — Transmit/receive FIFO Buffer Access Data Field 1 Register (k = 0 to 8)

Access: RSCANnCFDF1_k register can be read/written in 32-bit units
RSCANnCFDF1_kL, RSCANnCFDF1_kH registers can be read/written in 16-bit units
RSCANnCFDF1_kLL, RSCANnCFDF1_kLH, RSCANnCFDF1_kHL, RSCANnCFDF1_kHH registers can be read/written in 8-bit units

Address: RSCANnCFDF1_k: <RSCFDn_base> + $0E8C_H$ + (10_H × k)

RSCANnCFDF1_kL: <RSCFDn_base> + $0E8C_H$ + (10_H × k),
RSCANnCFDF1_kH: <RSCFDn_base> + $0E8E_H$ + (10_H × k)

RSCANnCFDF1_kLL: <RSCFDn_base> + $0E8C_H$ + (10_H × k),
RSCANnCFDF1_kLH: <RSCFDn_base> + $0E8D_H$ + (10_H × k),
RSCANnCFDF1_kHL: <RSCFDn_base> + $0E8E_H$ + (10_H × k),
RSCANnCFDF1_kHH: <RSCFDn_base> + $0E8F_H$ + (10_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB7[7:0]								CFDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB5[7:0]								CFDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.54 RSCANnCFDF1_k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB7[7:0]	Transmit/Receive FIFO Buffer Data Byte 7
23 to 16	CFDB6[7:0]	Transmit/Receive FIFO Buffer Data Byte 6
15 to 8	CFDB5[7:0]	Transmit/Receive FIFO Buffer Data Byte 5
7 to 0	CFDB4[7:0]	Transmit/Receive FIFO Buffer Data Byte 4 • When CFM[1:0] value is 01_B (transmit mode): Set the transmit/receive FIFO buffer data. • When CFM[1:0] value is 00_B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.

This register is writable only when the CFM[1:0] value in the RSCANnCFCCk register is 01_B (transmit mode).

This register is readable only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] value in the RSCANnCFPTRk register is smaller than 1000_B, data bytes for which no data is set are read as 00_H.

This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

23.3.8 Details of FIFO Status-Related Registers

23.3.8.1 RSCANnFESTS — FIFO Empty Status Register

Access: RSCANnFESTSregisters can be read only in 32-bit units
RSCANnFESTSL, RSCANnFESTSH registers can be read only in 16-bit units
RSCANnFESTSLL, RSCANnFESTSLH, RSCANnFESTSHL registers can be read only in 8-bit units

Address: RSCANnFESTS: <RSCFDn_base> + 0238_H

RSCANnFESTSL: <RSCFDn_base> + 0238_H,
RSCANnFESTSH: <RSCFDn_base> + $023A_H$

RSCANnFESTSLL: <RSCFDn_base> + 0238_H,
RSCANnFESTSLH: <RSCFDn_base> + 0239_H,
RSCANnFESTSHL: <RSCFDn_base> + $023A_H$

Value after reset: 03FF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF8EMP
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7EMP	CF6EMP	CF5EMP	CF4EMP	CF3EMP	CF2EMP	CF1EMP	CF0EMP	RF7EMP	RF6EMP	RF5EMP	RF4EMP	RF3EMP	RF2EMP	RF1EMP	RF0EMP
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.55 RSCANnFESTS Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	These bits are read as the value after reset.
16	CF8EMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: Transmit/receive FIFO buffer k contains a message. 1: Transmit/receive FIFO buffer k contains no message. (k = 0 to 8)
15	CF7EMP	
14	CF6EMP	
13	CF5EMP	
12	CF4EMP	
11	CF3EMP	
10	CF2EMP	
9	CF1EMP	
8	CF0EMP	
7	RF7EMP	Receive FIFO Buffer Empty Status Flag 0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message (buffer empty). (x = 0 to 7)
6	RF6EMP	
5	RF5EMP	
4	RF4EMP	
3	RF3EMP	
2	RF2EMP	
1	RF1EMP	
0	RF0EMP	

The RSCANnFESTS register is set to 03FF $FFFF_H$ in global reset mode.

CFkEMP Flag (k = 0 to 8)

The CFkEMP flag is set to 1 when the CFEMP flag in the RSCANnCFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)). When the CFEMP flag is cleared to 0 (the transmit/receive FIFO buffer contains messages), the CFkEMP flag is cleared to 0.

RFxEMP Flag (x = 0 to 7)

The RFxEMP flag is set to 1 when the RFEMP flag in the RSCANnRFSTSx register is set to 1 (the receive FIFO buffer contains no unread message). When the RFEMP flag is cleared to 0 (the receive FIFO buffer contains unread messages), the RFxEMP flag is cleared to 0.

23.3.8.2 RSCANnFFSTS — FIFO Full Status Register

Access: RSCANnFFSTS register can be read only in 32-bit units
RSCANnFFSTSL, RSCANnFFSTSH registers can be read only in 16-bit units
RSCANnFFSTSLL, RSCANnFFSTSLH, RSCANnFFSTSHL registers can be read only in 8-bit units

Address: RSCANnFFSTS: <RSCFDn_base> + $023C_H$

RSCANnFFSTSL: <RSCFDn_base> + $023C_H$,
RSCANnFFSTSH: <RSCFDn_base> + $023E_H$

RSCANnFFSTSLL: <RSCFDn_base> + $023C_H$,
RSCANnFFSTSLH: <RSCFDn_base> + $023D_H$,
RSCANnFFSTSHL: <RSCFDn_base> + $023E_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF8FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7FLL	CF6FLL	CF5FLL	CF4FLL	CF3FLL	CF2FLL	CF1FLL	CF0FLL	RF7FLL	RF6FLL	RF5FLL	RF4FLL	RF3FLL	RF2FLL	RF1FLL	RF0FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.56 RSCANnFFSTS Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	These bits are read as the value after reset.
16	CF8FLL	Transmit/Receive FIFO Buffer Full Status Flag 0: Transmit/receive buffer k is not full. 1: Transmit/receive buffer k is full. (k = 0 to 8)
15	CF7FLL	
14	CF6FLL	
13	CF5FLL	
12	CF4FLL	
11	CF3FLL	
10	CF2FLL	
9	CF1FLL	
8	CF0FLL	
7	RF7FLL	Receive FIFO Buffer Full Status Flag 0: Receive FIFO buffer x is not full. 1: Receive FIFO buffer x is full. (x = 0 to 7)
6	RF6FLL	
5	RF5FLL	
4	RF4FLL	
3	RF3FLL	
2	RF2FLL	
1	RF1FLL	
0	RF0FLL	

The RSCANnFFSTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkFLL Flag (k = 0 to 8)

The CFkFLL flag is set to 1 when the CFFLL flag in the RSCANnCFSTSk register is set to 1 (the transmit/receive FIFO buffer is full).

When the CFFLL flag is cleared to 0 (the transmit/receive FIFO buffer is not full), the CFkFLL flag is cleared to 0.

RFxFLL Flag (x = 0 to 7)

The RFxFLL flag is set to 1 when the RFFLL flag in the RSCANnRFSTSx register is set to 1 (the receive FIFO buffer is full). When the RFFLL flag is cleared to 0 (the receive FIFO buffer is not full), the RFxFLL flag is cleared to 0.

23.3.8.3 RSCANnFMSTS — FIFO Message Lost Status Register

Access: RSCANnFMSTS register can be read only in 32-bit units
RSCANnFMSTSL, RSCANnFMSTSH registers can be read only in 16-bit units
RSCANnFMSTSLL, RSCANnFMSTSLH, RSCANnFMSTSHL registers can be read only in 8-bit units

Address: RSCANnFMSTS: <RSCFDn_base> + 0240_H

RSCANnFMSTSL: <RSCFDn_base> + 0240_H,
RSCANnFMSTSH: <RSCFDn_base> + 0242_H

RSCANnFMSTSLL: <RSCFDn_base> + 0240_H,
RSCANnFMSTSLH: <RSCFDn_base> + 0241_H,
RSCANnFMSTSHL: <RSCFDn_base> + 0242_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF8MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7MLT	CF6MLT	CF5MLT	CF4MLT	CF3MLT	CF2MLT	CF1MLT	CF0MLT	RF7MLT	RF6MLT	RF5MLT	RF4MLT	RF3MLT	RF2MLT	RF1MLT	RF0MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.57 RSCANnFMSTS Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	These bits are read as the value after reset.
16	CF8MLT	Transmit/Receive FIFO Buffer Message Lost Status Flag 0: No transmit/receive FIFO buffer k message is lost. 1: A transmit/receive FIFO buffer k message is lost. (k = 0 to 8)
15	CF7MLT	
14	CF6MLT	
13	CF5MLT	
12	CF4MLT	
11	CF3MLT	
10	CF2MLT	
9	CF1MLT	
8	CF0MLT	
7	RF7MLT	Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost. (x = 0 to 7)
6	RF6MLT	
5	RF5MLT	
4	RF4MLT	
3	RF3MLT	
2	RF2MLT	
1	RF1MLT	
0	RF0MLT	

The RSCANnFMSTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkMLT Flag (k = 0 to 8)

The CFkMLT flag is set to 1 when the CFMLT flag in the RSCANnCFSTSk register is set to 1 (a transmit/receive FIFO message is lost). When the CFMLT flag is cleared to 0, the CFkMLT flag is cleared to 0.

RFxMLT Flag (x = 0 to 7)

The RFxMLT flag is set to 1 when the RFMLT flag in the RSCANnRFSTSx register is set to 1 (a receive FIFO message is lost). When the RFMLT flag is cleared to 0, the RFxMLT flag is cleared to 0.

23.3.8.4 RSCANnRFISTS — Receive FIFO Buffer Interrupt Flag Status Register

Access: RSCANnRFISTS register can be read only in 32-bit units
RSCANnRFISTSL register is a read-only register that can be read only in 16-bit units
RSCANnRFISTSLL register is a read-only register that can be read only in 8-bit units

Address: RSCANnRFISTS: <RSCFDn_base> + 0244_H

RSCANnRFISTSL: <RSCFDn_base> + 0244_H

RSCANnRFISTSLL: <RSCFDn_base> + 0244_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RF7IF	RF6IF	RF5IF	RF4IF	RF3IF	RF2IF	RF1IF	RF0IF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.58 RSCANnRFISTS Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset.
7	RF7IF	Receive FIFO Buffer Interrupt Request Status Flag 0: No receive FIFO buffer x interrupt request is present. 1: A receive FIFO buffer x interrupt request is present. (x = 0 to 7)
6	RF6IF	
5	RF5IF	
4	RF4IF	
3	RF3IF	
2	RF2IF	
1	RF1IF	
0	RF0IF	

The RSCANnRFISTS register is cleared to $0000\ 0000_H$ in global reset mode.

RFxIF Flag (x = 0 to 7)

The RFxIF flag is set to 1 when the RFIF flag in the RSCANnRFSTSx register is set to 1 (a receive FIFO interrupt request is present). When the RFIF flag is cleared to 0, the RFxIF flag is cleared to 0.

23.3.8.5 RSCANnCFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register

Access: RSCANnCFRISTS register can be read only in 32-bit units
RSCANnCFRISTSL register is a read-only register that can be read only in 16-bit units
RSCANnCFRISTSLL, RSCANnCFRISTSLH are the read-only registers that can be read only in 8-bit units

Address: RSCANnCFRISTS: <RSCFDn_base> + 0248_H

RSCANnCFRISTSL: <RSCFDn_base> + 0248_H

RSCANnCFRISTSLL: <RSCFDn_base> + 0248_H,
RSCANnCFRISTSLH: <RSCFDn_base> + 0249_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CF8RXIF	CF7RXIF	CF6RXIF	CF5RXIF	CF4RXIF	CF3RXIF	CF2RXIF	CF1RXIF	CF0RXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.59 RSCANnCFRISTS Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are read as the value after reset.
8	CF8RXIF	Transmit/Receive FIFO Buffer Receive Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k receive interrupt request is present. 1: A transmit/receive FIFO buffer k receive interrupt request is present. (k = 0 to 8)
7	CF7RXIF	
6	CF6RXIF	
5	CF5RXIF	
4	CF4RXIF	
3	CF3RXIF	
2	CF2RXIF	
1	CF1RXIF	
0	CF0RXIF	

The RSCANnCFRISTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkRXIF Flag (k = 0 to 8)

The CFkRXIF flag is set to 1 when the CFRXIF flag in the RSCANnCFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). When the CFRXIF flag is cleared to 0, the CFkRXIF flag is cleared to 0.

23.3.8.6 RSCANnCFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register

Access: RSCANnCFTISTS register can be read only in 32-bit units
RSCANnCFTISTSL register is a read-only register that can be read only in 16-bit units
RSCANnCFTISTSLL, RSCANnCFTISTSLH are the read-only registers that can be read only in 8-bit units

Address: RSCANnCFTISTS: <RSCFDn_base> + $024C_H$

RSCANnCFTISTSL: <RSCFDn_base> + $024C_H$

RSCANnCFTISTSLL: <RSCFDn_base> + $024C_H$,
RSCANnCFTISTSLH: <RSCFDn_base> + $024D_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CF8TXIF	CF7TXIF	CF6TXIF	CF5TXIF	CF4TXIF	CF3TXIF	CF2TXIF	CF1TXIF	CF0TXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.60 RSCANnCFTISTS Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are read as the value after reset.
8	CF8TXIF	Transmit/Receive FIFO Buffer Transmit Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k transmit interrupt request is present. 1: A transmit/receive FIFO buffer k transmit interrupt request is present. (k = 0 to 8)
7	CF7TXIF	
6	CF6TXIF	
5	CF5TXIF	
4	CF4TXIF	
3	CF3TXIF	
2	CF2TXIF	
1	CF1TXIF	
0	CF0TXIF	

The RSCANnCFTISTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkTXIF Flag (k = 0 to 8)

The CFkTXIF flag is set to 1 when the CFTXIF flag in the RSCANnCFSTSk register is set to 1 (a transmit/receive FIFO transmit interrupt request is present). When the CFTXIF flag is cleared to 0, the CFkTXIF flag is cleared to 0.

23.3.9 Details of Transmit Buffer-Related Registers

23.3.9.1 RSCANnTMCp — Transmit Buffer Control Register (p = 0 to 47)

Access: RSCANnTMCp register can be read/written in 8-bit units

Address: RSCANnTMCp: <RSCFDn_base> + 0250_H + (01_H × p)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W*1	R/W*1

Note 1. The only effective value for writing to this bit is 1, which sets the bit. Otherwise writing to the bit results in retention of its state.

Table 23.61 RSCANnTMCp Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2	TMOM	One-Shot Transmission Enable 0: One-shot transmission is disabled. 1: One-shot transmission is enabled.
1	TMTAR	Transmit Abort Request 0: Transmit abort is not requested. 1: Transmit abort is requested.
0	TMTR	Transmit Request 0: Transmission is not requested. 1: Transmission is requested.

When the RSCANnTMCp register meets any of the following conditions, set it to 00_H.

- The RSCANnTMCp register corresponds to the transmit buffer number selected by the CFTML[3:0] bits in the RSCANnCFCCk register (p = m × 16 + the value of CFTML[3:0] bits).
- The RSCANnTMCp register corresponds to the transmit buffer allocated to the transmit queue by the TXQDC[3:0] bits in the RSCANnTXQCCm (m = 0 to 2) register (p = (m × 16 + 15) to (m × 16 + 15 – the value of TXQDC[3:0] bits)).

Bits in the RSCANnTMCp register are all cleared to 0 in channel reset mode. Modify the RSCANnTMCp register in channel communication mode or channel halt mode.

TMOM Bit

Setting this bit to 1 enables one-shot transmission. When transmission fails, retransmission defined in the CAN protocol is not performed.

Modify the TMOM bit when the TMTRM flag in the RSCANnTMSTSp register is set to 0. Set the TMOM bit to 1 together with the TMTR bit.

TMTAR Bit

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or one that will be transmitted next cannot be aborted.

The TMTAR bit can be set to 1 when TMTR bit is 1.

The TMTAR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration-lost has been detected.

If this bit becomes 0 at the same time as the program writes 1 to this bit, this bit becomes 0.

TMTR Bit

Setting this bit to 1 transmits the message stored in the transmit buffer.

The TMTR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed after the TMTAR bit was set to 1.
- An error or arbitration-lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the value of TMTRF[1:0] in the RSCANnTMSTSp register is 00_B.

23.3.9.2 RSCANnTMSTSp — Transmit Buffer Status Register (p = 0 to 47)

Access: RSCANnTMSTSp register can be read/written in 8-bit units

Address: RSCANnTMSTSp: <RSCFDn_base> + $02D0_H$ + (01_H × p)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R

Table 23.62 RSCANnTMSTSp Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
4	TMTARM	Transmit Buffer Transmit Abort Request Status Flag 0: No transmit abort request is present. 1: A transmit abort request is present.
3	TMTRM	Transmit Buffer Transmit Request Status Flag 0: No transmit request is present. 1: A transmit request is present.
2, 1	TMTRF[1:0]	Transmit Buffer Transmit Result Status Flag b2 b1 0 0: Transmission is in progress or no transmit request is present. 0 1: Transmit abort has been completed. 1 0: Transmission has been completed (without transmit abort request). 1 1: Transmission has been completed (with transmit abort request).
0	TMTSTS	Transmit Buffer Transmit Status Flag 0: Transmission is not in progress. 1: Transmission is in progress.

The RSCANnTMSTSp register is cleared to all 0 in channel reset mode.

TMTARM Flag

The TMTARM flag is set to 1 when the TMTAR bit in the RSCANnTMCp register is set to 1.

The TMTARM flag is set to 0 when the TMTAR bit in the RSCANnTMCp register is set to 0.

TMTRM Flag

The TMTRM flag is set to 1 when the TMTR bit in the RSCANnTMCp register is set to 1.

The TMTRM flag is set to 0 when the TMTR bit in the RSCANnTMCp register is set to 0.

TMTRF[1:0] Flag

This flag indicates the result of transmission from the transmit buffer.

00_B: Transmission is in progress or no transmit request is present.

01_B: Transmission from the transmit buffer was aborted.

10_B: Transmission has been completed with the TMTAR bit in the RSCANnTMCp register set to 0 (transmit abort is not requested).

11_B: Transmission has been completed with the TMTAR bit in the RSCANnTMCp register set to 1 (transmit abort is requested).

Write 00_B to the TMTRF[1:0] flag in channel communication mode or channel halt mode. Do not write any value other than 00_B to this flag.

TMTSTS Flag

This flag is set to 1 when transmission from the transmit buffer starts, and is cleared to 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

23.3.9.3 RSCANnTMIDp — Transmit Buffer ID Register (p = 0 to 47)

Access: RSCANnTMIDp register can be read/written in 32-bit units
RSCANnTMIDpL, RSCANnTMIDpH registers can be read/written in 16-bit units
RSCANnTMIDpLL, RSCANnTMIDpLH, RSCANnTMIDpHL, RSCANnTMIDpHH registers can be read/written in 8-bit units

Address: RSCANnTMIDp: <RSCFDn_base> + 1000_H + (10_H × p)

RSCANnTMIDpL: <RSCFDn_base> + 1000_H + (10_H × p),
RSCANnTMIDpH: <RSCFDn_base> + 1002_H + (10_H × p)

RSCANnTMIDpLL: <RSCFDn_base> + 1000_H + (10_H × p),
RSCANnTMIDpLH: <RSCFDn_base> + 1001_H + (10_H × p),
RSCANnTMIDpHL: <RSCFDn_base> + 1002_H + (10_H × p),
RSCANnTMIDpHH: <RSCFDn_base> + 1003_H + (10_H × p)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIDE	TMRTR	THLEN	TMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.63 RSCANnTMIDp Register Contents

Bit Position	Bit Name	Function
31	TMIDE	Transmit Buffer IDE 0: Standard ID 1: Extended ID
30	TMRTR	Transmit Buffer RTR 0: Data frame 1: Remote frame
29	THLEN	Transmit History Data Store Enable 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	TMID[28:0]	Transmit Buffer ID Data Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11.

Modify this register when the TMTRM bit in the corresponding RSCANnTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write data to this register. If this register is allocated to the transmit queue, only write data to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMIDE Bit

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

TMRTR Bit

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, buffer type, and timestamp (timestamp is included if the TMTSCE bit in the RSCANnGCFG register is 1)) of transmit messages is stored in the transmit history buffer after transmission is completed.

TMID[28:0] Bits

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

23.3.9.4 RSCANnTMPTRp — Transmit Buffer Pointer Register (p = 0 to 47)

Access: RSCANnTMPTRp register can be read/written in 32-bit units
RSCANnTMPTRpH register can be read/written in 16-bit units
RSCANnTMPTRpHL, RSCANnTMPTRpHH registers can be read/written in 8-bit units

Address: RSCANnTMPTRp: <RSCFDn_base> + 1004_H + (10_H × p)

RSCANnTMPTRpH: <RSCFDn_base> + 1006_H + (10_H × p)

RSCANnTMPTRpHL: <RSCFDn_base> + 1006_H + (10_H × p),
RSCANnTMPTRpHH: <RSCFDn_base> + 1007_H + (10_H × p)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDLC[3:0]				—	—	—	—	TMPTR[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.64 RSCANnTMPTRp Register Contents

Bit Position	Bit Name	Function
31 to 28	TMDLC[3:0]	Transmit Buffer DLC Data b31 b30 b29 b28 0 0 0 0: 0 data bytes 0 0 0 1: 1 data byte 0 0 1 0: 2 data bytes 0 0 1 1: 3 data bytes 0 1 0 0: 4 data bytes 0 1 0 1: 5 data bytes 0 1 1 0: 6 data bytes 0 1 1 1: 7 data bytes 1 X X X: 8 data bytes
27 to 24	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
23 to 16	TMPTR[7:0]	Transmit Buffer Label Data Set the label information to be stored in the transmit history buffer.
15 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

Modify this register when the TMTRM bit in the corresponding RSCANnTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMDLC[3:0] Bits

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMRTR bit in the RSCANnTMIDp register is set to 0 (data frame). If the data length is set to 1001_B or more, the transmit data is 8 bytes long.

When the TMRTR bit is set to 1 (remote frame), set the data length of messages to be requested.

TMPTR[7:0] Bits

When message transmission has been completed, the TMPTR[7:0] value is stored in the transmit history buffer.

23.3.9.5 RSCANnTMDF0_p — Transmit Buffer Data Field 0 Register (p = 0 to 47)

Access: RSCANnTMDF0_p register can be read/written in 32-bit units
RSCANnTMDF0_pL, RSCANnTMDF0_pH registers can be read/written in 16-bit units
RSCANnTMDF0_pLL, RSCANnTMDF0_pLH, RSCANnTMDF0_pHL, RSCANnTMDF0_pHH registers can be read/written in 8-bit units

Address: RSCANnTMDF0_p: <RSCFDn_base> + 1008_H + (10_H × p)

RSCANnTMDF0_pL: <RSCFDn_base> + 1008_H + (10_H × p),
RSCANnTMDF0_pH: <RSCFDn_base> + $100A_H$ + (10_H × p)

RSCANnTMDF0_pLL: <RSCFDn_base> + 1008_H + (10_H × p),
RSCANnTMDF0_pLH: <RSCFDn_base> + 1009_H + (10_H × p),
RSCANnTMDF0_pHL: <RSCFDn_base> + $100A_H$ + (10_H × p),
RSCANnTMDF0_pHH: <RSCFDn_base> + $100B_H$ + (10_H × p)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB3[7:0]								TMDB2[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB1[7:0]								TMDB0[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.65 RSCANnTMDF0_p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB3[7:0]	Transmit Buffer Data Byte 3
23 to 16	TMDB2[7:0]	Transmit Buffer Data Byte 2
15 to 8	TMDB1[7:0]	Transmit Buffer Data Byte 1
7 to 0	TMDB0[7:0]	Transmit Buffer Data Byte 0 Set the transmit buffer data.

Modify this register when the TMTRM bit in the corresponding RSCANnTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

23.3.9.6 RSCANnTMDF1_p — Transmit Buffer Data Field 1 Register (p = 0 to 47)

Access: RSCANnTMDF1_p register can be read/written in 32-bit units
RSCANnTMDF1_pL, RSCANnTMDF1_pH registers can be read/written in 16-bit units
RSCANnTMDF1_pLL, RSCANnTMDF1_pLH, RSCANnTMDF1_pHL, RSCANnTMDF1_pHH registers can be read/written in 8-bit units

Address: RSCANnTMDF1_p: <RSCFDn_base> + $100C_H$ + (10_H × p)

RSCANnTMDF1_pL: <RSCFDn_base> + $100C_H$ + (10_H × p),
RSCANnTMDF1_pH: <RSCFDn_base> + $100E_H$ + (10_H × p)

RSCANnTMDF1_pLL: <RSCFDn_base> + $100C_H$ + (10_H × p),
RSCANnTMDF1_pLH: <RSCFDn_base> + $100D_H$ + (10_H × p),
RSCANnTMDF1_pHL: <RSCFDn_base> + $100E_H$ + (10_H × p),
RSCANnTMDF1_pHH: <RSCFDn_base> + $100F_H$ + (10_H × p)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB7[7:0]								TMDB6[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB5[7:0]								TMDB4[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.66 RSCANnTMDF1_p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB7[7:0]	Transmit Buffer Data Byte 7
23 to 16	TMDB6[7:0]	Transmit Buffer Data Byte 6
15 to 8	TMDB5[7:0]	Transmit Buffer Data Byte 5
7 to 0	TMDB4[7:0]	Transmit Buffer Data Byte 4 Set the transmit buffer data.

Modify this register when the TMTRM bit in the corresponding RSCANnTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

23.3.9.7 RSCANnTMIECy — Transmit Buffer Interrupt Enable Configuration Register (y = 0, 1)

Access: RSCANnTMIECy register can be read/written in 32-bit units
RSCANnTMIECyL, RSCANnTMIECyH registers can be read/written in 16-bit units
RSCANnTMIECyLL, RSCANnTMIECyLH, RSCANnTMIECyHL, RSCANnTMIECyHH registers can be read/written in 8-bit units

Address: RSCANnTMIECy: <RSCFDn_base> + 0390_H + (04_H × y)

RSCANnTMIECyL: <RSCFDn_base> + 0390_H + (04_H × y),
RSCANnTMIECyH: <RSCFDn_base> + 0392_H + (04_H × y)

RSCANnTMIECyLL: <RSCFDn_base> + 0390_H + (04_H × y),
RSCANnTMIECyLH: <RSCFDn_base> + 0391_H + (04_H × y),
RSCANnTMIECyHL: <RSCFDn_base> + 0392_H + (04_H × y),
RSCANnTMIECyHH: <RSCFDn_base> + 0393_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIEp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMIEp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.67 RSCANnTMIECy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMIEp	Transmit Buffer Interrupt Enable p (p = y × 32 + 31 to y × 32 + 16) 0: Transmit buffer interrupt is disabled 1: Transmit buffer interrupt is enabled
15 to 0	TMIEp	Transmit Buffer Interrupt Enable p (p = y × 32 + 15 to y × 32 + 0) 0: Transmit buffer interrupt is disabled. 1: Transmit buffer interrupt is enabled.

TMIEp Bits (p = 0 to 47)

When any of these bits is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify these bits when the TMTRM flag in the corresponding RSCANnTMSTSp register is 0 (no transmit request is present).

Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers or transmit buffers allocated to the transmit queue.

Table 23.68 shows the bit assignment.

Table 23.68 TMIEp Bit Assignment (1/2)

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
.	.	.
.	.	.

Table 23.68 TMIEp Bit Assignment (2/2)

Bit	Channel	Transmit Buffer Number
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

23.3.10 Details of Transmit Buffer Status-Related Registers

23.3.10.1 RSCANnTMTRSTSy — Transmit Buffer Transmit Request Status Register (y = 0 to 1)

Access: RSCANnTMTRSTSy register can be read only in 32-bit units
RSCANnTMTRSTSyL, RSCANnTMTRSTSyH registers can be read only in 16-bit units
RSCANnTMTRSTSyLL, RSCANnTMTRSTSyLH, RSCANnTMTRSTSyHL, RSCANnTMTRSTSyHH registers can be read only in 8-bit units

Address: RSCANnTMTRSTSy: <RSCFDn_base> + 0350_H + (04_H × y)

RSCANnTMTRSTSyL: <RSCFDn_base> + 0350_H + (04_H × y),
RSCANnTMTRSTSyH: <RSCFDn_base> + 0352_H + (04_H × y)

RSCANnTMTRSTSyLL: <RSCFDn_base> + 0350_H + (04_H × y),
RSCANnTMTRSTSyLH: <RSCFDn_base> + 0351_H + (04_H × y),
RSCANnTMTRSTSyHL: <RSCFDn_base> + 0352_H + (04_H × y),
RSCANnTMTRSTSyHH: <RSCFDn_base> + 0353_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTRSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTRSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.69 RSCANnTMTRSTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmit request is present. 1: A transmit request is present.
15 to 0	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit request is present. 1: A transmit request is present.

TMTRSTSp Flags (p = 0 to 47)

These flags indicate the status of the TMTR bit in the RSCANnTMCp register.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag is set to 1.

The corresponding TMTRSTSp flag is cleared to 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

Table 23.70 shows the bit assignment.

Table 23.70 TMTRSTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

23.3.10.2 RSCANnTMTARSTSy — Transmit Buffer Transmit Abort Request Status Register (y = 0, 1)

Access: RSCANnTMTARSTSy register can be read only in 32-bit units
RSCANnTMTARSTSyL, RSCANnTMTARSTSyH registers can be read only in 16-bit units
RSCANnTMTARSTSyLL, RSCANnTMTARSTSyLH, RSCANnTMTARSTSyHL, RSCANnTMTARSTSyHH registers can be read only in 8-bit units

Address: RSCANnTMTARSTSy: <RSCFDn_base> + 0360_H + (04_H × y)

RSCANnTMTARSTSyL: <RSCFDn_base> + 0360_H + (04_H × y),
RSCANnTMTARSTSyH: <RSCFDn_base> + 0362_H + (04_H × y)

RSCANnTMTARSTSyLL: <RSCFDn_base> + 0360_H + (04_H × y),
RSCANnTMTARSTSyLH: <RSCFDn_base> + 0361_H + (04_H × y),
RSCANnTMTARSTSyHL: <RSCFDn_base> + 0362_H + (04_H × y),
RSCANnTMTARSTSyHH: <RSCFDn_base> + 0363_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTARSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTARSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.71 RSCANnTMTARSTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmit abort request is present. 1: A transmit abort request is present.
15 to 0	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit abort request is present. 1: A transmit abort request is present.

TMTARSTSp Flags (p = 0 to 47)

These flags indicate the status of the TMTAR bit in the RSCANnTMCp register.

When the TMTAR bit is set to 1 (transmit abort is requested), the corresponding TMTARSTSp flag is set to 1.

The corresponding TMTARSTSp flag is cleared to 0 when the TMTAR bit is set to 0 (transmit abort is not requested) or in channel reset mode.

Table 23.72 shows the bit assignment.

Table 23.72 TMTARSTSp Bit Assignment (1/2)

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
.	.	.
.	.	.

Table 23.72 TMTARSTSp Bit Assignment (2/2)

Bit	Channel	Transmit Buffer Number
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

23.3.10.3 RSCANnTMTCSTSy — Transmit Buffer Transmit Complete Status Register (y = 0, 1)

Access: RSCANnTMTCSTSy register can be read only in 32-bit units
RSCANnTMTCSTSyL, RSCANnTMTCSTSyH registers can be read only in 16-bit units
RSCANnTMTCSTSyLL, RSCANnTMTCSTSyLH, RSCANnTMTCSTSyHL, RSCANnTMTCSTSyHH registers can be read only in 8-bit units

Address: RSCANnTMTCSTSy: <RSCFDn_base> + 0370_H + (04_H × y)

RSCANnTMTCSTSyL: <RSCFDn_base> + 0370_H + (04_H × y),
RSCANnTMTCSTSyH: <RSCFDn_base> + 0372_H + (04_H × y)

RSCANnTMTCSTSyLL: <RSCFDn_base> + 0370_H + (04_H × y),
RSCANnTMTCSTSyLH: <RSCFDn_base> + 0371_H + (04_H × y),
RSCANnTMTCSTSyHL: <RSCFDn_base> + 0372_H + (04_H × y),
RSCANnTMTCSTSyHH: <RSCFDn_base> + 0373_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTCSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTCSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.73 RSCANnTMTCSTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission has not been completed. 1: Transmission has been completed.
15 to 0	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission has not been completed. 1: Transmission has been completed.

TMTCSTSp Flags (p = 0 to 47)

When the TMTRF[1:0] flag in the RSCANnTMSTSp register is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)), the corresponding TMTCSTSp flag is set to 1.

A TMTCSTSp flag is cleared to 0 when the corresponding TMTRF[1:0] flag is set to 00_B or in channel reset mode.

Table 23.74 shows the bit assignment.

Table 23.74 TMTCSTSp Bit Assignment (1/2)

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0

Table 23.74 TMTCSTSp Bit Assignment (2/2)

Bit	Channel	Transmit Buffer Number
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

23.3.10.4 RSCANnTMTASTSy — Transmit Buffer Transmit Abort Status Register (y = 0 to 1)

Access: RSCANnTMTASTSy register can be read only in 32-bit units
RSCANnTMTASTSyL, RSCANnTMTASTSyH registers can be read only in 16-bit units
RSCANnTMTASTSyLL, RSCANnTMTASTSyLH, RSCANnTMTASTSyHL, RSCANnTMTASTSyHH registers can be read only in 8-bit units

Address: RSCANnTMTASTSy: <RSCFDn_base> + 0380_H + (04_H × y)

RSCANnTMTASTSyL: <RSCFDn_base> + 0380_H + (04_H × y),
RSCANnTMTASTSyH: <RSCFDn_base> + 0382_H + (04_H × y)

RSCANnTMTASTSyLL: <RSCFDn_base> + 0380_H + (04_H × y),
RSCANnTMTASTSyLH: <RSCFDn_base> + 0381_H + (04_H × y),
RSCANnTMTASTSyHL: <RSCFDn_base> + 0382_H + (04_H × y),
RSCANnTMTASTSyHH: <RSCFDn_base> + 0383_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTASTSp (p = y × 32 + 31 to y × 32 × 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTASTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.75 RSCANnTMTASTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission is not aborted 1: Transmission is aborted
15 to 0	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission is not aborted. 1: Transmission is aborted.

TMTASTSp Flags (p = 0 to 47)

When the TMTRF[1:0] flag in the RSCANnTMSTSp register is set to 01_B (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.

A TMTASTSp flag is cleared to 0 when the corresponding TMTRF[1:0] flag is set to 00_B or in channel reset mode.

Table 23.76 shows the bit assignment.

Table 23.76 TMTASTSp Bit Assignment (1/2)

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0

Table 23.76 TMTASTSp Bit Assignment (2/2)

Bit	Channel	Transmit Buffer Number
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

23.3.11 Details of Transmit Queue-Related Registers

23.3.11.1 RSCANnTXQCCm — Transmit Queue Configuration and Control Register (m = 0 to 2)

Access: RSCANnTXQCCm register can be read/written in 32-bit units
RSCANnTXQCCmL register can be read/written in 16-bit units
RSCANnTXQCCmLL, RSCANnTXQCCmLH registers can be read/written in 8-bit units

Address: RSCANnTXQCCm: <RSCFDn_base> + $03A0_H$ + (04_H × m)

RSCANnTXQCCmL: <RSCFDn_base> + $03A0_H$ + (04_H × m)

RSCANnTXQCCmLL: <RSCFDn_base> + $03A0_H$ + (04_H × m),
RSCANnTXQCCmLH: <RSCFDn_base> + $03A1_H$ + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TXQIM	TXQIE	TXQDC[3:0]				—	—	—	—	—	—	—	TXQE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 23.77 RSCANnTXQCCm Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
13	TXQIM	Transmit Queue Interrupt Source Select 0: When the transmit queue becomes empty upon completion of message transmission, a transmit queue interrupt request is generated. 1: A transmit queue interrupt request is generated each time a message has been transmitted.
12	TXQIE	Transmit Queue Interrupt Enable 0: Transmit queue interrupt is disabled. 1: Transmit queue interrupt is enabled.
11 to 8	TXQDC[3:0]	Transmit Queue Depth Configuration Setting these bits to g (g = 2 to 15) makes the (g + 1) transmit queue available. Setting these bits to 0 disables the transmit queue. Setting these bits to 1 is prohibited.
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	TXQE	Transmit Queue Enable 0: The transmit queue is not used. 1: The transmit queue is used.

TXQIM Bit

This bit is used to select a transmit queue interrupt source. Modify this bit in channel reset mode.

TXQIE Bit

When the TXQIE bit is set to 1 and the source selected by the TXQIM bit occurs, an interrupt request is generated.

Set the TXQE bit to 0 before modifying the TXQIE bit.

TXQDC[3:0] Bits

These bits are used to specify the number of transmit buffers to be allocated to the transmit queues. Transmit buffers are allocated to transmit queues in descending order of buffer number, that is, from (m × 16 + 15) to (m × 16 + 0) (see **Table 23.17**). For examples of how buffer allocation is done, see **Figure 23.9**. Modify these bits only in channel reset mode.

TXQE Bit

Setting this bit to 1 makes the transmit queue available. Modify this bit in channel communication mode or channel halt mode. This bit is cleared to 0 in channel reset mode.

Before setting the TXQE bit to 1, set the TXQDC[3:0] bits to 0010_B or more.

23.3.11.2 RSCANnTXQSTSm — Transmit Queue Status Register (m = 0 to 2)

Access: RSCANnTXQSTSm register can be read/written in 32-bit units
RSCANnTXQSTSmL register can be read/written in 16-bit units
RSCANnTXQSTSmLL register can be read/written in 8-bit units

Address: RSCANnTXQSTSm: <RSCFDn_base> + $03C0_H$ + (04_H × m)

RSCANnTXQSTSmL: <RSCFDn_base> + $03C0_H$ + (04_H × m)

RSCANnTXQSTSmLL: <RSCFDn_base> + $03C0_H$ + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQIF	TXQFLL	TXQEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.78 RSCANnTXQSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 8	Reserved	When read, the undefined value is returned. When writing to these bits, write the value after reset.
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	TXQIF	Transmit Queue Interrupt Request Flag 0: No transmit queue interrupt request is present. 1: A transmit queue interrupt request is present.
1	TXQFLL	Transmit Queue Full Status Flag 0: The transmit queue is not full. 1: The transmit queue is full.
0	TXQEMP	Transmit Queue Empty Status Flag 0: The transmit queue contains messages. 1: The transmit queue contains no message (transmit queue empty).

TXQIF Flag

The TXQIF flag is set to 1 when the event specified by the TXQIM bit in the RSCANnTXQCCm register has occurred.

The TXQIF flag is cleared to 0 in channel reset mode or by writing 0 to this flag. This flag is not cleared to 0 by setting the TXQE bit in the RSCANnTXQCCm register to 0 (the transmit queue is not used).

TXQFLL Flag

The TXQFLL flag is set to 1 when the number of messages set for the transmit queue matches the transmit queue depth set by the TXQDC[3:0] bits in the RSCANnTXQCCm register.

This flag is cleared to 0 in any of the following cases.

- The number of messages set for the transmit queue is smaller than the transmit queue depth set by the TXQDC[3:0] bits.
- In channel reset mode

TXQEMP Flag

The TXQEMP flag is cleared to 0 when even a single message is set for the transmit queue.

This flag is set to 1 in any of the following cases.

- The TXQE bit is set to 0 (the transmit queue is not used).
- The transmit queue becomes empty.
- In channel reset mode

23.3.11.3 RSCANnTXQPCTRm — Transmit Queue Pointer Control Register (m = 0 to 2)

Access: RSCANnTXQPCTRm register can only be written in 32-bit units
RSCANnTXQPCTRmL register can only be written in 16-bit units
RSCANnTXQPCTRmLL register can only be written in 8-bit units

Address: RSCANnTXQPCTRm: <RSCFDn_base> + $03E0_H$ + (04_H × m)

RSCANnTXQPCTRmL: <RSCFDn_base> + $03E0_H$ + (04_H × m)

RSCANnTXQPCTRmLL: <RSCFDn_base> + $03E0_H$ + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.79 RSCANnTXQPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be the value after reset.
7 to 0	TXQPC[7:0]	Transmit Queue Pointer Control Writing FF_H to these bits moves the write pointer of the transmit queue to the next queue buffer.

TXQPC[7:0] Bits

Writing FF_H to the TXQPC[7:0] bits moves the write pointer to the next transmit queue buffer and generates a transmit request of the message. Write transmit messages to the RSCANnTMIDp, RSCANnTMPTRp, RSCANnTMDF0_p, and RSCANnTMDF1_p registers (p = 15, 31, and 47) before writing FF_H to the TXQPC[7:0] bits.

When writing FF_H to these bits, make sure that the TXQE bit in the RSCANnTXQCCm register is set to 1 (the transmit queue is used) and the TXQFLL flag in the RSCANnTXQSTSm register is 0 (he transmit queue is not full).

23.3.12 Details of Transmission History-Related Registers

23.3.12.1 RSCANnTHLCCm — Transmit History Configuration and Control Register (m = 0 to 2)

Access: RSCANnTHLCCm register can be read/written in 32-bit units
RSCANnTHLCCmL register can be read/written in 16-bit units
RSCANnTHLCCmLL, RSCANnTHLCCmLH registers can be read/written in 8-bit units

Address: RSCANnTHLCCm: <RSCFDn_base> + 0400_H + (04_H × m)

RSCANnTHLCCmL: <RSCFDn_base> + 0400_H + (04_H × m)

RSCANnTHLCCmLL: <RSCFDn_base> + 0400_H + (04_H × m),
RSCANnTHLCCmLH: <RSCFDn_base> + 0401_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 23.80 RSCANnTHLCCm Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
10	THLDTE	Transmit History Target Buffer Select 0: Entry from transmit/receive FIFO buffers and transmit queue 1: Entry from transmit buffers, transmit/receive FIFO buffers, and transmit queue
9	THLIM	Transmit History Interrupt Source Select 0: When 12 sets of data have been stored in the transmit history buffer 1: When a single set of transmit history data has been stored
8	THLIE	Transmit History Interrupt Enable 0: Transmit history interrupt is disabled. 1: Transmit history interrupt is enabled.
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	THLE	Transmit History Buffer Enable 0: Transmit history buffer is not used. 1: Transmit history buffer is used.

THLDTE Bit

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers and the transmit queue is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers, transmit/receive FIFO buffers, and the transmit queue is stored in the transmit history buffer.

Modify this bit only in channel reset mode.

THLIM Bit

This bit is used to select a transmit history interrupt source.

Modify this bit only in channel reset mode.

THLIE Bit

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit only when the THLE bit set to 0.

THLE Bit

Setting this bit to 1 makes the transmit history buffer available. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer.

Modify this bit in channel communication mode or channel halt mode.

This bit is cleared to 0 in channel reset mode.

23.3.12.2 RSCANnTHLSTSm — Transmit History Status Register (m = 0 to 2)

Access: RSCANnTHLSTSm register can be read/written in 32-bit units
RSCANnTHLSTSmL register can be read/written in 16-bit units
RSCANnTHLSTSmLL register can be read/written in 8-bit units
RSCANnTHLSTSmLH register is a read-only register that can be read in 8-bit units

Address: RSCANnTHLSTSm: <RSCFDn_base> + 0420_H + (04_H × m)

RSCANnTHLSTSmL: <RSCFDn_base> + 0420_H + (04_H × m)

RSCANnTHLSTSmLL: <RSCFDn_base> + 0420_H + (04_H × m),
RSCANnTHLSTSmLH: <RSCFDn_base> + 0421_H + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THLMC[4:0]					—	—	—	—	THLIF	THLELT	THLFLL	THLEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.81 RSCANnTHLSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 8	THLMC[4:0]	Transmit History Buffer Unread Data Counter These bits indicate the number of unread data sets stored in the transmit history buffer.
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	THLIF	Transmit History Interrupt Request Flag 0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present.
2	THLELT	Transmit History Buffer Overflow Flag 0: Transmit history buffer overflow has not occurred. 1: Transmit history buffer overflow has occurred.
1	THLFLL	Transmit history Buffer Full Status Flag 0: Transmit history buffer is not full. 1: Transmit history buffer is full.
0	THLEMP	Transmit History Buffer Empty Status Flag 0: Transmit history buffer contains unread data. 1: Transmit history buffer contains no unread data (buffer empty).

THLMC[4:0] Bits

These bits indicate the number of unread data sets stored in the transmit history buffer.

These bits are cleared to 0 in channel reset mode.

THLIF Flag

The THLIF flag is set to 1 when the interrupt source specified with the THLIM bit in the RSCANnTHLCCm register occurs.

This flag is cleared to 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLELT Flag

The THLELT flag is set to 1 when an attempt is made to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded. This flag becomes 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLFLL Flag

The THLFLL flag is set to 1 when 16 data sets have been stored in the transmit history buffer, and is cleared to 0 when the number of data sets stored in the transmit history buffer has decreased to less than 16. This bit is also cleared to 0 in channel reset mode or when the THLE bit in the RSCANnTHLCCm register is set to 0 (transmit history buffer is not used).

THLEMP Flag

The THLEMP flag is cleared to 0 when even a single set of transmit history data has been stored in the transmit history buffer.

This flag is set to 1 when all the data in the transmit history buffer has been read. This flag is also set to 1 in channel reset mode or when the THLE bit in the RSCANnTHLCCm register is set to 0 (transmit history buffer is not used).

NOTE

To clear THLIF or THLELT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

23.3.12.3 RSCANnTHLPCTRm — Transmit History Pointer Control Register (m = 0 to 2)

Access: RSCANnTHLPCTRm register can only be written in 32-bit units
RSCANnTHLPCTRmL register can only be written in 16-bit units
RSCANnTHLPCTRmLL register can only be written in 8-bit units

Address: RSCANnTHLPCTRm: <RSCFDn_base> + 0440_H + (04_H × m)

RSCANnTHLPCTRmL: <RSCFDn_base> + 0440_H + (04_H × m)

RSCANnTHLPCTRmLL: <RSCFDn_base> + 0440_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	THLPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.82 RSCANnTHLPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write the value after reset.
7 to 0	THLPC[7:0]	Transmit History List Pointer Control Writing FF_H to these bits moves the read pointer to the next unread data in the transmit history buffer.

THLPC[7:0] Bits

When the THLPC[7:0] bits are set to FF_H, the read pointer moves to the next data in the transmit history buffer. At this time, the THLMC[4:0] (transmit history buffer unread data counter) value in the RSCANnTHLSTSm register is decremented. Write FF_H to the THLPC[7:0] bits after reading from the RSCANnTHLACCm register.

When writing FF_H to these bits, make sure that the THLE bit in the RSCANnTHLCCm register is set to 1 (transmit history buffer is used) and the THLEMP flag in the RSCANnTHLSTSm register is 0.

23.3.12.4 RSCANnTHLACCm — Transmit History Access Register (m = 0 to 2)

Access: RSCANnTHLACCm register can be read only in 32-bit units
RSCANnTHLACCmL, RSCANnTHLACCmH registers can be read only in 16-bit units
RSCANnTHLACCmLL, RSCANnTHLACCmLH, RSCANnTHLACCmHL, RSCANnTHLACCmHH registers can be read only in 8-bit units

Address: RSCANnTHLACCm: <RSCFDn_base> + 1800_H + (04_H × m)

RSCANnTHLACCmL: <RSCFDn_base> + 1800_H + (04_H × m),
RSCANnTHLACCmH: <RSCFDn_base> + 1802_H + (04_H × m)

RSCANnTHLACCmLL: <RSCFDn_base> + 1800_H + (04_H × m),
RSCANnTHLACCmLH: <RSCFDn_base> + 1801_H + (04_H × m),
RSCANnTHLACCmHL: <RSCFDn_base> + 1802_H + (04_H × m),
RSCANnTHLACCmHH: <RSCFDn_base> + 1803_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TID[7:0]								—	BN[3:0]				BT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.83 RSCANnTHLACCm Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTS[15:0]	Timestamp Data The timestamp data of stored data can be read.
15 to 8	TID[7:0]	Label Data The label information of stored data can be read.
7	Reserved	When read, the value after reset is returned.
6 to 3	BN[3:0]	Buffer Number Data The buffer number of transmit source (transmit buffer, transmit/receive FIFO or transmit queue) can be read.
2 to 0	BT[2:0]	Buffer Type Data b2 b1 b0 0 0 1: Transmit buffer 0 1 0: Transmit/receive FIFO buffer 1 0 0: Transmit queue

TMTS[15:0] Bits

When the TMTSCE bit in the RSCANnGCFG register is 1, timestamp values in transmit history data stored in the transmit history buffer are displayed. When the TMTSCE bit is 0, these bits are always read as 0.

TID[7:0] Bits

These bits indicate the label information of transmit history data stored in the transmit history buffer.

BN[3:0] Bits

These bits indicate the transmit source buffer number in the transmit history data stored in the transmit history buffer.

BT[2:0] Bits

These bits indicate the type of the transmit source buffer in the transmit history data stored in the transmit history buffer.

23.3.13 Details of Test-Related Registers

23.3.13.1 RSCANnGTSTCFG — Global Test Configuration Register

Access: RSCANnGTSTCFG register can be read/written in 32-bit units
RSCANnGTSTCFGL, RSCANnGTSTCFGH registers can be read/written in 16-bit units
RSCANnGTSTCFGLL, RSCANnGTSTCFGHL registers can be read/written in 8-bit units

Address: RSCANnGTSTCFG: <RSCFDn_base> + 0468_H

RSCANnGTSTCFGL: <RSCFDn_base> + 0468_H,
RSCANnGTSTCFGH: <RSCFDn_base> + $046A_H$

RSCANnGTSTCFGLL: <RSCFDn_base> + 0468_H,
RSCANnGTSTCFGHL: <RSCFDn_base> + $046A_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	RTMPS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	C2ICBCE	C1ICBCE	C0ICBCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 23.84 RSCANnGTSTCFG Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
22 to 16	RTMPS[6:0]	RAM Test Page Configuration Set a value within a range of page 0 (00_H) to page 29 ($1D_H$).
15 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	C2ICBCE	CAN2 Inter-channel Communication Test Enable 0: CAN2 inter-channel communication test is disabled 1: CAN2 inter-channel communication test is enabled.
1	C1ICBCE	CAN1 Inter-Channel Communication Test Enable 0: CAN1 inter-channel communication test is disabled. 1: CAN1 inter-channel communication test is enabled.
0	C0ICBCE	CAN0 Inter-Channel Communication Test Enable 0: CAN0 inter-channel communication test is disabled. 1: CAN0 inter-channel communication test is enabled.

Modify the RSCANnGTSTCFG register only in global test mode.

RTMPS[6:0] Bits

These bits are used to set the RAM test target page number for RAM test. Set a value in the range of 00_H to $1D_H$, inclusive.

C2ICBCE Bit

Setting this bit to 1 enables the channel 2 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C1ICBCE Bit

Setting this bit to 1 enables the channel 1 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C0ICBCE Bit

Setting this bit to 1 enables the channel 0 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

23.3.13.2 RSCANnGTSTCTR — Global Test Control Register

Access: RSCANnGTSTCTR register can be read/written in 32-bit units
RSCANnGTSTCTRL register can be read/written in 16-bit units
RSCANnGTSTCTRLL register can be read/written in 8-bit units

Address: RSCANnGTSTCTR: <RSCFDn_base> + $046C_H$

RSCANnGTSTCTRL: <RSCFDn_base> + $046C_H$

RSCANnGTSTCTRLL: <RSCFDn_base> + $046C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	ICBCTME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Table 23.85 RSCANnGTSTCTR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	RTME	RAM Test Enable 0: RAM test is disabled. 1: RAM test is enabled.
1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	ICBCTME	Communication Test between Channels Enable 0: Communication test between channels disabled 1: Communication test between channels enabled

RTME Bit

Setting this bit to 1 enables the RAM test. Modify this bit only in global test mode.

This bit is cleared to 0 in global reset mode (See Figure 23.37, RAM Test Setting Procedure).

1. Set the GMDC[1:0] bits in the RSCANnGCTR register to 10_B (Global test mode).
2. Set the RTME bit to 1.
3. Check that the RTME bit is set to 1.

ICBCTME Bit

When this bit is set to 1, a communication test is enabled between the channels for which the CmICBCE bit (m = 0 to 2) in the RSCANnGTSTCFG register has been set to 1. Modify the ICBCTME bit only in global test mode.

This bit is cleared to 0 in global reset mode.

23.3.13.3 RSCANnGLOCKK — Global Lock Key Register

Access: RSCANnGLOCKK register can be write only in 32-bit units.
RSCANnGLOCKKL register can be write only in 16-bit units.

Address: RSCANnGLOCKK: <RSCFDn_base> + $047C_H$

RSCANnGLOCKKL: <RSCFDn_base> + $047C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1

Note 1. Writing to these bits is effective only when the RS-CANFD module is in global test mode.

Table 23.86 RSCANnGLOCKK Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing these bits, write the value after reset.
15 to 0	LOCK[15:0]	Lock Key These bits are key bits to release protection of test mode.

The RSCANnGLOCKK register releases protection of special test bits and is write only.

For the protection release data, see Section 23.11.4.2, Procedure for Releasing the Protection.

LOCK[15:0] Bits

Writing the protection release data to the LOCK[15:0] bits in succession enables writing 1 to the RTME bit in the RSCANnGTSTCTR register.

After the protection has been released, writing to the I/O register area (<RSCFDn_base> + 0000_H to <RSCFDn_base> + $04FF_H$) of the CAN (except the RAM) enables the protection again.

Reading from the I/O register area of the CAN or reading from/writing to other areas does not enable the protection.

23.3.13.4 RSCANnRPGACCr — RAM Test Page Access Register (r = 0 to 63)

Access: RSCANnRPGACCr register can be read/written in 32-bit units
RSCANnRPGACCrL, RSCANnRPGACCrH registers can be read/written in 16-bit units
RSCANnRPGACCrLL, RSCANnRPGACCrLH, RSCANnRPGACCrHL, RSCANnRPGACCrHH registers can be read/written in 8-bit units

Address: RSCANnRPGACCr: <RSCFDn_base> + 1900_H + (04_H × r)

RSCANnRPGACCrL: <RSCFDn_base> + 1900_H + (04_H × r)),
RSCANnRPGACCrH: <RSCFDn_base> + 1902_H + (04_H × r))

RSCANnRPGACCrLL: <RSCFDn_base> + 1900_H + (04_H × r)),
RSCANnRPGACCrLH: <RSCFDn_base> + 1901_H + (04_H × r)),
RSCANnRPGACCrHL: <RSCFDn_base> + 1902_H + (04_H × r)),
RSCANnRPGACCrHH: <RSCFDn_base> + 1903_H + (04_H × r))

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.87 RSCANnRPGACCr Register Contents

Bit Position	Bit Name	Function
31 to 0	RDTA [31:0]	RAM Data Test Access Data can be read and written in RSCAN RAM.

Modify the RSCANnRPGACCr register in global test mode with the RTME bit in the RSCANnGTSTCTR register set to 1 (RAM test is enabled).

The RSCANnRPGACCr register is readable and writable when the RTME bit is set to 1.

23.4 Registers (CAN FD Mode)

This section describes all registers to be used when the RS-CANFD is used in CAN FD mode.

23.4.1 List of Registers

The following tables list RS-CANFD registers to be used in CAN FD mode.

For details about <RSCFDn_base>, see Section 23.1.2, Register Base Address.

For details about registers initialized in Global reset mode or Channel reset mode, see following:

- Table 23.177, Registers Initialized in Global Reset Mode or Channel Reset Mode
- Table 23.178, Registers Initialized Only in Global Reset Mode

Table 23.88 Registers (1/3)

Module	Register	Symbol	Address
Interface mode-related registers			
RSCFDn	Global interface mode select register	RSCFDnCFDGRMCFG	<RSCFDn_base> + $04FC_H$
Channel-related registers			
RSCFDn	Channel m nominal bit rate configuration register	RSCFDnCFDCmNCFG	<RSCFDn_base> + 0000_H + (10_H × m)
RSCFDn	Channel m control register	RSCFDnCFDCmCTR	<RSCFDn_base> + 0004_H + (10_H × m)
RSCFDn	Channel m status register	RSCFDnCFDCmSTS	<RSCFDn_base> + 0008_H + (10_H × m)
RSCFDn	Channel m error flag register	RSCFDnCFDCmERFL	<RSCFDn_base> + $000C_H$ + (10_H × m)
RSCFDn	Channel m data bit rate configuration register	RSCFDnCFDCmDCFG	<RSCFDn_base> + 0500_H + (20_H × m)
RSCFDn	Channel m CAN FD configuration register	RSCFDnCFDCmFDCFG	<RSCFDn_base> + 0504_H + (20_H × m)
RSCFDn	Channel m CAN FD control register	RSCFDnCFDCmFDCTR	<RSCFDn_base> + 0508_H + (20_H × m)
RSCFDn	Channel m CAN FD status register	RSCFDnCFDCmFDSTS	<RSCFDn_base> + $050C_H$ + (20_H × m)
RSCFDn	Channel m CAN FD CRC register	RSCFDnCFDCmFDCRC	<RSCFDn_base> + 0510_H + (20_H × m)
Global-related registers			
RSCFDn	Global configuration register	RSCFDnCFDGCFG	<RSCFDn_base> + 0084_H
RSCFDn	Global control register	RSCFDnCFDGCTR	<RSCFDn_base> + 0088_H
RSCFDn	Global status register	RSCFDnCFDGSTS	<RSCFDn_base> + $008C_H$
RSCFDn	Global error flag register	RSCFDnCFDGERFL	<RSCFDn_base> + 0090_H
RSCFDn	Global timestamp counter register	RSCFDnCFDGTSC	<RSCFDn_base> + 0094_H
RSCFDn	Global TX Interrupt Status Register 0	RSCFDnCFDGTINTSTS0	<RSCFDn_base> + 0460_H
RSCFDn	Glboal FD configuration register	RSCFDnCFDGFDCFG	<RSCFDn_base> + 0474_H
RSCFDn	Global CRC configuration register	RSCFDnCFDGCRCCFG	<RSCFDn_base> + 0478_H
Receive rule-related registers			
RSCFDn	Receive Rule Entry Control Register	RSCFDnCFDGAFLECTR	<RSCFDn_base> + 0098_H
RSCFDn	Receive Rule Configuration Register 0	RSCFDnCFDGAFLCFG0	<RSCFDn_base> + $009C_H$
RSCFDn	Receive Rule ID Register j	RSCFDnCFDGAFLIDj	<RSCFDn_base> + 1000_H + (10_H × j)
RSCFDn	Receive Rule Mask Register j	RSCFDnCFDGAFLMj	<RSCFDn_base> + 1004_H + (10_H × j)
RSCFDn	Receive Rule Pointer 0 Register j	RSCFDnCFDGAFLP0_j	<RSCFDn_base> + 1008_H + (10_H × j)
RSCFDn	Receive Rule Pointer 1 Register j	RSCFDnCFDGAFLP1_j	<RSCFDn_base> + $100C_H$ + (10_H × j)
Receive buffer-related registers			
RSCFDn	Receive Buffer Number Register	RSCFDnCFDRMNB	<RSCFDn_base> + $00A4_H$
RSCFDn	Receive Buffer New Data Register y	RSCFDnCFDRMNDy	<RSCFDn_base> + $00A8_H$ + (04_H × y)
RSCFDn	Receive Buffer ID Register q	RSCFDnCFDRMIDq	<RSCFDn_base> + 2000_H + (20_H × q)
RSCFDn	Receive Buffer Pointer Register q	RSCFDnCFDRMPTRq	<RSCFDn_base> + 2004_H + (20_H × q)
RSCFDn	Receive buffer CAN FD status register q	RSCFDnCFDRMFDSTSq	<RSCFDn_base> + 2008_H + (20_H × q)

Table 23.88 Registers (2/3)

Module	Register	Symbol	Address
RSCFDn	Receive Buffer Data Field b Register q	RSCFDnCFDRMDFb_q	<RSCFDn_base> + $200C_H$ + (04_H × b) + (20_H × q)
Receive FIFO buffer-related registers			
RSCFDn	Receive FIFO Buffer Configuration and Control Register x	RSCFDnCFDRFCCx	<RSCFDn_base> + $00B8_H$ + (04_H × x)
RSCFDn	Receive FIFO Buffer Status Register x	RSCFDnCFDRFSTSx	<RSCFDn_base> + $00D8_H$ + (04_H × x)
RSCFDn	Receive FIFO Buffer Pointer Control Register x	RSCFDnCFDRFPCTRx	<RSCFDn_base> + $00F8_H$ + (04_H × x)
RSCFDn	Receive FIFO Buffer Access ID Register x	RSCFDnCFDRFIDx	<RSCFDn_base> + 3000_H + (80_H × x)
RSCFDn	Receive FIFO Buffer Access Pointer Register x	RSCFDnCFDRFPTRx	<RSCFDn_base> + 3004_H + (80_H × x)
RSCFDn	Receive FIFO CAN FD status register x	RSCFDnCFDRFFDSTSx	<RSCFDn_base> + 3008_H + (80_H × x)
RSCFDn	Receive FIFO Buffer Access Data Field d Register x	RSCFDnCFDRFDFd_x	<RSCFDn_base> + $300C_H$ + (04_H × d) + (80_H × x)
Transmit/Receive FIFO buffer related registers			
RSCFDn	Transmit/receive FIFO Buffer Configuration and Control Register k	RSCFDnCFDCFCCk	<RSCFDn_base> + 0118_H + (04_H × k)
RSCFDn	Transmit/receive FIFO Buffer Status Register k	RSCFDnCFDCFSTSk	<RSCFDn_base> + 0178_H + (04_H × k)
RSCFDn	Transmit/receive FIFO Buffer Pointer Control Register k	RSCFDnCFDCFPCTRk	<RSCFDn_base> + $01D8_H$ + (04_H × k)
RSCFDn	Transmit/receive FIFO Buffer Access ID Register k	RSCFDnCFDCFIDk	<RSCFDn_base> + 3400_H + (80_H × k)
RSCFDn	Transmit/receive FIFO Buffer Access Pointer Register k	RSCFDnCFDCFPTRk	<RSCFDn_base> + 3404_H + (80_H × k)
RSCFDn	Transmit/receive FIFO CAN FD configuration/status register k	RSCFDnCFDCFFDCSTSk	<RSCFDn_base> + 3408_H + (80_H × k)
RSCFDn	Transmit/receive FIFO Buffer Access Data Field d Register k	RSCFDnCFDCFDFd_k	<RSCFDn_base> + $340C_H$ + (04_H × d) + (80_H × k)
FIFO status-related registers			
RSCFDn	FIFO Empty Status Register	RSCFDnCFDFESTS	<RSCFDn_base> + 0238_H
RSCFDn	FIFO Full Status Register	RSCFDnCFDFFSTS	<RSCFDn_base> + $023C_H$
RSCFDn	FIFO Message Lost Status Register	RSCFDnCFDFMSTS	<RSCFDn_base> + 0240_H
RSCFDn	Receive FIFO Buffer Interrupt Flag Status Register	RSCFDnCFDRFISTS	<RSCFDn_base> + 0244_H
RSCFDn	Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register	RSCFDnCFDCFRISTS	<RSCFDn_base> + 0248_H
RSCFDn	Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register	RSCFDnCFDCFTISTS	<RSCFDn_base> + $024C_H$
Transmit buffer-related registers			
RSCFDn	Transmit Buffer Control Register p	RSCFDnCFDTMCp	<RSCFDn_base> + 0250_H + (01_H × p)
RSCFDn	Transmit Buffer Status Register p	RSCFDnCFDTMSTSp	<RSCFDn_base> + $02D0_H$ + (01_H × p)
RSCFDn	Transmit Buffer ID Register p	RSCFDnCFDTMIDp	<RSCFDn_base> + 4000_H + (20_H × p)
RSCFDn	Transmit Buffer Pointer Register p	RSCFDnCFDTMPTRp	<RSCFDn_base> + 4004_H + (20_H × p)
RSCFDn	Transmit buffer CAN FD configuration register p	RSCFDnCFDTMFDCTRp	<RSCFDn_base> + 4008_H + (20_H × p)
RSCFDn	Transmit Buffer Data Field b Register p	RSCFDnCFDTMDFb_p	<RSCFDn_base> + $400C_H$ + (04_H × b) + (20_H × p)
RSCFDn	Transmit Buffer Interrupt Enable Configuration Register y	RSCFDnCFDTMIECy	<RSCFDn_base> + 0390_H + (04_H × y)
Transmit buffer status-related registers			
RSCFDn	Transmit Buffer Transmit Request Status Register y	RSCFDnCFDTMTRSTSy	<RSCFDn_base> + 0350_H + (04_H × y)
RSCFDn	Transmit Buffer Transmit Abort Request Status Register y	RSCFDnCFDTMTARSTSy	<RSCFDn_base> + 0360_H + (04_H × y)
RSCFDn	Transmit Buffer Transmit Complete Status Register y	RSCFDnCFDTMTCSTSy	<RSCFDn_base> + 0370_H + (04_H × y)
RSCFDn	Transmit Buffer Transmit Abort Status Register y	RSCFDnCFDTMTASTSy	<RSCFDn_base> + 0380_H + (04_H × y)
Transmit queue-related registers			
RSCFDn	Transmit Queue Configuration and Control Register m	RSCFDnCFDTXQCCm	<RSCFDn_base> + $03A0_H$ + (04_H × m)
RSCFDn	Transmit Queue Status Register m	RSCFDnCFDTXQSTSm	<RSCFDn_base> + $03C0_H$ + (04_H × m)
RSCFDn	Transmit Queue Pointer Control Register m	RSCFDnCFDTXQPCTRm	<RSCFDn_base> + $03E0_H$ + (04_H × m)
Transmit history-related registers			
RSCFDn	Transmit History Configuration and Control Register m	RSCFDnCFDTHLCCm	<RSCFDn_base> + 0400_H + (04_H × m)

Table 23.88 Registers (3/3)

Module	Register	Symbol	Address
RSCFDn	Transmit History Status Register m	RSCFDnCFDTHLSTSm	<RSCFDn_base> + 0420_H + (04_H × m)
RSCFDn	Transmit History Pointer Control Register m	RSCFDnCFDTHLPCTRm	<RSCFDn_base> + 0440_H + (04_H × m)
RSCFDn	Transmit History Access Register m	RSCFDnCFDTHLACCm	<RSCFDn_base> + 6000_H + (04_H × m)
Test-related registers			
RSCFDn	Global Test Configuration Register	RSCFDnCFDGTSTCFG	<RSCFDn_base> + 0468_H
RSCFDn	Global Test Control Register	RSCFDnCFDGTSTCTR	<RSCFDn_base> + $046C_H$
RSCFDn	Global Lock Key Register	RSCFDnCFDGLOCKK	<RSCFDn_base> + $047C_H$
RSCFDn	RAM Test Page Access Register r	RSCFDnCFDRPGACCr	<RSCFDn_base> + 6400_H + (04_H × r)

Table 23.89 Transmit Buffer p Allocated to Each Channel

	CANm
Transmit buffer p	Transmit buffer 16 × m + 0
	Transmit buffer 16 × m + 1
	Transmit buffer 16 × m + 2
	Transmit buffer 16 × m + 3
	Transmit buffer 16 × m + 4
	Transmit buffer 16 × m + 5
	Transmit buffer 16 × m + 6
	Transmit buffer 16 × m + 7
	Transmit buffer 16 × m + 8
	Transmit buffer 16 × m + 9
	Transmit buffer 16 × m + 10
	Transmit buffer 16 × m + 11
	Transmit buffer 16 × m + 12
	Transmit buffer 16 × m + 13
	Transmit buffer 16 × m + 14
	Transmit buffer 16 × m + 15

Table 23.90 Transmit/Receive FIFO Buffer k Allocated to Each Channel

	CANm
Transmit/receive FIFO buffer k	Transmit/receive FIFO buffer 3 × m + 0
	Transmit/receive FIFO buffer 3 × m + 1
	Transmit/receive FIFO buffer 3 × m + 2

Table 23.91 Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer by the Setting of Bits CFTML[3:0] (1/2)

Setting of Bits CFTML[3:0]	Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer
0000_B	Transmit buffer 16 × m + 0
0001_B	Transmit buffer 16 × m + 1
0010_B	Transmit buffer 16 × m + 2
0011_B	Transmit buffer 16 × m + 3
0100_B	Transmit buffer 16 × m + 4
0101_B	Transmit buffer 16 × m + 5

Table 23.91 Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer by the Setting of Bits CFTML[3:0] (2/2)

Setting of Bits CFTML[3:0]	Transmit Buffer p Linked to the Transmit/Receive FIFO Buffer
0110_B	Transmit buffer 16 × m + 6
0111_B	Transmit buffer 16 × m + 7
1000_B	Transmit buffer 16 × m + 8
1001_B	Transmit buffer 16 × m + 9
1010_B	Transmit buffer 16 × m + 10
1011_B	Transmit buffer 16 × m + 11
1100_B	Transmit buffer 16 × m + 12
1101_B	Transmit buffer 16 × m + 13
1110_B	Transmit buffer 16 × m + 14
1111_B	Transmit buffer 16 × m + 15

Table 23.92 Transmit Buffer p Allocated to the Transmit Queue of Each Channel

Setting of Bits TXQDC[3:0]	Transmit Buffer p Allocated to the Transmit Queue
0000_B	Setting prohibited
0001_B	Setting prohibited
0010_B	Transmit buffer 16 × m + 15 to 16 × m + 13
0011_B	Transmit buffer 16 × m + 15 to 16 × m + 12
0100_B	Transmit buffer 16 × m + 15 to 16 × m + 11
0101_B	Transmit buffer 16 × m + 15 to 16 × m + 10
0110_B	Transmit buffer 16 × m + 15 to 16 × m + 9
0111_B	Transmit buffer 16 × m + 15 to 16 × m + 8
1000_B	Transmit buffer 16 × m + 15 to 16 × m + 7
1001_B	Transmit buffer 16 × m + 15 to 16 × m + 6
1010_B	Transmit buffer 16 × m + 15 to 16 × m + 5
1011_B	Transmit buffer 16 × m + 15 to 16 × m + 4
1100_B	Transmit buffer 16 × m + 15 to 16 × m + 3
1101_B	Transmit buffer 16 × m + 15 to 16 × m + 2
1110_B	Transmit buffer 16 × m + 15 to 16 × m + 1
1111_B	Transmit buffer 16 × m + 15 to 16 × m + 0

23.4.2 Details of Interface Mode-Related Registers

23.4.2.1 RSCFDnCFDGRMCFG — Global Interface Mode Select Register

Access: RSCFDnCFDGRMCFG register can be read/written in 32-bit units
RSCFDnCFDGRMCFGL register can be read/written in 16-bit units
RSCFDnCFDGRMCFGLL register can be read/written in 8-bit units

Address: RSCFDnCFDGRMCFG: <RSCFDn_base> + $04FC_H$

RSCFDnCFDGRMCFGL: <RSCFDn_base> + $04FC_H$

RSCFDnCFDGRMCFGLL: <RSCFDn_base> + $04FC_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RCMC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 23.93 RSCFDnCFDGRMCFG Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
0	RCMC	Interface Mode Select 0: Classical CAN mode 1: CAN FD mode

Note 1. The RSCANnGRMCFG register and the RSCFDnCFDGRMCFG register are the same register. Therefore, set either one of the registers.

Modify the RSCFDnCFDGRMCFG register only in global reset mode. Before setting other RS-CANFD registers, set this register.

RCMC Bit

Setting this bit to 1 makes CAN-FD mode available. To switch classical CAN mode to CAN FD mode, set the value after reset to all registers and bits allocated to the register map of classical CAN mode and then modify the RSCFDnCFDGRMCFG register.

To switch the RS-CAN FD module from classical CAN mode to CAN FD mode, set the values after reset to all respective registers and bits allocated only to the register map in classical CAN mode, and then modify the value of RSCFDnCFDGRMCFG register.

23.4.3 Details of Channel-Related Registers

23.4.3.1 RSCFDnCFDCmNCFG — Channel Nominal Bit Rate Configuration Register (m = 0 to 2)

Access: RSCFDnCFDCmNCFG register can be read/written in 32-bit units
RSCFDnCFDCmNCFGL, RSCFDnCFDCmNCFGH registers can be read/written in 16-bit units
RSCFDnCFDCmNCFGLL, RSCFDnCFDCmNCFGLH, RSCFDnCFDCmNCFGHL,
RSCFDnCFDCmNCFGHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCmNCFG: <RSCFDn_base> + 0000_H + (10_H × m)

RSCFDnCFDCmNCFGL: <RSCFDn_base> + 0000_H + (10_H × m),
RSCFDnCFDCmNCFGH: <RSCFDn_base> + 0002_H + (10_H × m)

RSCFDnCFDCmNCFGLL: <RSCFDn_base> + 0000_H + (10_H × m),
RSCFDnCFDCmNCFGLH: <RSCFDn_base> + 0001_H + (10_H × m),
RSCFDnCFDCmNCFGHL: <RSCFDn_base> + 0002_H + (10_H × m),
RSCFDnCFDCmNCFGHH: <RSCFDn_base> + 0003_H + (10_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	NTSEG2 [4:0]					—	NTSEG1 [6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	NSJW [4:0]					—	NBRP [9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.94 RSCFDnCFDCmNCFG register contents

Bit Position	Bit Name	Function
31 to 29	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
28 to 24	NTSEG2 [4:0]	Nominal Bit Rate Time Segment 2 Control b28 b27 b26 b25 b24 0 0 0 0 0: Setting prohibited 0 0 0 0 1: 2 Tq : : 1 1 1 1 0: 31 Tq 1 1 1 1 1: 32 Tq
23	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
22 to 16	NTSEG1 [6:0]	Nominal Bit Rate Time Segment 1 Control b22 b21 b20 b19 b18 b17 b16 0 0 0 0 0 0 0: Setting prohibited 0 0 0 0 0 0 1: Setting prohibited 0 0 0 0 0 1 0: Setting prohibited 0 0 0 0 0 1 1: 4 Tq : : 1 1 1 1 1 1 0: 127 Tq 1 1 1 1 1 1 1: 128 Tq

Table 23.94 RSCFDnCFDCmNCFG register contents

Bit Position	Bit Name	Function
15 to 11	NSJW [4:0]	Nominal Bit Rate Resynchronization Jump Width Control b15 b14 b13 b12 b11 0 0 0 0 0: 1 Tq 0 0 0 0 1: 2 Tq 0 0 0 1 0: 3 Tq : : 1 1 1 1 0: 31 Tq 1 1 1 1 1: 32 Tq
10	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
9 to 0	NBRP [9:0]	Nominal Bit Rate Prescaler Division Ratio Setting When the set value = P (0 to 1023), the nominal bit rate prescaler divides fCAN by (P + 1).

Modify the RSCFDnCFDCmNCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode, and then transition to channel communication mode or channel halt mode. For the description and settings of bit timing parameters, see Section 23.11.1, Initial Settings.

NTSEG2[4:0] Bits

These bits are used to specify a Tq value for the length of phase segment 2 (PHASE_SEG2) of nominal bit rate. Allowed values are 2 Tq to 32 Tq, inclusive. Set a value smaller than the value of the NTSEG1[6:0] bits.

NTSEG1[6:0] Bits

These bits specify the total length of propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1) of nominal bit rate as a Tq value.

A value of 4 to 128 Tq is settable.

NSJW[4:0] Bits

These bits specify the resynchronization jump width of nominal bit rate as a Tq value. A value of 1 to 32 Tq is settable. Specify a value equal to or smaller than the NTSEG2[4:0] value.

NBRP[9:0] Bits

The clock obtained by dividing the CAN clock (fCAN) by the nominal bit rate prescaler ((NBRP[9:0]) + 1) becomes CANmTq(N) clock (fCANTQ(N)m). One clock of the CANmTq(N) clock becomes one Time Quantum (Tq).

Be sure to specify the same value for both NBRP[9:0] and DBRP[7:0].
To specify a different value for the nominal bit rate and the data bit rate, change the values of the RSCFDnCFDCmNCFG.NTSEG1[6:0] and NTSEG2[4:0] bits and RSCFDnCFDCmDCFG.DTSEG1[3:0] and DTSEG2[2:0] bits, respectively.

23.4.3.2 RSCFDnCFDCmCTR — Channel Control Register (m = 0 to 2)

Access: RSCFDnCFDCmCTR register can be read/written in 32-bit units
RSCFDnCFDCmCTRL, RSCFDnCFDCmCTRH registers can be read/written in 16-bit units
RSCFDnCFDCmCTRLL, RSCFDnCFDCmCTRLH, RSCFDnCFDCmCTRHL, RSCFDnCFDCmCTRHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCmCTR: <RSCFDn_base> + 0004_H + (10_H × m)

RSCFDnCFDCmCTRL: <RSCFDn_base> + 0004_H + (10_H × m),
RSCFDnCFDCmCTRH: <RSCFDn_base> + 0006_H + (10_H × m)

RSCFDnCFDCmCTRLL: <RSCFDn_base> + 0004_H + (10_H × m),
RSCFDnCFDCmCTRLH: <RSCFDn_base> + 0005_H + (10_H × m),
RSCFDnCFDCmCTRHL: <RSCFDn_base> + 0006_H + (10_H × m),
RSCFDnCFDCmCTRHH: <RSCFDn_base> + 0007_H + (10_H × m)

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ROM	CRCT	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	TDCVFIE	SOCOIE	EOCOIE	TAIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 23.95 RSCFDnCFDCmCTR Register Contents (1/2)

Bit Position	Bit Name	Function
31	ROM	Restricted Operation Mode Enable 0: Restricted operation mode is disabled 1: Restricted operation mode is enabled.
30	CRCT	CRC Error Test Enable 0: The first bit of the reception ID field is not inverted. 1: The first bit of the reception ID field is inverted.
29 to 27	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
26, 25	CTMS[1:0]	Communication Test Mode Select b26 b25 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback mode) 1 1: Self-test mode 1 (internal loopback mode)
24	CTME	Communication Test Mode Enable 0: Communication test mode is disabled. 1: Communication test mode is enabled.
23	ERRD	Error Display Mode Select 0: Error flags are displayed only for the first error information after bits 14 to 8 in the RSCFDnCFDCmERFL register are all cleared. 1: Error flags for all error information are displayed.
22, 21	BOM[1:0]	Bus Off Recovery Mode Select b22 b21 0 0: ISO11898-1 compliant 0 1: Entry to channel halt mode automatically at bus-off entry 1 0: Entry to channel halt mode automatically at bus-off end 1 1: Entry to channel halt mode (in bus-off state) by program request
20	Reserved	These bits are read as the value after reset. The write value should be the value after reset.

Table 23.95 RSCFDnCFDCmCTR Register Contents (2/2)

Bit Position	Bit Name	Function
19	TDCVFIE	Transmitter Delay Compensation Violation Interrupt Enable 0: A transmitter delay compensation violation interrupt is disabled. 1: A transmitter delay compensation violation interrupt is enabled.
18	SOCOIE	Successful Occurrence Counter Overflow Interrupt Enable 0: A successful occurrence counter overflow interrupt is disabled. 1: A successful occurrence counter overflow interrupt is enabled.
17	EOCOIE	Error Occurrence Counter Overflow Interrupt Enable 0: An error occurrence counter overflow interrupt is disabled. 1: An error occurrence counter overflow interrupt is enabled.
16	TAIE	Transmit Abort Interrupt Enable 0: Transmit abort interrupt is disabled. 1: Transmit abort interrupt is enabled.
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt is disabled. 1: Arbitration lost interrupt is enabled.
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt is disabled. 1: Bus lock interrupt is enabled.
13	OLIE	Overload Frame Transmit Interrupt Enable 0: Overload frame transmit interrupt is disabled. 1: Overload frame transmit interrupt is enabled.
12	BORIE	Bus Off Recovery Interrupt Enable 0: Bus off recovery interrupt is disabled. 1: Bus off recovery interrupt is enabled.
11	BOEIE	Bus Off Entry Interrupt Enable 0: Bus off entry interrupt is disabled. 1: Bus off entry interrupt is enabled.
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt is disabled. 1: Error passive interrupt is enabled.
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt is disabled. 1: Error warning interrupt is enabled.
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt is disabled. 1: Bus error interrupt is enabled.
7 to 4	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
3	RTBO	Forcible Return from Bus-off When this bit is set to 1, forcible return from the bus off state is made. This bit is always read as 0.
2	CSLPR	Channel Stop Mode 0: Other than channel stop mode 1: Channel stop mode
1, 0	CHMDC[1:0]	Mode Select b1 b0 0 0: Channel communication mode 0 1: Channel reset mode 1 0: Channel halt mode 1 1: Setting prohibited

ROM Bit

When the ROM bit and the CTME bit in the RSCFDnCFDCmCTR register are set to 1, restricted operation mode is enabled. Use the restricted operation mode only when the CTMS[1:0] value in the RSCFDnCFDCmCTR register is 00_B (standard test mode). Modify this bit only in channel halt mode. This bit is always 0 in channel reset mode.

CRCT Bit

This bit is used to test the CRC generation circuit in the RS-CANFD module. Setting this bit to 1 inverts the first bit of the ID field when a message is received. With this inversion of bit, the CRC calculation result does not match the normal CRC value of the received frame, which can detect a CRC error (the CERR bit in the RSCFDnCFDCmERFL register is 1). When using this function, note the following.

- This function is available while the CTME bit in the RSCFDnCFDCmCTR register is 1 (communication test mode enabled).
- This function cannot communicate with other CAN nodes. Use this function for inter-channel communication test (the CmICBCE bit in the RSCFDnCFDGTSTCFG register is 1).
- Bit inversion in the ID field may cause bit stuffing rule violation. In that case, no CRC error is detected but a stuff error is detected.

Modify this bit only in channel halt mode. This bit is always 0 in channel reset mode.

CTMS[1:0] Bits

These bits are used to select a communication test mode. Modify these bits in channel halt mode only. These bits are set to 0 in channel reset mode.

CTME Bit

Setting this bit to 1 enables communication test mode. Modify these bits in channel halt mode. This bit is set to 0 in channel reset mode.

ERRD Bit

This bit is used to control the display mode of bits 14 to 8 in the RSCFDnCFDCmERFL register. When this bit is clear to 0, if any error is detected while the flags of bits 14-8 in the RSCFDnCFDCmERFL register are all 0, only the flags of the first error event are set to 1. If two or more errors occur in the first error event, all the flags of the detected errors are set to 1.

When this bit is set to 1, all the flags of errors that have occurred are set to 1 regardless of the error occurrence order.

Modify this bit only in channel reset mode or channel halt mode.

BOM[1:0] Bits

These bits are used to select the bus off recovery mode of the RS-CANFD module.

When the BOM[1:0] bits are set to 00_B, return from the bus off state to the error active state is compliant with the CAN specifications. That is, the RS-CANFD module reenters the CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. A bus off recovery interrupt request is generated at the time of return from the bus off state. Even if the CHMDC[1:0] bits are set to 10_B (channel halt mode) before recessive bits are detected 128 times, the RS-CANFD module does not transition to channel halt mode until recessive bits are detected 128 times.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 01_B, the CHMDC[1:0] bits in the RSCFDnCFDCmCTR register (m = 0 to 2) are set to 10_B and the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated and the TEC[7:0] and REC[7:0] bits in the RSCFDnCFDCmSTS register are cleared to 00_H.

When the RS-CANFD module reaches the bus off state when the BOM[1:0] bits are set to 10_B, the CHMDC[1:0] bits are set to 10_B and the RS-CANFD module transitions to channel halt mode after return from the bus off state (11 consecutive recessive bits are detected 128 times). A bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H.

When the BOM[1:0] bits are set to 11_B and the CHMDC[1:0] bits are set to 10_B while the RS-CANFD module is in the bus off state, the RS-CANFD module transitions to channel halt mode. No bus off recovery interrupt request is generated at the time of return from the bus off state and the TEC[7:0] and REC[7:0] bits are cleared to 00_H. However, if 11 consecutive recessive bits are detected 128 times and the RS-CANFD module has recovered to the error active state from the bus off state before the CHMDC[1:0] bits are set to 10_B, a bus off recovery interrupt request is generated.

If a program writes to the CHMDC[1:0] bit at the same time as the RS-CANFD module transition to channel halt mode (at bus off entry when the BOM[1:0] bits are 01B or at bus off end when the BOM[1:0] bits are 10_B), the program's writing takes precedence. Modify the BOM[1:0] bits only in channel reset mode.

TDCVFIE Bit

When the TDCVF flag in the RSCFDnCFDCmFDSTS register is set to 1 after the TDCVFIE bit is set to 1, an interrupt request occurs. Modify this bit only in channel reset mode.

SOCOIE Bit

When the SOCO flag in the RSCFDnCFDCmFDSTS register is set to 1 after the SOCOIE bit is set to 1, an interrupt request occurs. Modify this bit only in channel reset mode.

EOCOIE Bit

When the EOCO flag in the RSCFDnCFDCmFDSTS register is set to 1 after the EOCOIE bit is set to 1, an interrupt request occurs. Modify this bit only in channel reset mode.

TAIE Bit

When transmit abort of the transmit buffer is completed with the TAIE bit set to 1, an interrupt request is generated. Modify this bit only in channel reset mode.

ALIE Bit

When the ALF flag in the RSCFDnCFDCmERFL register is set to 1 with the ALIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BLIE Bit

When the BLF flag in the RSCFDnCFDCmERFL register is set to 1 with the BLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

OLIE Bit

When the OVLF flag in the RSCFDnCFDCmERFL register is set to 1 with the OLIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BORIE Bit

When the BORF flag in the RSCFDnCFDCmERFL register is set to 1 with the BORIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BOEIE Bit

When the BOEF flag in the RSCFDnCFDCmERFL register is set to 1 with the BOEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EPIE Bit

When the EPF flag in the RSCFDnCFDCmERFL register is set to 1 with the EPIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

EWIE Bit

When the EWF flag in the RSCFDnCFDCmERFL register is set to 1 with the EWIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

BEIE Bit

When the BEF flag in the RSCFDnCFDCmERFL register is set to 1 with the BEIE bit set to 1, an error interrupt request is generated. Modify this bit only in channel reset mode.

RTBO Bit

Setting this bit to 1 in the bus off state forcibly changes the state from the bus off state to the error active state. This bit is automatically cleared to 0. Setting this bit to 1 clears the TEC[7:0] and REC[7:0] bits in the RSCFDnCFDCmSTS register to 00_H and also clears the BOSTS flag in the RSCFDnCFDCmSTS register to 0 (not in bus off state). The other registers remain unchanged. No bus off recovery interrupt request is generated upon return from the bus off state in this case. Use this bit only when the BOM[1:0] bits in the RSCFDnCFDCmCTR register are 00_B (ISO11898-1 compliant).

A delay of up to 1 CAN bit time occurs after the RTBO bit is set to 1 until the RSCAN module transitions to the error active state. Set this bit to 1 in channel communication mode.

CSLPR Bit

Setting this bit to 1 places the channel into channel stop mode.
Clearing this bit to 0 makes the channel exit channel stop mode.
This bit should not be modified in channel communication mode or channel halt mode.

CHMDC[1:0] Bits

These bits are used to select a channel mode (channel communication mode, channel reset mode, or channel halt mode). For details, see Section 23.6.2, Channel Modes. Setting the CSLPR bit to 1 in channel reset mode allows transition to channel stop mode. Do not set the CHMDC[1:0] bits to 11_B. When the CAN module has automatically transitioned to channel halt mode based on the setting of the BOM[1:0] bits, the CHMDC[1:0] bits automatically become 10_B.

23.4.3.3 RSCFDnCFDCmSTS — Channel Status Register (m = 0 to 2)

Access: RSCFDnCFDCmSTS register can be read/written in 32-bit units
RSCFDnCFDCmSTSL register can be read/written in 16-bit units
RSCFDnCFDCmSTSH register is a read-only register that can be read in 16-bit units
RSCFDnCFDCmSTSLH register can be read/written in 8-bit units
RSCFDnCFDCmSTSLL, RSCFDnCFDCmSTSHL, RSCFDnCFDCmSTSHH registers are the read-only registers that can be read in 8-bit units

Address: RSCFDnCFDCmSTS: <RSCFDn_base> + 0008_H + (10_H × m)

RSCFDnCFDCmSTSL: <RSCFDn_base> + 0008_H + (10_H × m),
RSCFDnCFDCmSTSH: <RSCFDn_base> + $000A_H$ + (10_H × m)

RSCFDnCFDCmSTSLL: <RSCFDn_base> + 0008_H + (10_H × m),
RSCFDnCFDCmSTSLH: <RSCFDn_base> + 0009_H + (10_H × m),
RSCFDnCFDCmSTSHL: <RSCFDn_base> + $000A_H$ + (10_H × m),
RSCFDnCFDCmSTSHH: <RSCFDn_base> + $000B_H$ + (10_H × m)

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TEC[7:0]								REC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ESIF	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R/W*1	R	R	R	R	R	R	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.96 RSCFDnCFDCmSTS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	TEC[7:0]	The transmit error counter (TEC) can be read.
23 to 16	REC[7:0]	The receive error counter (REC) can be read.
15 to 9	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
8	ESIF	Error State Indication Flag 0: No CAN FD message whose ESI bit is recessive has been received. 1: At least one CAN FD message whose ESI bit is recessive has been received.
7	COMSTS	Communication Status Flag 0: Communication is not ready. 1: Communication is ready.
6	RECSTS	Receive Status Flag 0: Bus idle, in transmission or bus off state 1: In reception
5	TRMSTS	Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus off state
4	BOSTS	Bus Off Status Flag 0: Not in bus off state 1: In bus off state
3	EPSTS	Error Passive Status Flag 0: Not in error passive state 1: In error passive state

Table 23.96 RSCFDnCFDCmSTS Register Contents (2/2)

Bit Position	Bit Name	Function
2	CSLPSTS	Channel Stop Status Flag 0: Not in channel stop mode 1: In channel stop mode
1	CHLTSTS	Channel Halt Status Flag 0: Not in channel halt mode 1: In channel halt mode
0	CRSTSTS	Channel Reset Status Flag 0: Not in channel reset mode 1: In channel reset mode

TEC[7:0] Bits

These bits contain the transmit error counter value. For transmit error counter increment/decrement conditions, see the CAN specification (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

REC[7:0] Bits

These bits contain the receive error counter value. For receive error counter increment/decrement conditions, see the CAN specifications (ISO11898-1).

These bits are cleared to 0 in channel reset mode.

ESIF Flag

When the recessive ESI bit is detected in a successfully received message, this flag is set to 1. In loopback mode or mirror mode, the own transmission message is regarded as a received message. To clear this flag to 0, write 0 to this bit by the program. This bit cannot be set to 1 by the program. If the flag setting (to 1) timing matches the writing 0 (by the program) timing, this flag is set to 1.

This flag is 0 in channel reset mode.

COMSTS Flag

This bit indicates that communication is ready.

This flag becomes 1 when the CAN module has detected 11 consecutive recessive bits after it has transitioned from channel reset mode or channel halt mode to channel communication mode. This flag is cleared to 0 in channel reset mode or channel halt mode.

RECSTS Flag

This flag is set to 1 when reception has started, and is cleared to 0 when the bus has become idle or transmission has started.

TRMSTS Flag

This flag is set to 1 when transmission has started, and is cleared to 0 when the bus has become idle or reception has started. This flag remains 1 in the bus off state.

BOSTS Flag

This flag is set to 1 when the bus off state (TEC[7:0] > 255) is entered. It is cleared to 0 when the CAN module has exited the bus off state.

EPSTS Flag

This flag is set to 1 when the RS-CANFD module has entered the error passive state ((128 ≤ TEC[7:0] ≤ 255) or (128 ≤ REC[7:0])), It is cleared to 0 when the RS-CANFD module has exited the error passive state or has entered channel reset mode.

CSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel stop mode, and is cleared to 0 when the CAN module has returned from channel stop mode.

CHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel halt mode, and is cleared to 0 when the CAN module has returned from channel halt mode.

CRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to channel reset mode, and is cleared to 0 when the CAN module has transitioned to channel communication mode or channel halt mode. This flag remains 1 when the CAN module transitions from channel reset mode to channel stop mode.

23.4.3.4 RSCFDnCFDCmERFL — Channel Error Flag Register (m = 0 to 2)

Access: RSCFDnCFDCmERFL register can be read/written in 32-bit units
RSCFDnCFDCmERFLL register can be read/written in 16-bit units
RSCFDnCFDCmERFLH register is the read-only register that can be read in 16-bit units
RSCFDnCFDCmERFLLL, RSCFDnCFDCmERFLLH registers can be read/written in 8-bit units
RSCFDnCFDCmERFLHL, RSCFDnCFDCmERFLHH registers are the read-only registers that can be read in 8-bit units

Address: RSCFDnCFDCmERFL: <RSCFDn_base> + $000C_H$ + (10_H × m)

RSCFDnCFDCmERFLL: <RSCFDn_base> + $000C_H$ + (10_H × m),
RSCFDnCFDCmERFLH: <RSCFDn_base> + $000E_H$ + (10_H × m)

RSCFDnCFDCmERFLLL: <RSCFDn_base> + $000C_H$ + (10_H × m),
RSCFDnCFDCmERFLLH: <RSCFDn_base> + $000D_H$ + (10_H × m),
RSCFDnCFDCmERFLHL: <RSCFDn_base> + $000E_H$ + (10_H × m),
RSCFDnCFDCmERFLHH: <RSCFDn_base> + $000F_H$ + (10_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	CRCREG[14:0]														
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.97 RSCFDnCFDCmERFL Register Contents (1/2)

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
30 to 16	CRCREG[14:0]	CRC Calculation Data (CRC length:15 bits) A CRC value calculated based on the transmit message or receive message is indicated.
15	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
14	ADERR	ACK Delimiter Error Flag 0: No ACK delimiter error is detected. 1: ACK delimiter error is detected.
13	B0ERR	Dominant Bit Error Flag 0: No dominant bit error is detected. 1: Dominant bit error is detected.
12	B1ERR	Recessive Bit Error Flag 0: No recessive bit error is detected. 1: Recessive bit error is detected.
11	CERR	CRC Error Flag 0: No CRC error is detected. 1: CRC error is detected.
10	AERR	ACK Error Flag 0: No ACK error is detected. 1: ACK error is detected.
9	FERR	Form Error Flag 0: No form error is detected. 1: Form error is detected.

Table 23.97 RSCFDnCFDCmERFL Register Contents (2/2)

Bit Position	Bit Name	Function
8	SERR	Stuff Error Flag 0: No stuff error is detected. 1: Stuff error is detected.
7	ALF	Arbitration-lost Flag 0: No arbitration-lost is detected. 1: Arbitration-lost is detected.
6	BLF	Bus Lock Flag 0: No channel bus lock is detected. 1: Channel bus lock is detected.
5	OVLF	Overload Flag 0: No overload is detected. 1: Overload is detected.
4	BORF	Bus Off Recovery Flag 0: No bus off recovery is detected. 1: Bus off recovery is detected.
3	BOEF	Bus Off Entry Flag 0: No bus off entry is detected. 1: Bus off entry is detected.
2	EPF	Error Passive Flag 0: No error passive is detected. 1: Error passive is detected.
1	EWF	Error Warning Flag 0: No error warning is detected. 1: Error warning is detected.
0	BEF	Bus Error Flag 0: No channel bus error is detected. 1: Channel bus error is detected.

See the CAN specification (ISO11898-1) for a description of error occurrence conditions. To clear each flag of this register, the program must write a 0 to the corresponding bit. These flags cannot be set to 1 by the program. If any of these flags is set to 1 at the same time that the program writes 0 to the flag, the flag is still set to 1. The channel reset mode transition clears all of these flags to 0.

If the ERRD bit in the RSCFDnCFDCmCTR register is set to 0 (ie, only the flags for the first error event are displayed) and an error related to bits 14 to 8 of RSCFDnCFDCmERFL is detected, the flag bits are only set to 1 by the error event if bits 14 to 8 were all 0 at the time when the error occurred.

CRCREG[14:0] Flag

When the CTME bit in the RSCFDnCFDCmCTR register is set to 1 (communication test mode is enabled), if transmit or receive message is a classical CAN frame (CRC length = 15 bits), this flag is updated and the CRC value calculated based on the message can be read. When a CAN FD frame is sent or received, the value of CRCREG[20:0] bits in the RSCFDnCFDCmFDCRC register is updated. When the CTME bit is set to 0 (communication test mode is disabled), these bits are always read as 0. This bit is always 0 in channel reset mode.

ADERR Flag

This flag is set to 1 when a form error has been detected in the ACK delimiter during transmission.

B0ERR Flag

This flag is set to 1 when a recessive bit has been detected though a dominant bit was transmitted.

B1ERR Flag

This flag is set to 1 when a dominant bit has been detected though a recessive bit was transmitted.

CERR Flag

This flag is set to 1 when a CRC error has been detected.

AERR Flag

This flag is set to 1 when an ACK error has been detected.

FERR Flag

This flag is set to 1 when a form error has been detected.

SERR Flag

This flag is set to 1 when a stuff error has been detected.

ALF Flag

This flag is set to 1 when an arbitration-lost has been detected.

BLF Flag

This flag is set to 1 when 32 consecutive dominant bits have been detected on the CAN bus in channel communication mode. After that, detection of a dominant lock is restarted when either of the following conditions is met.

- A recessive bit is detected after the BLF bit has been cleared from 1 to 0.
- The CAN module transitions to channel reset mode and returns to channel communication mode after the BLF bit has been cleared from 1 to 0.

OVLF Flag

This flag is set to 1 when the overload frame transmit condition has been detected when performing reception or transmission.

BORF Flag

This flag is set to 1 when 11 consecutive recessive bits have been detected 128 times and the CAN module returns from the bus off state. However, this flag is not set to 1 if the CAN module returns from the bus off state in any of the following ways before 11 consecutive recessive bits are detected 128 times.

- The CHMDC[1:0] bits in the RSCFDnCFDCmCTR register are set to 01_B (channel reset mode).
- The RTBO bit in the RSCFDnCFDCmCTR register is set to 1 (forcible return from the bus off state is made).
- The BOM[1:0] bits in the RSCFDnCFDCmCTR register are set to 01_B (transition to channel halt mode at bus off entry).
- The CHMDC[1:0] bits in the RSCFDnCFDCmCTR register are set to 10_B (channel halt mode) before 11 consecutive recessive bits are detected 128 times with the BOM[1:0] bits set to 11_B (transition to channel halt mode upon a request from the program during bus off).

BOEF Flag

This flag is set to 1 when the bus off state is entered (TEC[7:0] value > 255). This flag is also set to 1 if the bus off state is entered when the BOM[1:0] bits in the RSCFDnCFDCmCTR register (m = 0 to 2) set to 01_B (transition to channel halt mode at bus off entry).

EPF Flag

This flag becomes 1 when the error passive state is reached ((128 ≤ TEC[7:0] ≤ 255) or (128 ≤ REC[7:0])).
This flag becomes 1 only when the REC[7:0] or TEC[7:0] value first exceeds 127. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 127, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 127 or less and then the REC[7:0] or TEC[7:0] value exceeds 127 again.

EWF Flag

This flag is set to 1 only when the REC[7:0] or TEC[7:0] value first exceeds 95. Therefore, if the program writes 0 to this flag while the value of REC[7:0] or TEC[7:0] remains over 95, this bit is not set to 1 until both REC [7:0] and TEC[7:0] values become 95 or less and then the REC[7:0] or TEC[7:0] value exceeds 95 again.

BEF Flag

This flag is set to 1 when any one of the ADERR, B0ERR, B1ERR, CERR, AERR, FERR, and SERR flags in the RSCFDnCFDCmERFL register is set to 1.

NOTE

To clear the flag of this register to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

23.4.3.5 RSCFDnCFDCmDCFG — Channel Data Bit Rate Configuration register (m = 0 to 2)

Access: RSCFDnCFDCmDCFG register can be read/written in 32-bit units
RSCFDnCFDCmDCFGL, RSCFDnCFDCmDCFGH registers can be read/written in 16-bit units
RSCFDnCFDCmDCFGLL, RSCFDnCFDCmDCFGHL, RSCFDnCFDCmDCFGHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCmDCFG: <RSCFDn_base> + 0500_H + (20_H × m)

RSCFDnCFDCmDCFGL: <RSCFDn_base> + 0500_H + (20_H × m),
RSCFDnCFDCmDCFGH: <RSCFDn_base> + 0502_H + (20_H × m)

RSCFDnCFDCmDCFGLL: <RSCFDn_base> + 0500_H + (20_H × m),
RSCFDnCFDCmDCFGHL: <RSCFDn_base> + 0502_H + (20_H × m),
RSCFDnCFDCmDCFGHH: <RSCFDn_base> + 0503_H + (20_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	DSJW [2:0]			—	DTSEG2 [2:0]			DTSEG1 [3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DBRP [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.98 RSCFDnCFDCmDCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 27	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
26 to 24	DSJW [2:0]	Data Bit Rate Resynchronization Jump Width Control b26 b25 b24 0 0 0: 1 Tq 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq
23	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
22 to 20	DTSEG2 [2:0]	Data Bit Rate Time Segment 2 Control b22 b21 b20 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq

Table 23.98 RSCFDnCFDCmDCFG Register Contents (2/2)

Bit Position	Bit Name	Function
19 to 16	DTSEG1 [3:0]	Data Bit Rate Time Segment 1 Control b19 b18 b17 b16 0 0 0 0: Setting prohibited 0 0 0 1: 2 Tq 0 0 1 0: 3 Tq 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq
15 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
7 to 0	DBRP [7:0]	Data Bit Rate Prescaler Division Ratio Setting When the set value = P (0 to 255), the data bit rate prescaler divides fCAN by (P + 1).

Modify the RSCFDnCFDCmDCFG register in channel reset mode or channel halt mode. Set this register in channel reset mode, and then transition to channel communication mode or channel halt mode. When only classical CAN frames are used in CAN FD mode, set the RSCFDnCFDCmDCFG register to the value equal to the set RSCFDnCFDCmNCFG register value. For the description and settings of bit timing parameters, see Section 23.11.1, Initial Settings..

DSJW[2:0] Bits

These bits specify the resynchronization jump width of data bit rate as a Tq value. A value of 1 to 8 Tq is settable. Specify a value equal to or smaller than the DTSEG2[2:0] bits value.

DTSEG2[2:0] Bits

These bits are used to specify a Tq value for the length of phase segment 2 (PHASE_SEG2) of nominal bit rate.

Allowed values are 2 Tq to 8 Tq, inclusive.

Set a value smaller than the value of the DTSEG1[3:0] bits.

DTSEG1[3:0] Bits

These bits specify the total length of propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1) of data bit rate as a Tq value.

A value of 2 to 16 Tq is settable.

DBRP[7:0] Bits

The clock obtained by dividing the CAN clock (fCAN) by the data bit rate prescaler ((DBRP[7:0]) + 1) becomes CANmTq(D) clock (fCANTQ(D)m). One clock of the CANmTq(D) clock becomes one Time Quantum (Tq).

Be sure to specify the same value for both NBRP[9:0] and DBRP[7:0].
To specify a different value for the nominal bit rate and the data bit rate, change the values of the

RSCFDnCFDCmNCFG.NTSEG1[6:0] and NTSEG2[4:0] bits and RSCFDnCFDCmDCFG.DTSEG1[3:0] and DTSEG2[2:0] bits, respectively.

When the TDCE bit is set to 1(Transmitter delay compensation is enabled) in the RSCFDnCFDCmFDCFG register, set the equal value of 1 or less to the bits NBRP[9:0] and DBRP[7:0].

23.4.3.6 RSCFDnCFDCmFDCFG — Channel CAN FD Configuration Register (m = 0 to 2)

Access: RSCFDnCFDCmFDCFG register can be read/written in 32-bit units
RSCFDnCFDCmFDCFGL, RSCFDnCFDCmFDCFGH registers can be read/written in 16-bit units
RSCFDnCFDCmFDCFGLL, RSCFDnCFDCmFDCFGLH, RSCFDnCFDCmFDCFGHL, RSCFDnCFDCmFDCFGHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCmFDCFG: <RSCFDn_base> + 0504_H + (20_H × m)

RSCFDnCFDCmFDCFGL: <RSCFDn_base> + 0504_H + (20_H × m),
RSCFDnCFDCmFDCFGH: <RSCFDn_base> + 0506_H + (20_H × m)

RSCFDnCFDCmFDCFGLL: <RSCFDn_base> + 0504_H + (20_H × m),
RSCFDnCFDCmFDCFGLH: <RSCFDn_base> + 0505_H + (20_H × m),
RSCFDnCFDCmFDCFGHL: <RSCFDn_base> + 0506_H + (20_H × m),
RSCFDnCFDCmFDCFGHH: <RSCFDn_base> + 0507_H + (20_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	REFE	FDOE	TMME	GWBRS	GWFDF	GWEN	—	TDCO[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ESIC	TDCE	TDCOC	—	—	—	—	—	EOCCFG[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 23.99 RSCFDnCFDCmFDCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31, 30	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
29	REFE	Reception data edge filter enable bit 0: Reception data edge filter disabled 1: Reception data edge filter enabled
28	FDOE	FD-only mode enable bit 0: FD-only mode disabled 1: FD-only mode enabled
27	TMME	Transmit Buffer Merge Mode Enable 0: Transmit buffer merge mode is disabled. 1: Transmit buffer merge mode is enabled.
26	GWBRS	Gateway BRS 0: A frame is transmitted with the BRS bit in the received frame set to 0. 1: A frame is transmitted with the BRS bit in the received frame set to 1.
25	GWFDF	Gateway FDF 0: A frame is transmitted regarding the received frame as a classical CAN frame. 1: A frame is transmitted regarding the received frame as a CAN FD frame.
24	GWEN	CAN-CAN FD Gateway Enable 0: The CAN-CAN FD gateway is disabled. 1: The CAN-CAN FD gateway is enabled.
23	Reserved	This bit is read as the value after reset. The write value should be the value after reset.
22 to 16	TDCO[6:0]	Transmitter Delay Compensation Offset These bits are set to the transmitter delay compensation offset value.
15 to 11	Reserved	These bits are read as the value after reset. The write value should be the value after reset.

Table 23.99 RSCFDnCFDCmFDCFG Register Contents (2/2)

Bit Position	Bit Name	Function
10	ESIC	Error State Display Mode Select 0: Always displays the node error state. 1: When the node is not in the error passive state: Displays the message buffer error state. When the node is in the error passive state: Displays the node error state.
9	TDCE	Transmitter delay compensation Enable 0: Transmitter delay compensation is disabled. 1: Transmitter delay compensation is enabled.
8	TDCOC	Transmitter delay compensation Measurement Select 0: Measurement and offset 1: Only offset
7 to 3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2 to 0	EOCCFG[2:0]	Error Occurrence Counting Method Select b2 b1 b0 0 0 0: All transmit messages and receive messages 0 0 1: All transmit messages 0 1 0: All receive messages 0 1 1: Setting prohibited 1 0 0: Only data phase of transmitted or received CAN FD message 1 0 1: Only data phase of transmitted CAN FD message 1 1 0: Only data phase of received CAN FD message 1 1 1: Setting prohibited

REFE bit

Setting this bit to 1 enables reception data edge filtering when the idle condition is detected, and a dominant level with less than 2 time quanta is ignored. A dominant level with more than or equal to 2 time quanta is detected as an edge. Modify this bit only in channel reset mode.

FDOE bit

Setting this bit to 1 enables FD-only mode. When data is transmitted, a CAN FD frame will be sent regardless of the settings to the CFFDF bit in the RSCFDnCFDCFFDCSTSk register or the TMFDF bit in the RSCFDnCFDTMFDCTRp register. When a Classical CAN frame is received, a form error is detected. Modify this bit only in channel reset mode.

TMME Bit

Setting this bit to 1 enables transmit buffer merge mode. Modify this bit only in channel reset mode or channel halt mode.

GWBRS Bit

When the GWEN bit is 1, the BRS bit in a CAN FD frame to be transmitted by the gateway function is set.When the GWFDF bit is set to 0, write 0 to this bit. Modify this bit only in channel reset mode.

GWFDF Bit

When the GWEN bit is 1, the FDF bit in a CAN FD frame to be transmitted by the gateway function is set. Modify this bit only in channel reset mode.

GWEN Bit

This bit is used to control the operation of the transmit/receive FIFO buffer with the CFM[1:0] bits in the RSCFDnCFDCFCCk register set to 10_B (gateway mode).

Setting this bit to 1 enables the CAN-CAN FD gateway, enabling transmission in a format different from that of frames received by the gateway function. Received frames are replaced in accordance with the settings of the GWFDF bit and the GWBRS bit. When the DLC value in the received classical CAN frame is 1001_B or more and the GWFDF bit is 1 (CAN FD frame), the DLC value is replaced with 1000_B.

While this bit is 1, do not perform routing the following frames by using the gateway function.

- CAN FD frames with a payload length of more than 8 bytes
- Remote frames

While this bit is 1, the following frame should be transmitted from the channel according to the setting of GWFDF.

- When GWFDF bit is set to 0, only classical CAN frame should be transmitted.
- When GWFDF bit is set to 1, only CAN FD frame should be transmitted.

Modify this bit only in channel reset mode.

Table 23.100 shows the settings and formats of transmit frame and receive frame while the CAN-CAN FD gateway is enabled.

Table 23.100 Operation when the CAN-CAN FD Gateway Is Enabled

Receive Frame			GWFDF Bit	Transmit Frame		
Format	BRS Bit	Received DLC Value		Format	BRS Bit	DLC Value to be Transmitted
Classical CAN	None	DLC ≤ 1000_B	0	Classical CAN	None	Not replaced
		DLC > 1000_B				
CAN FD	Arbitrary	DLC ≤ 1000_B				
Classical CAN	None	DLC ≤ 1000_B	1	CAN FD	According to GWBRS bit setting	Not replaced
		DLC > 1000_B				Replaced with 1000_B
CAN FD	Arbitrary	DLC ≤ 1000_B				Not replaced

TDCO[6:0] Bits

These bits set the SSP offset value. How to use this value depends on the TDCOC bit in the RSCFDnCFDCmFDCFG register.

These bits are based on CAN clock frequency (fCAN).

When the TDCOC bit is 0, the transmitter delay compensation result equals to the total value of the measured delay value and the TDCO[6:0] value (rounded down to the nearest integer Tq).

When the TDCOC bit is 1, the transmitter delay compensation result equals to the TDCO[6:0] value.

The SSP offset value = (set value of TDCO[6:0] bits + 1).

Modify these bits only in channel reset mode or channel halt mode.

ESIC Bit

When the ESIC bit is set to 1, if the channel is in the error active state, the ESI bit value (CFESI bit in the RSCFDnCFDCFFDCSTSk register or TMESI bit in the RSCFDnCFDTMFDCTRp register) set in

the transmit/receive FIFO buffer or transmit buffer is transmitted as an ESI bit value of the transmit message. When the channel is in the error passive state or the ESIC bit is 0, the channel status is transmitted as an ESI bit value. Modify this bit only in channel reset mode or channel halt mode.

Table 23.101 ESI Value to Be Transmitted

ESIC Bit	Channel Status	ESI Value to be Transmitted
0	Error active	0 (error active node)
	Error passive	1 (error passive node)
1	Error active	ESI value set in the transmit/receive FIFO buffer or transmit buffer (CFESI bit in the RSCFDnCFDCFFDCSTSk register or TMESI bit in the RSCFDnCFDTMFDCTRp register)
	Error passive	1 (error passive node)

TDCE Bit

Setting this bit to 1 enables transmitter delay compensation. Modify this bit only in channel reset mode or channel halt mode.

TDCOC Bit

When this bit is 0, the SSP position is defined by the total of the measured delay value and the SSP offset value (fixed value).

When this bit is 1, the SSP position is defined only by the SSP offset value.

Modify this bit only in channel reset mode or channel halt mode.

EOCCFG[2:0] Bits

These bits are used to select a frame format and a transmission/reception direction when the error occurrence counter counts CAN bus errors.

Modify these bits only in channel reset mode or channel halt mode.

23.4.3.7 RSCFDnCFDCmFDCTR — Channel CAN FD Control Register (m = 0 to 2)

Access: RSCFDnCFDCmFDCTR register can be read/written in 32-bit units
RSCFDnCFDCmFDCTRL register can be read/written in 16-bit units
RSCFDnCFDCmFDCTRLL register can be read/written in 8-bit units

Address: RSCFDnCFDCmFDCTR: <RSCFDn_base> + 0508_H + (20_H × m)

RSCFDnCFDCmFDCTRL: <RSCFDn_base> + 0508_H + (20_H × m)

RSCFDnCFDCmFDCTRLL: <RSCFDn_base> + 0508_H + (20_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOCCLR	EOCCLR
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 23.102 RSCFDnCFDCmFDCTR Register Contents

Bit Position	Bit Name	Function
31 to 2	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
1	SOCCLR	Successful Occurrence Counter Clear Setting the SOCCLR bit to 1 clears the successful occurrence counter. This bit is always read as 0.
0	EOCCLR	Error Occurrence Counter Clear Setting the EOCCLR bit to 1 clears the error occurrence counter. This bit is always read as 0.

SOCCLR Bit

Setting this bit to 1 clears the successful occurrence counter (SOC[7:0] bits in the RSCFDnCFDCmFDSTS register). This bit is automatically cleared to 0.

EOCCLR Bit

Setting this bit to 1 clears the error occurrence counter (EOC[7:0] bits in the RSCFDnCFDCmFDSTS register). This bit is automatically cleared to 0.

23.4.3.8 RSCFDnCFDCmFDSTS — Channel CAN FD Status Register (m = 0 to 2)

Access: RSCFDnCFDCmFDSTS register can be read/written in 32-bit units
RSCFDnCFDCmFDSTSL register can be read/written in 16-bit units
RSCFDnCFDCmFDSTSH register is the read-only register that can be read in 8-bit units
RSCFDnCFDCmFDSTSLH register can be read/written in 8-bit units.
RSCFDnCFDCmFDSTSLL, RSCFDnCFDCmFDSTSHL, RSCFDnCFDCmFDSTSHH register is the read-only register that can be read in 8-bit units.

Address: RSCFDnCFDCmFDSTS: <RSCFDn_base> + $050C_H$ + (20_H × m)

RSCFDnCFDCmFDSTSL: <RSCFDn_base> + $050C_H$ + (20_H × m),
RSCFDnCFDCmFDSTSH: <RSCFDn_base> + $050E_H$ + (20_H × m)

RSCFDnCFDCmFDSTSLL: <RSCFDn_base> + $050C_H$ + (20_H × m),
RSCFDnCFDCmFDSTSLH: <RSCFDn_base> + $050D_H$ + (20_H × m),
RSCFDnCFDCmFDSTSHL: <RSCFDn_base> + $050E_H$ + (20_H × m),
RSCFDnCFDCmFDSTSHH: <RSCFDn_base> + $050F_H$ + (20_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SOC[7:0]								EOC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SOCO	EOCO	TDCVF	TDCR[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R	R	R	R	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.103 RSCFDnCFDCmFDSTS Register Contents

Bit Position	Bit Name	Function
31 to 24	SOC[7:0]	Successful Occurrence Counter The successful occurrence counter value can be read.
23 to 16	EOC[7:0]	Error Occurrence Counter The error occurrence counter value can be read.
15 to 10	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
9	SOCO	Successful Occurrence Counter Overflow Flag 0: The successful occurrence counter does not overflow. 1: The successful occurrence counter has overflowed.
8	EOCO	Error Occurrence Counter Overflow Flag 0: The error occurrence counter does not overflow. 1: The error occurrence counter has overflowed.
7	TDCVF	Transmitter Delay Compensation Violation Flag 0: No transmitter delay compensation violation is present. 1: A transmitter delay compensation violation is present.
6 to 0	TDCR[6:0]	Transmitter Delay Compensation Result Status The transmitter delay compensation result can be read.

SOC[7:0] Bits

These bits show the successful occurrence counter value. The successful occurrence counter is incremented upon completion of message reception or transmission without an error. This counter stops counting when it reaches FFH. In loopback mode, this counter is incremented twice.

These bits are cleared to 0 by writing 1 to the SOCCLR bit in the RSCFDnCFDCmCTR register. These bits are 0 in channel reset mode.

EOC[7:0] Bits

These bits show the error occurrence counter value. The error occurrence counter is incremented each time an error occurs according to the condition specified by the EOCCFG[2:0] bits in the RSCFDnCFDCmFDCFG register. This counter stops counting when it reaches FFH.

These bits are cleared to 0 by writing 1 to the EOCCLR bit in the RSCFDnCFDCmCTR register. These bits are 0 in channel reset mode.

SOCO Flag

This bit indicates that successful occurrence counter overflow has occurred.

This flag is set to 1 when message reception or transmission is completed while the SOC[7:0] value has reached FF_H. This flag is 0 in channel reset mode.

EOCO Flag

This bit indicates that error occurrence counter overflow has occurred.

This flag is set to 1 when a CAN bus error is detected under the condition specified by the EOCCFG[2:0] bits in the RSCFDnCFDCmFDCFG register when the EOC[7:0] value has reached FF_H. This flag is 0 in channel reset mode.

TDCVF Flag

This bit indicates violation of transmitter delay compensation.

The transmit data is compared with the reception CAN bus level delayed due to the transceiver's loop delay. This delay changes due to physical factors such as temperature. Because the TDCR[6:0] flags are updated for each message, temporary maximum delay cannot be confirmed.

This bit is set to 1 when the transmitter delay compensation exceeds the maximum compensation 3 CANm bit times - 2 fCAN (CANm bit time is the values of data bit rate).

This flag is 0 in channel reset mode.

TDCR[6:0] Flag

These bits indicate the transmitter delay compensation result as a multiple of CAN clock frequency (fCAN).

This result depends on the settings of the TDCOC bit and TDCO[6:0] bits in the RSCFDnCFDCmFDCFG register.

This flag is updated at a falling edge between the FDF bit and res bit when the TDCE bit in the RSCFDnCFDCmFDCFG register is set to 1 and also the TDCOC bit in the RSCFDnCFDCmFDCFG register is set to 0.

This flag is 0 in channel reset mode.

NOTE

To clear the flag of this register to 0, use a store instruction to write "0" to the given flag and "1" to the other flags.

23.4.3.9 RSCFDnCFDCmFDCRC — Channel CAN FD CRC Register (m = 0 to 2)

Access: RSCFDnCFDCmFDCRC register can be read only in 32-bit units
RSCFDnCFDCmFDCRCL, RSCFDnCFDCmFDCRCH registers can be read only in 16-bit units
RSCFDnCFDCmFDCRCLL, RSCFDnCFDCmFDCRCLH, RSCFDnCFDCmFDCRCHL, RSCFDnCFDCmFDCRCHH registers can be read only in 8-bit units

Address: RSCFDnCFDCmFDCRC: <RSCFDn_base> + 0510_H + (20_H × m)

RSCFDnCFDCmFDCRCL: <RSCFDn_base> + 0510_H + (20_H × m),
RSCFDnCFDCmFDCRCH: <RSCFDn_base> + 0512_H + (20_H × m)

RSCFDnCFDCmFDCRCLL: <RSCFDn_base> + 0510_H + (20_H × m),
RSCFDnCFDCmFDCRCLH: <RSCFDn_base> + 0511_H + (20_H × m),
RSCFDnCFDCmFDCRCHL: <RSCFDn_base> + 0512_H + (20_H × m),
RSCFDnCFDCmFDCRCHH: <RSCFDn_base> + 0513_H + (20_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	SCNT[3:0]				—	—	—	CRCREG[20:16]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CRCREG[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.104 RSCFDnCFDCmFDCRC Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	These bits are read as the value after reset.
27 to 24	SCNT[3:0]	Stuff count bit Indicate a value of the stuff count in a CAN FD frame. Bits 25-27 indicates the Gray-coded value of the stuff bit count modulo 8 in the transsmitted/received frames. Bit 24 indicates an even parity value of bits 25-27.
23 to 21	Reserved	These bits are read as the value after reset.
20 to 0	CRCREG[20:0]	CRC Calculation Data (CRC Length:17 Bit or 21 Bit) These bits show the CRC value calculated based on the transmit message or receive message. When the CRC length is 17 bits, bits b20 to b17 are read as 0.

SCNT[3:0] flag

When the CTME bit in the RSCFDnCFDCmCTR register is set to 1 (communication test mode enabled), a stuff count bit value of the CAN FD frame can be read if a message transmitted/received is a CAN FD frame. When the CTME bit is 0 (communication test mode disabled), this flag is always read as 0. This flag is updated at the first bit in the CRC field of the CAN FD frame. These bits are cleared to 0 in channel reset mode.

CRCREG[20:0] Flag

When the CTME bit in the RSCFDnCFDCmCTR register is 1 (communication test mode enabled), if transmit or receive message is a CAN FD frame (CRC length = 17 or 21 bits), these flags are updated and the CRC value calculated based on the message can be read. When the CRC length of the message is 17 bits, bits b20 to b17 are always read as 0. When a classical CAN frame is transmitted or received, the CRCREG[14:0] value in the RSCFDnCFDCmERFL register is updated. When the CTME bit is 0 (communication test mode disabled), these bits are always read as 0.

These bits are cleared to 0 in channel reset mode.

23.4.4 Details of Global-Related Registers

23.4.4.1 RSCFDnCFDGCFG — Global Configuration Register

Access: RSCFDnCFDGCFG register can be read/written in 32-bit units
RSCFDnCFDGCFGL, RSCFDnCFDGCFGH registers can be read/written in 16-bit units
RSCFDnCFDGCFGLL, RSCFDnCFDGCFGLH, RSCFDnCFDGCFGHL, RSCFDnCFDGCFGHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGCFG: <RSCFDn_base> + 0084_H

RSCFDnCFDGCFGL: <RSCFDn_base> + 0084_H,
RSCFDnCFDGCFGH: <RSCFDn_base> + 0086_H

RSCFDnCFDGCFGLL: <RSCFDn_base> + 0084_H,
RSCFDnCFDGCFGLH: <RSCFDn_base> + 0085_H,
RSCFDnCFDGCFGHL: <RSCFDn_base> + 0086_H,
RSCFDnCFDGCFGHH: <RSCFDn_base> + 0087_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ITRCP[15:0}															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSBTCS[2:0]			TSSS	TSP[3:0]				—	—	CMPOC	DCS	MME	DRE	DCE	TPRI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.105 RSCFDnCFDGCFG Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	ITRCP[15:0]	Interval Timer Prescaler Set When these bits are set to M, the pclk is divided by M. Setting 0000_H is prohibited when the interval timer is in use.
15 to 13	TSBTCS[2:0]	Timestamp Clock Source Select b15 b14 b13 0 0 0: Channel 0 nominal bit time clock 0 0 1: Channel 1 nominal bit time clock 0 1 0: Channel 2 nominal bit time clock 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Setting prohibited
12	TSSS	Timestamp Source Select 0: pclk/2*1 1: Nominal bit time clock

Table 23.105 RSCFDnCFDGCFG Register Contents (2/2)

Bit Position	Bit Name	Function
11 to 8	TSP[3:0]	Timestamp Clock Source Division b11 b10 b9 b8 0 0 0 0: Not divided 0 0 0 1: Divided by 2 0 0 1 0: Divided by 4 0 0 1 1: Divided by 8 0 1 0 0: Divided by 16 0 1 0 1: Divided by 32 0 1 1 0: Divided by 64 0 1 1 1: Divided by 128 1 0 0 0: Divided by 256 1 0 0 1: Divided by 512 1 0 1 0: Divided by 1024 1 0 1 1: Divided by 2048 1 1 0 0: Divided by 4096 1 1 0 1: Divided by 8192 1 1 1 0: Divided by 16384 1 1 1 1: Divided by 32768
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	CMPOC	Payload Overflow Mode Select 0: No message is stored. 1: Messages are stored and payloads exceeding the buffer size are discarded.
4	DCS	CAN Clock Source Select*2 0: clkc 1: clk_xincan
3	MME	Mirror Function Enable 0: Mirror function is disabled. 1: Mirror function is enabled.
2	DRE	DLC Replacement Enable 0: DLC replacement is disabled. 1: DLC replacement is enabled.
1	DCE	DLC Check Enable 0: DLC check is disabled. 1: DLC check is enabled.
0	TPRI	Transmit Priority Select 0: ID priority 1: Transmit buffer number priority

Note 1. When specifying pclk/2 as the timestamp counter count source, set bits TSBTCS[2:0] to 000B.

Note 2. For the CAN clock frequency settings, see Table 23.8, Setting Range example of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/D1L/D1M.

Modify the RSCFDnCFDGCFG register only in global reset mode.

ITRCP[15:0] Bits

These bits are used to set a clock source division value of the interval timer for FIFO buffers. See Section 23.8.3.1, Interval Transmission Function.

TSBTCS[2:0] Bits

When the TSSS bit is 1, these bits are used to select the channel of the nominal bit time clock that will be the clock source of the timestamp counter. However, do not select the channel that handles the CAN FD frames.

TSSS Bit

This bit is used to select a clock source of the timestamp counter. Select pclk if there is no channel that handles only classical CAN frames.

TSP[3:0] Bits

A clock obtained by dividing the clock source selected with the TSBTCS[2:0] bits and TSSS bit according to the TSP[3:0] bits is used as the timestamp counter count source.

CMPOC Bit

This bit is used to select operation in case the payload length of received message exceeds the payload storage size of the storage buffer.

When this bit is 0, the received message in which the payload overflows is not stored in the buffer.

When this bit is 1, the received message in which the payload overflows is stored in the buffer, and depending on the DRE bit the received DLC value or the DLC value of the receive rule is stored in the buffer. At this time, payloads exceeding the buffer's payload storage size are discarded.

The buffer’s payload storage size is set by the following bits.

- Receive buffer: RMPLS[1:0] bits in the RSCFDnCFDRMNB register
- Receive FIFO buffer: RFPLS[2:0] bits in the RSCFDnCFDRFCCx register
- Transmit/receive FIFO buffer: CFPLS[2:0] bits in the RSCFDnCFDCFCCk register

DCS Bit

When this bit is set to 0, clkc is used as the clock source of the CAN clock (fCAN).

When this bit is set to 1, clk_xincan is used as the clock source of the CAN clock (fCAN).

For the CAN clock frequency settings, see Table 23.8, Setting Range example of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/D1L/D1M.

MME Bit

Setting this bit to 1 makes the mirror function available.

DRE Bit

When the DRE bit is set to 1, the DLC value of the receive rule is stored in the buffer instead of the DLC value of the received message after the DLC value has passed through the DLC filter. In this case, a value of 00_H is stored in each data byte beyond the DLC value of the receive rule.

The DLC replacement function is only available when the DCE bit is set to 1 (DLC check is enabled).

DCE Bit

Setting this bit to 1 makes the DLC check function available. When disabling the DLC check function, set the GAFLDLC[3:0] bits in the RSCFDnCFDGAFLP0_j register to 0000_B before clearing the DCE bit in the RSCFDnCFDGCFG register to 0.

TPRI Bit

This bit is used to set the transmit priority.

When this bit is set to 0, ID priority is selected and the transmit priority complies with the CAN bus arbitration rule (ISO11898-1 specifications). When this bit is set to 1, transmit buffer number priority is selected and the transmit buffer with the smallest number of those has the highest priority.

While the transmit queue is in use, this bit should be set to 0.

23.4.4.2 RSCFDnCFDGCTR — Global Control Register

Access: RSCFDnCFDGCTR register can be read/written in 32-bit units
RSCFDnCFDGCTRL, RSCFDnCFDGCTRH registers can be read/written in 16-bit units
RSCFDnCFDGCTRLL, RSCFDnCFDGCTRLH, RSCFDnCFDGCTRHL registers can be read/written in 8-bit units

Address: RSCFDnCFDGCTR: <RSCFDn_base> + 0088_H

RSCFDnCFDGCTRL: <RSCFDn_base> + 0088_H,
RSCFDnCFDGCTRH: <RSCFDn_base> + $008A_H$

RSCFDnCFDGCTRLL: <RSCFDn_base> + 0088_H,
RSCFDnCFDGCTRLH: <RSCFDn_base> + 0089_H,
RSCFDnCFDGCTRHL: <RSCFDn_base> + $008A_H$

Value after reset: $0000\ 0005_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	CMPOFIE	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 23.106 RSCFDnCFDGCTR Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
16	TSRST	Timestamp Counter Reset Setting the TSRST bit to 1 resets the timestamp counter. This bit is always read as 0.
15 to 12	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
11	CMPOFIE	Payload Overflow Interrupt Enable 0: A payload overflow interrupt is disabled. 1: A payload overflow interrupt is enabled.
10	THLEIE	Transmit History Buffer Overflow Interrupt Enable 0: Transmit history buffer overflow interrupt is disabled. 1: Transmit history buffer overflow interrupt is enabled.
9	MEIE	FIFO Message Lost Interrupt Enable 0: FIFO message lost interrupt is disabled. 1: FIFO message lost interrupt is enabled.
8	DEIE	DLC Error Interrupt Enable 0: DLC error interrupt is disabled. 1: DLC error interrupt is enabled.
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	GSLPR	Global Stop Mode 0: Other than global stop mode 1: Global stop mode
1, 0	GMDC[1:0]	Global Mode Select b1 b0 0 0: Global operating mode 0 1: Global reset mode 1 0: Global test mode 1 1: Setting prohibited

TSRST Bit

This bit is used to reset the timestamp counter. When this bit is set to 1, the RSCFDnCFDGTSC register is cleared to 0000_H.

CMPOFIE Bit

When the CMPOF flag in the RSCFDnCFDGERFL register is set to 1 after the CMPOFIE bit is set to 1, an interrupt request occurs. Modify this bit only in global reset mode.

THLEIE Bit

When the THLEIE bit is set to 1 and the THLES flag in the RSCFDnCFDGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

MEIE Bit

When the MEIE bit is set to 1 and the MES flag in the RSCFDnCFDGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

DEIE Bit

When the DEIE bit is set to 1 and the DEF flag in the RSCFDnCFDGERFL register is set to 1, an interrupt request is generated. Modify this bit only in global reset mode.

GSLPR Bit

Setting this bit to 1 places the RSCAN module into global stop mode.
Clearing this bit to 0 makes the RSCAN module leave from global stop mode.
This bit should not be modified in global operating mode or global test mode.

GMDC[1:0] Bits

These bits are used to select the mode of entire RS-CANFD module (global operating mode, global reset mode, or global test mode). For details, see Section 23.6.1, Global Modes. Setting the GSLPR bit to 1 when in global reset mode places the RS-CANFD module into global stop mode.

23.4.4.3 RSCFDnCFDGSTS — Global Status Register

Access: RSCFDnCFDGSTS register can be read only in 32-bit units
RSCFDnCFDGSTSL register is a read-only register that can be read only in 16-bit units
RSCFDnCFDGSTSLL register is a read-only register that can be read only in 8-bit units

Address: RSCFDnCFDGSTS: <RSCFDn_base> + $008C_H$

RSCFDnCFDGSTSL: <RSCFDn_base> + $008C_H$

RSCFDnCFDGSTSLL: <RSCFDn_base> + $008C_H$

Value after reset: $0000\ 000D_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	GRAMINIT	GSLPSTS	GHLTSTS	GRSTSTS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.107 RSCFDnCFDGSTS Register Contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, the value after reset is returned.
3	GRAMINIT	CAN RAM Initialization Status Flag 0: CAN RAM initialization is completed. 1: CAN RAM initialization is ongoing.
2	GSLPSTS	Global Stop Status Flag 0: Not in global stop mode 1: In global stop mode
1	GHLTSTS	Global Test Status Flag 0: Not in global test mode 1: In global test mode
0	GRSTSTS	Global Reset Status Flag 0: Not in global reset mode 1: In global reset mode

GRAMINIT Flag

This flag indicates the initialization status of the CAN RAM.

This flag is set to 1 after the MCU has been reset, and is cleared to 0 when CAN RAM initialization is completed.

GSLPSTS Flag

This flag is set to 1 when the CAN module has transitioned to global stop mode, and is cleared to 0 when the CAN module has returned from global stop mode.

GHLTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global test mode, and is cleared to 0 when the CAN module has exited global test mode.

GRSTSTS Flag

This flag is set to 1 when the CAN module has transitioned to global reset mode, and is cleared to 0 when the CAN module has exited global reset mode. This flag remains 1 even when the CAN module has transitioned from global reset mode to global stop mode.

23.4.4.4 RSCFDnCFDGERFL — Global Error Flag Register

Access: RSCFDnCFDGERFL register can be read/written in 32-bit units
RSCFDnCFDGERFLL, RSCFDnCFDGERFLH registers can be read/written in 16-bit units
RSCFDnCFDGERFLLL, RSCFDnCFDGERFLHL registers can be read/written in 8-bit units

Address: RSCFDnCFDGERFL: <RSCFDn_base> + 0090_H

RSCFDnCFDGERFLL: <RSCFDn_base> + 0090_H,
RSCFDnCFDGERFLH: <RSCFDn_base> + 0092_H

RSCFDnCFDGERFLLL: <RSCFDn_base> + 0090_H
RSCFDnCFDGERFLHL: <RSCFDn_base> + 0092_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	EEF2	EEF1	EEF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	CMPOF	THLES	MES	DEF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R	R	R/W*1

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.108 RSCFDnCFDGERFL Register Contents

Bit Position	Bit Name	Function
31 to 19, 15, 14, 7, 6, 4	Reserved	When read, the value after reset is returned. When writing these bits, write the value after reset.
18	EEF2	ECC Error Flag for Channel 2 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
17	EEF1	ECC Error Flag for Channel 1 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
16	EEF0	ECC Error Flag for Channel 0 0: No 2-bit ECC error when deciding transmission priority 1: A 2-bit ECC error when deciding transmission priority
13 to 8, 5	Reserved	When read, the undefined value is returned. When writing these bits, write the value after reset.
3	CMPOF	Payload Overflow Flag 0: No payload overflow has occurred. 1: A payload overflow has occurred.
2	THLES	Transmit History Buffer Overflow Status Flag 0: No transmit history buffer overflow has occurred. 1: A transmit history buffer overflow has occurred.
1	MES	FIFO Message Lost Status Flag 0: No FIFO message lost error has occurred. 1: A FIFO message lost error has occurred.
0	DEF	DLC Error Flag 0: No DLC error has occurred. 1: A DLC error has occurred.

All flags in the RSCFDnCFDGERFL register are cleared to 0 in global reset mode.

EEFm Flag

When a 2-bit ECC error is detected during the transmission priority determination of channel m (m = 0 to 2), the EEFm flag is set to 1, disabling message transmission. This flag can be cleared to 0 by writing 0 by the program.

CMPOF Flag

When a payload overflow occurs in any of channel m (m = 0 to 2), the CMPOF flag is set to 1. This flag can be cleared to 0 by writing 0 to this bit by the program.

THLES Flag

The THLES flag is set to 1 when any one of the THLELT flags in the RSCFDnCFDTHLSTSm register (m = 0 to 2) is set to 1.

This flag is cleared to 0 when the THLELT flags of all channels are set to 0.

MES Flag

The MES flag is set to 1 when any one of the RFMLT flags in the RSCFDnCFDRFSTSx register (x = 0 to 7) or the CFMLT flags in the RSCFDnCFDCFSTSk register (k = 0 to 8) is set to 1.

This flag is cleared to 0 when all RFMLT flags and CFMLT flags are set to 0.

DEF Flag

The DEF flag is set to 1 when an error has been detected during the DLC check. The program can clear this flag by writing 0 to this bit.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

NOTE

To clear the flag of this register to 0, use a store instruction to write 0 to the given flag and 1 to the other flags.

23.4.4.5 RSCFDnCFDGTSC — Global Timestamp Counter Register

Access: RSCFDnCFDGTSC register is a read-only register that can be read only in 32-bit units.
RSCFDnCFDGTSCL register is a read-only register that can be read only in 16-bit units.

Address: RSCFDnCFDGTSC: <RSCFDn_base> + 0094_H

RSCFDnCFDGTSCL: <RSCFDn_base> + 0094_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.109 RSCFDnCFDGTSC Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset.
15 to 0	TS[15:0]	Timestamp Value The timestamp counter value can be read. Counter Value: 0000_H to $FFFF_H$

TS[15:0] Bits

When the TS[15:0] bits are read, the read value shows the timestamp counter (16-bit free-running counter) value at that time. When the SOF is detected, the TS[15:0] value is captured and later stored in the receive buffer or the FIFO buffer. Furthermore, the TS[15:0] value is stored in the transmit history buffer. The timestamp counter is initialized in global reset mode.

The timestamp counter starts and stops counting differently, depending on the count source.

- When the TSSS bit in the RSCFDnCFDGCFG register is 0 (pclk):
  The timestamp counter starts counting when the RSCAN module has transitioned to global operating mode.
  This counter stops counting when the RSCAN module has transitioned to global stop mode or global test mode.
- When the TSSS bit is 1 (CANm nominal bit time clock):
  The timestamp counter starts counting when the corresponding channel has transitioned to channel communication mode.
  This counter stops counting when the corresponding channel has transitioned to channel reset mode or channel halt mode.

23.4.4.6 RSCFDnCFDGTINTSTS0 — Global TX Interrupt Status Register 0

Access: RSCFDnCFDGTINTSTS0 register can be read only in 32-bit units
RSCFDnCFDGTINTSTS0L, RSCFDnCFDGTINTSTS0H registers can be read only in 16-bit units
RSCFDnCFDGTINTSTS0LL, RSCFDnCFDGTINTSTS0LH, RSCFDnCFDGTINTSTS0HL registers can be read only in 8-bit units

Address: RSCFDnCFDGTINTSTS0: <RSCFDn_base> + 0460_H

RSCFDnCFDGTINTSTS0L: <RSCFDn_base> + 0460_H,
RSCFDnCFDGTINTSTS0H: <RSCFDn_base> + 0462_H

RSCFDnCFDGTINTSTS0LL: <RSCFDn_base> + 0460_H,
RSCFDnCFDGTINTSTS0LH: <RSCFDn_base> + 0461_H,
RSCFDnCFDGTINTSTS0HL: <RSCFDn_base> + 0462_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	THIF2	CFTIF2	TQIF2	TAIF2	TSIF2
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R*1	R*1	R*1	R*1	R*1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	—	—	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R*1	R*1	R*1	R*1	R*1	R	R	R	R*1	R*1	R*1	R*1	R*1

Note 1. This bit is automatically cleared in the global reset or channel reset mode.

Table 23.110 RSCFDnCFDGTINTSTS0 Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 21	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
20	THIF2	Channel 2 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
19	CFTIF2	Channel 2 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
18	TQIF2	Channel 2 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
17	TAIF2	Channel 2 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
16	TSIF2	Channel 2 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
15 to 13	Reserved	These bits are read as the value after reset.
12	THIF1	Channel 1 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
11	CFTIF1	Channel 1 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
10	TQIF1	Channel 1 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.

Table 23.110 RSCFDnCFDGTINTSTS0 Register Contents (2/2)

Bit Position	Bit Name	Function
9	TAIF1	Channel 1 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
8	TSIF1	Channel 1 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.
7 to 5	Reserved	These bits are read as the value after reset.
4	THIF0	Channel 0 Transmit History Interrupt Status Flag 0: Transmit history interrupt is not requested. 1: Transmit history interrupt is requested.
3	CFTIF0	Channel 0 Transmit/receive FIFO Transmit Interrupt Status Flag 0: Transmit/receive FIFO transmit interrupt is not requested. 1: Transmit/receive FIFO transmit interrupt is requested.
2	TQIF0	Channel 0 Transmit Queue Interrupt Status Flag 0: Transmit queue interrupt is not requested. 1: Transmit queue interrupt is requested.
1	TAIF0	Channel 0 Transmit Buffer Abort Interrupt Status Flag 0: Transmit buffer abort interrupt is not requested. 1: Transmit buffer abort interrupt is requested.
0	TSIF0	Channel 0 Transmit Buffer Interrupt Status Flag 0: Transmit buffer transmit complete interrupt is not requested. 1: Transmit buffer transmit complete interrupt is requested.

TSIFm Bits

The TSIFm bit is set to 1 when the TMIEp bit in the RSCFDnCFDTMIECy register is set to 1 (transmit buffer interrupt enabled) and the TMTRF[1:0] flags in the RSCFDnCFDTMSTSp register are set to 10_B (transmit completed without abort request) or 11_B (transmit completed with abort request).

When the TMTRF[1:0] flags are cleared to 00_B under the condition that the TSIFm bit can be set to 1, this flag is cleared to 0. In addition, clearing the TMIEp bit to 0 also clears this flag to 0.

TAIFm Bits

The TAIFm bit is set to 1 when the TAIE bit in the RSCFDnCFDCmCTR register is 1 (transmit abort interrupt enabled) and the TMTRF[1:0] flags in the RSCFDnCFDTMSTSp register are set to 01_B (transmit abort completed).

This flag is cleared to 0 when the TMTRF[1:0] flags are cleared to 00_B after the transmit abort is completed.

TQIFm Bits

When the TXQIE bit in the RSCFDnCFDTXQCCm register is set to 1 (transmit queue interrupt enabled) and the TXQIF bit in the RSCFDnCFDTXQSTSm register is set to 1 (transmit queue interrupt request), the TQIFm bit is set to 1.

When the TXQIF bit (transmit queue interrupt request) in the RSCFDnCFDTXQSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the TXQIE bit is cleared to 0.

CFTIFm Bits

When the CFTXIE bit in the RSCFDnCFDCFCCk register is set to 1 (transmit/receive FIFO transmit interrupt enabled) and the CFTXIF bit in the RSCFDnCFDCFSTSk register is set to 1 (transmit/receive FIFO transmit interrupt request), the CFTIFm bit is set to 1.

When the CFTXIF bit is cleared to 0 under the conditions that the CFTIFm bit can be set to 1, this bit is cleared to 0. This flag is also cleared to 0 when the CFTXIE bit is cleared to 0.

THIFm Bits

When the THLIE bit in the RSCFDnCFDTHLCCm register is set to 1 (transmit history interrupt enabled) and the THLIF bit in the RSCFDnCFDTHLSTSm register is set to 1 (transmit history interrupt request), the THIFm bit is set to 1.

When the THLIF bit in the RSCFDnCFDTHLSTSm register is cleared to 0, this bit is cleared to 0. This flag is also cleared to 0 when the THLIE bit is cleared to 0.

23.4.4.7 RSCFDnCFDGFDCFG — Global FD configuration register

Access: RSCFDnCFDGFDCFG register can be read/written in 32-bit unit
RSCFDnCFDGFDCFGL register can be read/written in 16-bit unit
RSCFDnCFDGFDCFGLL, RSCFDnCFDGFDCFGLH registers can be read/written in 8-bit unit

Address: RSCFDnCFDGFDCFG: <RSCFDn_base> + 0474_H

RSCFDnCFDGFDCFG L: <RSCFDn_base> + 0474_H,

RSCFDnCFDGFDCFG LL: <RSCFDn_base> + 0474_H,
RSCFDnCFDGFDCFG LH: <RSCFDn_base> + 0475_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TSCCFG[1:0]		—	—	—	—	—	—	—	RPED
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 23.111 RSCFDnCFDGFDCFG Register Contents

Bit Position	Bit Name	Function
31 to 10	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
9, 8	TSCCFG[1:0]	Time-stamp capture setting bit b9 b8 0 0: Captured at a sample point in the SOF bit. 0 1: Captured when a valid frame has been transmitted/received. 1 0: Captured at a sample point of the res bit.*1 1 1: Setting prohibited.
7 to 1	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
0	RPED	Protocol exception event detection disabled bit 0: Protocol exception event detection enabled 1: Protocol exception event detection disabled

Note 1. When a Classical CAN frame is transmitted/received, a time-stamp value will be captured at the sample point in the SOF bit.

TSCCFG[1:0] bit

Select a point where a time-stamp value is captured. Modify this bit only in global reset mode.

RPED bit

Setting this bit to 1 disables the protocol exception event detection. When a protocol exception event is detected while this bit is set to 1, the event is regarded as a form error and an error frame will be output. Modify this bit only in global reset mode.

23.4.4.8 RSCFDnCFDGCRCCFG —Global CRC configuration register

Access: RSCFDnCFDGCRCCFG register can be read/written in 32-bit unit
RSCFDnCFDGCRCCFGL register can be read/written in 16-bit unit
RSCFDnCFDGCRCCFGLL register can be read/written in 8-bit unit

Address: RSCFDnCFDGCRCCFG: <RSCFDn_base> + 0478_H

RSCFDnCFDGCRCCFG L: <RSCFDn_base> + 0478_H

RSCFDnCFDGCRCCFG LL: <RSCFDn_base> + 0478_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 23.112 RSCFDnCFDGCRCCFG Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
0	NIE	ISO11898-1 protocol switchover bit 0: Support CAN FD according to the ISO 11898-1 2015 protocol. 1: Support CAN FD ISO/DIS 11898-1 (August 12, 2014 version) protocol. (non-ISO 11898-1 2015 compliant)

NIE bit

Setting this bit to 1 supports CAN FD ISO/DIS 11898-1 (August 12, 2014 version). This setting is not based on the ISO 11898-1 2015 protocol. Modify this bit only in global reset mode.

This bit is cleared to 0 when the RCMC bit in the RSCFDnCFDGRMCFG register is 0 (Classical CAN mode is selected.)

23.4.5 Details of Receive Rule-related Registers

23.4.5.1 RSCFDnCFDGAFLECTR — Receive Rule Entry Control Register

Access: RSCFDnCFDGAFLECTR register can be read/written in 32-bit units
RSCFDnCFDGAFLECTRL register can be read/written in 16-bit units
RSCFDnCFDGAFLECTRLL, RSCFDnCFDGAFLECTRLH registers can be read/written in 8-bit units

Address: RSCFDnCFDGAFLECTR: <RSCFDn_base> + 0098_H

RSCFDnCFDGAFLECTRL: <RSCFDn_base> + 0098_H

RSCFDnCFDGAFLECTRLL: <RSCFDn_base> + 0098_H,
RSCFDnCFDGAFLECTRLH: <RSCFDn_base> + 0099_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	AFLDAE	—	—	—	AFLPN[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 23.113 RSCFDnCFDGAFLECTR Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
8	AFLDAE	Receive Rule Table Write Enable 0: Receive rule table write is disabled. 1: Receive rule table write is enabled.
7 to 5	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
4 to 0	AFLPN[4:0]	Receive Rule Table Page Number Configuration A page number can be selected from a range of page 0 (00000_B) to page 11 (01011_B).

AFLDAE Bit

Setting this bit to 0 disables the write to the receive rule table. After writes to the receive rule table are completed, set this bit to 0 to disable the write to the table. The receive rule table can be read regardless of the value of this bit.

Set the AFLDAE bit to 1 only in global reset mode.

AFLPN[4:0] Bits

These bits are used to set the page number of the receive rule table. Sixteen receive rules can be set per page.

Set these bits to a value within the range of 00000_B to 01011_B.

23.4.5.2 RSCFDnCFDGAFLCFG0 — Receive Rule Configuration Register 0

Access: RSCFDnCFDGAFLCFG0 register can be read/written in 32-bit units
RSCFDnCFDGAFLCFG0L, RSCFDnCFDGAFLCFG0H registers can be read/written in 16-bit units
RSCFDnCFDGAFLCFG0LH, RSCFDnCFDGAFLCFG0HL, RSCFDnCFDGAFLCFG0HH registers can be read/written in 8-bit units

Address: RSCFDnCFDGAFLCFG0: <RSCFDn_base> + $009C_H$

RSCFDnCFDGAFLCFG0L: <RSCFDn_base> + $009C_H$, RSCFDnCFDGAFLCFG0H: <RSCFDn_base> + $009E_H$

RSCFDnCFDGAFLCFG0LH: <RSCFDn_base> + $009D_H$,
RSCFDnCFDGAFLCFG0HL: <RSCFDn_base> + $009E_H$,
RSCFDnCFDGAFLCFG0HH: <RSCFDn_base> + $009F_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RNC0[7:0]								RNC1[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RNC2[7:0]								—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 23.114 RSCFDnCFDGAFLCFG0 Register Contents

Bit Position	Bit Name	Function
31 to 24	RNC0[7:0]	Number of Rules for Channel 0 Set the number of receive rules exclusively used for channel 0.
23 to 16	RNC1[7:0]	Number of Rules for Channel 1 Set the number of receive rules exclusively used for channel 1.
15 to 8	RNC2[7:0]	Number of Rules for Channel 2 Set the number of receive rules exclusively used for channel 2.
7 to 0	Reserved	These bits are read as the value after reset. The write value should be the value after reset.

Modify the RSCFDnCFDGAFLCFG0 register only in global reset mode.

Up to 64 × (number of channels) rules can be registered in the receive rule table for the entire unit. The number of receive rules per channel should meet the following conditions.

- The maximum number of rules per channel is 128.
- The total of the number of rules allocated to each channel is not larger than the number of rules that can be registered for the entire unit.

RNC0[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 0 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC1[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 1 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

RNC2[7:0] Bits

These bits are used to set the number of rules to be registered in the channel 2 receive rule table.

Set these bits to a value within the range of 00_H to 80_H.

23.4.5.3 RSCFDnCFDGAFLIDj — Receive Rule ID Register (j = 0 to 15)

Access: RSCFDnCFDGAFLIDj register can be read/written in 32-bit units
RSCFDnCFDGAFLIDjL, RSCFDnCFDGAFLIDjH registers can be read/written in 16-bit units
RSCFDnCFDGAFLIDjLL, RSCFDnCFDGAFLIDjLH, RSCFDnCFDGAFLIDjHL, RSCFDnCFDGAFLIDjHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGAFLIDj: <RSCFDn_base> + 1000_H + (10_H × j)

RSCFDnCFDGAFLIDjL: <RSCFDn_base> + 1000_H + (10_H × j),
RSCFDnCFDGAFLIDjH: <RSCFDn_base> + 1002_H + (10_H × j)

RSCFDnCFDGAFLIDjLL: <RSCFDn_base> + 1000_H + (10_H × j),
RSCFDnCFDGAFLIDjLH: <RSCFDn_base> + 1001_H + (10_H × j),
RSCFDnCFDGAFLIDjHL: <RSCFDn_base> + 1002_H + (10_H × j),
RSCFDnCFDGAFLIDjHH: <RSCFDn_base> + 1003_H + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.115 RSCFDnCFDGAFLIDj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDE	IDE Select 0: Standard ID 1: Extended ID
30	GAFLRTR	RTR Select 0: Data frame 1: Remote frame
29	GAFLLB	Receive Rule Target Message Select 0: When a message transmitted from another CAN node is received 1: When the own transmitted message is received
28 to 0	GAFLID[28:0]	ID Set the ID of the receive rule. For the standard ID, set the ID in bits b10 to b0 and set bits b28 to b11 to 0.

Modify the RSCFDnCFDGAFLIDj register when the AFLDAE bit in the RSCFDnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDE Bit

This bit is used to select the ID format (standard ID or extended ID) of the receive rule. This bit is compared with the IDE bit in the received message during the acceptance filter processing.

GAFLRTR Bit

This bit is used to select the frame format (data frame or remote frame) of the receive rule. This bit is compared with the RTR bit in the received message during the acceptance filter processing.

GAFLLB Bit

When this bit is set to 0, data processing using the receive rule is performed when receiving messages transmitted from another CAN node.

When this bit is set to 1 when the mirror function is used, data processing using the receive rule is performed when the CAN node is receiving its own transmitted messages.

GAFLID[28:0] Bits

These bits are used to set the ID field of the receive rule. The ID value set by these bits is compared with the ID of the received message during the acceptance filter processing.

23.4.5.4 RSCFDnCFDGAFLMj — Receive Rule Mask Register (j = 0 to 15)

Access: RSCFDnCFDGAFLMj register can be read/written in 32-bit units
RSCFDnCFDGAFLMjL, RSCFDnCFDGAFLMjH registers can be read/written in 16-bit units
RSCFDnCFDGAFLMjLL, RSCFDnCFDGAFLMjLH, RSCFDnCFDGAFLMjHL, RSCFDnCFDGAFLMjHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGAFLMj: <RSCFDn_base> + 1004_H + (10_H × j)

RSCFDnCFDGAFLMjL: <RSCFDn_base> + 1004_H + (10_H × j),
RSCFDnCFDGAFLMjH: <RSCFDn_base> + 1006_H + (10_H × j)

RSCFDnCFDGAFLMjLL: <RSCFDn_base> + 1004_H + (10_H × j),
RSCFDnCFDGAFLMjLH: <RSCFDn_base> + 1005_H + (10_H × j),
RSCFDnCFDGAFLMjHL: <RSCFDn_base> + 1006_H + (10_H × j),
RSCFDnCFDGAFLMjHH: <RSCFDn_base> + 1007_H + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLIDEM	GAFLRTRM	—	GAFLIDM[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLIDM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.116 RSCFDnCFDGAFLMj Register Contents

Bit Position	Bit Name	Function
31	GAFLIDEM	IDE Mask 0: The IDE bit is not compared. 1: The IDE bit is compared.
30	GAFLRTRM	RTR Mask 0: The RTR bit is not compared. 1: The RTR bit is compared
29	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
28 to 0	GAFLIDM[28:0]	ID Mask 0: The corresponding ID bit is not compared. 1: The corresponding ID bit is compared.

Modify the RSCFDnCFDGAFLMj register when the AFLDAE bit in the RSCFDnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLIDEM Bit

When this bit is set to 1, filter processing is performed only for messages of the ID format specified by the GAFLIDE bit in the RSCFDnCFDGAFLIDj register.

When this bit is cleared to 0, the IDs of all the receive messages and the specified IDs are regarded as matched. To set the GAFLIDEM bit to 0, set the GAFLIDM[28:0] bits to all 0 at the same time.

GAFLRTRM Bit

This bit is used to mask the RTR bit of the receive rule.

GAFLIDM[28:0] Bits

These bits are used to mask the corresponding ID bit of the receive rule.

23.4.5.5 RSCFDnCFDGAFLP0_j — Receive Rule Pointer 0 Register (j = 0 to 15)

Access: RSCFDnCFDGAFLP0_j register can be read/written in 32-bit units
RSCFDnCFDGAFLP0_jL, RSCFDnCFDGAFLP0_jH registers can be read/written in 16-bit units
RSCFDnCFDGAFLP0_jLH, RSCFDnCFDGAFLP0_jHL, RSCFDnCFDGAFLP0_jHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGAFLP0_j: <RSCFDn_base> + 1008_H + (10_H × j)

RSCFDnCFDGAFLP0_jL: <RSCFDn_base> + 1008_H + (10_H × j),
RSCFDnCFDGAFLP0_jH: <RSCFDn_base> + $100A_H$ + (10_H × j)

RSCFDnCFDGAFLP0_jLH: <RSCFDn_base> + 1009_H + (10_H × j),
RSCFDnCFDGAFLP0_jHL: <RSCFDn_base> + $100A_H$ + (10_H × j),
RSCFDnCFDGAFLP0_jHH: <RSCFDn_base> + $100B_H$ + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAFLDLC[3:0]				GAFLPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLRMV	GAFLRMDP[6:0]							—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 23.117 RSCFDnCFDGAFLP0_j Register Contents

Bit Position	Bit Name	Function
31 to 28	GAFLDLC[3:0]	Receive Rule DLC
27 to 16	GAFLPTR[11:0]	Receive Rule Label Set the 12-bit label information.
15	GAFLRMV	Receive Buffer Enable 0: No receive buffer is used. 1: A receive buffer is used.

b31	b30	b29	b28	Classical CAN Frame	CAN FD Frame
0	0	0	0	DLC check is disabled	
0	0	0	1	1 data byte	
0	0	1	0	2 data bytes	
0	0	1	1	3 data bytes	
0	1	0	0	4 data bytes	
0	1	0	1	5 data bytes	
0	1	1	0	6 data bytes	
0	1	1	1	7 data bytes	
1	0	0	0	8 data bytes	
1	0	0	1	8 data bytes	12 data bytes
1	0	1	0		16 data bytes
1	0	1	1		20 data bytes
1	1	0	0		24 data bytes
1	1	0	1		32 data bytes
1	1	1	0		48 data bytes
1	1	1	1		64 data bytes

Table 23.117 RSCFDnCFDGAFLP0_j Register Contents

Bit Position	Bit Name	Function
14 to 8	GAFLRMDP[6:0]	Receive Buffer Number Select Set the receive buffer number to store receive messages.
7 to 0	Reserved	These bits are read as the value after reset. The write value should be the value after reset.

Modify the RSCFDnCFDGAFLP0_j register when the AFLDAE bit in the RSCFDnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLDLC[3:0] Bits

These bits are used to set the minimum data length necessary for receiving messages. If the data length of a message that is being filtered is equal to or larger than the value set by the GAFLDLC[3:0] bits, the message passes the DLC check. Setting these bits to 0000_B disables the DLC check function allowing messages with any data length to pass the DLC check.

GAFLPTR[11:0] Bits

These bits are used to set a 12-bit label to be attached to messages that have passed through the filter. A label is attached when a message is stored in the receive buffer or the FIFO buffer.

GAFLRMV Bit

When this bit is set to 1, receive messages that have passed through the filter are stored in the receive buffer selected by the GAFLRMDP[6:0] bits.

GAFLRMDP[6:0] Bits

These bits are used to select the number of the receive buffer that stores receive messages that have passed through the filter when the GAFLRMV bit is set to 1. Set these bits to a value smaller than the value set by the NRXMB[7:0] bits in the RSCFDnCFDRMNB register.

23.4.5.6 RSCFDnCFDGAFLP1_j — Receive Rule Pointer 1 Register (j = 0 to 15)

Access: RSCFDnCFDGAFLP1_j register can be read/written in 32-bit units
RSCFDnCFDGAFLP1_jL, RSCFDnCFDGAFLP1_jH registers can be read/written in 16-bit units
RSCFDnCFDGAFLP1_jLL, RSCFDnCFDGAFLP1_jLH, RSCFDnCFDGAFLP1_jHL,
RSCFDnCFDGAFLP1_jHH registers can be read/written in 8-bit units

Address: RSCFDnCFDGAFLP1_j: <RSCFDn_base> + $100C_H$ + (10_H × j)

RSCFDnCFDGAFLP1_jL: <RSCFDn_base> + $100C_H$ + (10_H × j),
RSCFDnCFDGAFLP1_jH: <RSCFDn_base> + $100E_H$ + (10_H × j)

RSCFDnCFDGAFLP1_jLL: <RSCFDn_base> + $100C_H$ + (10_H × j),
RSCFDnCFDGAFLP1_jLH: <RSCFDn_base> + $100D_H$ + (10_H × j),
RSCFDnCFDGAFLP1_jHL: <RSCFDn_base> + $100E_H$ + (10_H × j),
RSCFDnCFDGAFLP1_jHH: <RSCFDn_base> + $100F_H$ + (10_H × j)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	GAFLFDP[25:16]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAFLFDP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.118 RSCFDnCFDGAFLP1_j Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
25 to 8	GAFLFDP[25:8]	Transmit/Receive FIFO Buffer k Select (Bit position –8 = target transmit/receive FIFO buffer number k) 0: Transmit/receive FIFO buffer is not selected. 1: Transmit/receive FIFO buffer is selected.
7 to 0	GAFLFDP[7:0]	Receive FIFO Buffer x Select (Bit position = target receive FIFO buffer number x) 0: Receive FIFO buffer is not selected. 1: Receive FIFO buffer is selected.

Modify the RSCFDnCFDGAFLP1_j register when the AFLDAE bit in the RSCFDnCFDGAFLECTR register is set to 1 (receive rule table write is enabled) in global reset mode.

GAFLFDP [25:0] Bits

These bits are used to specify FIFO buffers that store receive messages that have passed through the filter. Up to eight FIFO buffers are selectable. However, when the GAFLRMV bit in the RSCFDnCFDGAFLP0_j register is set to 1 (a message is stored in the receive buffer), up to seven FIFO buffers can be selected. Only receive FIFO buffers and the transmit/receive FIFO buffer for which the CFM[1:0] bits in the RSCFDnCFDCFCCk register are set to 00_B (receive mode) or 10_B (gateway mode) are selectable.

23.4.6 Details of Receive Buffer-related Registers

23.4.6.1 RSCFDnCFDRMNB — Receive Buffer Number Register

Access: RSCFDnCFDRMNB register can be read/written in 32-bit units
RSCFDnCFDRMNBL register can be read/written in 16-bit units
RSCFDnCFDRMNBLL, RSCFDnCFDRMNBLH registers can be read/written in 8-bit units

Address: RSCFDnCFDRMNB: <RSCFDn_base> + $00A4_H$

RSCFDnCFDRMNBL: <RSCFDn_base> + $00A4_H$

RSCFDnCFDRMNBLL: <RSCFDn_base> + $00A4_H$,
RSCFDnCFDRMNBLH: <RSCFDn_base> + $00A5_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RMPLS[1:0]		NRXMB[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.119 RSCFDnCFDRMNB Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
9, 8	RMPLS[1:0]	Receive Buffer Payload Storage Size Select b9 b8 0 0: 8 bytes 0 1: 12 bytes 1 0: 16 bytes 1 1: 20 bytes
7 to 0	NRXMB[7:0]	Receive Buffer Number Configuration Set the number of receive buffers. Set a value of 0 to 48.

Modify the RSCFDnCFDRMNB register only in global reset mode.

RMPLS[1:0] Bits

These bits are used to select the maximum payload size that can be stored in the receive buffer.

NRXMB[7:0] Bits

These bits are used to set the total number of receive buffers of the RS-CANFD module. The maximum value is 16 × (number of channels).

Setting these bits all to 0 makes receive buffers unavailable.

23.4.6.2 RSCFDnCFDRMNDy — Receive Buffer New Data Register (y = 0, 1)

Access: RSCFDnCFDRMNDy register can be read/written in 32-bit units
RSCFDnCFDRMNDyL, RSCFDnCFDRMNDyH registers can be read/written in 16-bit units
RSCFDnCFDRMNDyLL, RSCFDnCFDRMNDyLH, RSCFDnCFDRMNDyHL, RSCFDnCFDRMNDyHH registers can be read/written in 8-bit units

Address: RSCFDnCFDRMNDy: <RSCFDn_base> + $00A8_H$ + (04_H × y)

RSCFDnCFDRMNDyL: <RSCFDn_base> + $00A8_H$ + (04_H × y),
RSCFDnCFDRMNDyH: <RSCFDn_base> + $00AA_H$ + (04_H × y)

RSCFDnCFDRMNDyLL: <RSCFDn_base> + $00A8_H$ + (04_H × y),
RSCFDnCFDRMNDyLH: <RSCFDn_base> + $00A9_H$ + (04_H × y),
RSCFDnCFDRMNDyHL: <RSCFDn_base> + $00AA_H$ + (04_H × y),
RSCFDnCFDRMNDyHH: <RSCFDn_base> + $00AB_H$ + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMNSq (q = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMNSq (q = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.120 RSCFDnCFDRMNDy Register Contents

Bit Position	Bit Name	Function
31 to 16	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 31 to y × 32 + 16) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.
15 to 0	RMNSq	Receive Buffer Receive Complete Flag q (q = y × 32 + 15 to y × 32 + 0) 0: There is no new message in receive buffer q. 1: There is a new message in receive buffer q.

Write 0 to the RSCFDnCFDRMNDy register in global operating mode or global test mode.

RMNSq Flags (q = 0 to 47)

Each RMNSq flag is set to 1 when the processing for storing a message in the corresponding receive buffer starts.

To clear a flag to 0, the program must write 0 to the flag. Use a store instruction to write “0” to the flag and “1” to other flags. These bits cannot be set to 0 while a message is being stored. The message storing time depends on the storage payload size of the receive buffer. When the RMPLS[1:0] value in the RSCFDnCFDRMNB register is 00_B (8 bytes), the message storing time is 12 pclk clock cycles. When the RMPLS[1:0] value is 11_B (20 bytes), the message storing time is 18 pclk clock cycles. (2 pclk clock cycles per 4 bytes of storage payload size).

These flags are cleared to 0 in global reset mode.

23.4.6.3 RSCFDnCFDRMIDq — Receive Buffer ID Register (q = 0 to 47)

Access: RSCFDnCFDRMIDq register can be read only in 32-bit units
RSCFDnCFDRMIDqL, RSCFDnCFDRMIDqH registers can be read only in 16-bit units
RSCFDnCFDRMIDqLL, RSCFDnCFDRMIDqLH, RSCFDnCFDRMIDqHL, RSCFDnCFDRMIDqHH registers can be read only in 8-bit units

Address: RSCFDnCFDRMIDq: <RSCFDn_base> + 2000_H + (20_H × q)

RSCFDnCFDRMIDqL: <RSCFDn_base> + 2000_H + (20_H × q),
RSCFDnCFDRMIDqH: <RSCFDn_base> + 2002_H + (20_H × q)

RSCFDnCFDRMIDqLL: <RSCFDn_base> + 2000_H + (20_H × q),
RSCFDnCFDRMIDqLH: <RSCFDn_base> + 2001_H + (20_H × q),
RSCFDnCFDRMIDqHL: <RSCFDn_base> + 2002_H + (20_H × q),
RSCFDnCFDRMIDqHH: <RSCFDn_base> + 2003_H + (20_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMIDE	RMRTR	—	RMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.121 RSCFDnCFDRMIDq Register Contents

Bit Position	Bit Name	Function
31	RMIDE	Receive Buffer IDE 0: Standard ID 1: Extended ID
30	RMRTR	Receive Buffer RTR/RRS • When the received message is a classical CAN frame 0: Data frame 1: Remote frame • When the received message is a CAN FD frame The RRS bit value of the received message can be read.
29	Reserved	These bits are read as the value after reset.
28 to 0	RMID[28:0]	Receive Buffer ID Data These bits contain the standard ID or extended ID of the received message. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RMIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive buffer.

RMRTR Bit

When the received message is a classical CAN frame, this bit indicates the frame format (data frame or remote frame) of the message stored in the receive buffer. When the received message is a CAN FD frame, this bit indicates the RRS bit value in the message.

RMID[28:0] Bits

These bits contain the ID of the message stored in the receive buffer.

23.4.6.4 RSCFDnCFDRMPTRq — Receive Buffer Pointer Register (q = 0 to 47)

Access: RSCFDnCFDRMPTRq register can be read only in 32-bit units
RSCFDnCFDRMPTRqL, RSCFDnCFDRMPTRqH registers can be read only in 16-bit units
RSCFDnCFDRMPTRqLL, RSCFDnCFDRMPTRqLH, RSCFDnCFDRMPTRqHL, RSCFDnCFDRMPTRqHH registers can be read only in 8-bit units

Address: RSCFDnCFDRMPTRq: <RSCFDn_base> + 2004_H + (20_H × q)

RSCFDnCFDRMPTRqL: <RSCFDn_base> + 2004_H + (20_H × q),
RSCFDnCFDRMPTRqH: <RSCFDn_base> + 2006_H + (20_H × q)

RSCFDnCFDRMPTRqLL: <RSCFDn_base> + 2004_H + (20_H × q),
RSCFDnCFDRMPTRqLH: <RSCFDn_base> + 2005_H + (20_H × q),
RSCFDnCFDRMPTRqHL: <RSCFDn_base> + 2006_H + (20_H × q),
RSCFDnCFDRMPTRqHH: <RSCFDn_base> + 2007_H + (20_H × q)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDLC[3:0]				RMPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.122 RSCFDnCFDRMPTRq Register Contents

Bit Position	Bit Name	Function
31 to 28	RMDLC[3:0]	Receive Buffer DLC Data

b31	b30	b29	b28	Classical CAN Frame	CAN FD Frame
0	0	0	0	0 data bytes	
0	0	0	1	1 data byte	
0	0	1	0	2 data bytes	
0	0	1	1	3 data bytes	
0	1	0	0	4 data bytes	
0	1	0	1	5 data bytes	
0	1	1	0	6 data bytes	
0	1	1	1	7 data bytes	
1	0	0	0	8 data bytes	
1	0	0	1	8 data bytes	12 data bytes
1	0	1	0		16 data bytes
1	0	1	1		20 data bytes
1	1	0	0		24 data bytes
1	1	0	1		32 data bytes
1	1	1	0		48 data bytes
1	1	1	1		64 data bytes

27 to 16	RMPTR[11:0]	Receive Buffer Label Data Label information of the received message.
15 to 0	RMTS[15:0]	Receive Buffer Timestamp Data Timestamp value of the received message.

RMDLC[3:0] Bits

These bits indicate the data length of the message stored in the receive buffer. The number of bytes of the payload to be stored in the receive buffer is determined by the RMPLS[1:0] bits in the RSCFDnCFDRMNB register.

RMPTR[11:0] Bits

These bits indicate the label information of the message stored in the receive buffer.

RMTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the receive buffer.

23.4.6.5 RSCFDnCFDRMFDSTSq — Receive Buffer CAN FD Status Register (q = 0 to 47)

Access: RSCFDnCFDRMFDSTSq register can be read only in 32-bit units
RSCFDnCFDRMFDSTSqL register can be read only in 16-bit units
RSCFDnCFDRMFDSTSqLL register can be read only in 8-bit units

Address: RSCFDnCFDRMFDSTSq: <RSCFDn_base> + 2008_H + (20_H × q)

RSCFDnCFDRMFDSTSqL: <RSCFDn_base> + 2008_H + (20_H × q)

RSCFDnCFDRMFDSTSqLL: <RSCFDn_base> + 2008_H + (20_H × q)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RMFDF	RMBRS	RMESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.123 RSCFDnCFDRMFDSTSq Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	These bits are read as the value after reset.
2	RMFDF	FDF 0: Classical CAN frame 1: CAN FD frame
1	RMBRS	BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes.
0	RMESI	ESI 0: Error active node 1: Error passive node

RMFDF Bit

This bit indicates the FD format (classical CAN frame or CAN FD frame) of the message stored in the receive buffer.

RMBRS Bit

When the RMFDF bit is 1, this bit indicates the BRS bit value of the message stored in the receive buffer. When the RMFDF bit is 0, this bit is always read as 0.

RMESI Bit

When the RMFDF bit is 1, this bit indicates the ESI bit value of the message stored in the receive buffer. When the RMFDF bit is 0, this bit is always read as 0.

23.4.6.6 RSCFDnCFDRMDFb_q — Receive Buffer Data Field b Register (b = 0 to 4, q = 0 to 47)

Access: RSCFDnCFDRMDFb_q register can be read only in 32-bit units
RSCFDnCFDRMDFb_qL, RSCFDnCFDRMDFb_qH registers can be read only in 16-bit units
RSCFDnCFDRMDFb_qLL, RSCFDnCFDRMDFb_qLH, RSCFDnCFDRMDFb_qHL, RSCFDnCFDRMDFb_qHH registers can be read only in 8-bit units

Address: RSCFDnCFDRMDFb_q: <RSCFDn_base> + $200C_H$ + (04_H × b) + (20_H × q)

RSCFDnCFDRMDFb_qL: <RSCFDn_base> + $200C_H$ + (04_H × b) + (20_H × q),
RSCFDnCFDRMDFb_qH: <RSCFDn_base> + $200E_H$ + (04_H × b) + (20_H × q)

RSCFDnCFDRMDFb_qLL: <RSCFDn_base> + $200C_H$ + (04_H × b) + (20_H × q),
RSCFDnCFDRMDFb_qLH: <RSCFDn_base> + $200D_H$ + (04_H × b) + (20_H × q),
RSCFDnCFDRMDFb_qHL: <RSCFDn_base> + $200E_H$ + (04_H × b) + (20_H × q),
RSCFDnCFDRMDFb_qHH: <RSCFDn_base> + $200F_H$ + (04_H × b) + (20_H × q)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RMDB4 × b + 3 [7:0]								RMDB4 × b + 2 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RMDB4 × b + 1 [7:0]								RMDB4 × b + 0 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.124 RSCFDnCFDRMDFb_q Register Contents

Bit Position	Bit Name	Function
31 to 24	RMDB4 × b + 3 [7:0]	Receive Buffer Data Byte 4 × b + 3 Receive Buffer Data Byte 4 × b + 2 Receive Buffer Data Byte 4 × b + 1 Receive Buffer Data Byte 4 × b + 0 Data for a message stored in the receive buffer can be read.
23 to 16	RMDB4 × b + 2 [7:0]	
15 to 8	RMDB4 × b + 1 [7:0]	
7 to 0	RMDB4 × b + 0 [7:0]	

When the RMDLC[3:0] value in the RSCFDnCFDRMPTRq register is smaller than the payload storage size of the receive buffer, data bytes for which no data is set are read as 00_H.

Specify the payload storage size of the receive buffer by the RMPLS[1:0] bits in the RSCFDnCFDRMNB register. Do not read or write the RSCFDnCFDRMDFb_q register corresponding to an area larger than the specified size.

23.4.7 Details of Receive FIFO Buffer-related Registers

23.4.7.1 RSCFDnCFDRFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7)

Access: RSCFDnCFDRFCCx register can be read/written in 32-bit units
RSCFDnCFDRFCCxL register can be read/written in 16-bit units
RSCFDnCFDRFCCxLL, RSCFDnCFDRFCCxLH registers can be read/written in 8-bit units

Address: RSCFDnCFDRFCCx: <RSCFDn_base> + $00B8_H$ + (04_H × x)

RSCFDnCFDRFCCxL: <RSCFDn_base> + $00B8_H$ + (04_H × x)

RSCFDnCFDRFCCxLL: <RSCFDn_base> + $00B8_H$ + (04_H × x),
RSCFDnCFDRFCCxLH: <RSCFDn_base> + $00B9_H$ + (04_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	RFPLS[2:0]			—	—	RFIE	RFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R	R/W	R/W

Table 23.125 RSCFDnCFDRFCCx Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
15 to 13	RFIGCV[2:0]	Receive FIFO Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.
12	RFIM	Receive FIFO Interrupt Source Select 0: An interrupt occurs when the condition set by the RFIGCV[2:0] bits is met. 1: An interrupt occurs each time a message has been received.
11	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
10 to 8	RFDC[2:0]	Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7	Reserved	These bits are read as the value after reset. The write value should be the value after reset.

Table 23.125 RSCFDnCFDRFCCx Register Contents (2/2)

Bit Position	Bit Name	Function
6 to 4	RFPLS[2:0]	Receive FIFO Buffer Payload Storage Size Select b6 b5 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes
3 to 2	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
1	RFIE	Receive FIFO Interrupt Enable 0: Receive FIFO interrupt is disabled. 1: Receive FIFO interrupt is enabled.
0	RFE	Receive FIFO Buffer Enable 0: No receive FIFO buffer is used. 1: Receive FIFO buffers are used.

RFIGCV[2:0] Bits

These bits are used to specify the number of received messages for generating a receive FIFO interrupt request when the RFIM bit is set to 0 with a fraction for the total number of buffers (the setting of RFDC[2:0]).

When the RFDC[2:0] bits are set to 001_B (4 messages), set the RFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B. Modify these bits only in global reset mode.

RFIM Bit

This bit is used to select a FIFO interrupt source. Modify this bit only in global reset mode.

RFDC[2:0] Bits

These bits are used to select the number of messages that can be stored in a single receive FIFO buffer. When these bits are set to 000_B, no receive FIFO buffer should be used. Modify these bits only in global reset mode.

RFPLS[2:0] Bits

These bits are used to select the maximum payload size that can be stored in the receive FIFO buffer. Modify these bits only in global reset mode.

RFIE Bit

Setting the RFIE bit to 1 enables receive FIFO interrupts. Modify this bit when the RFE bit set to 0 (no receive FIFO buffer is used).

RFE Bit

Setting the RFE bit to 1 makes receive FIFO buffers available. Clearing this bit to 0 sets the RFEMP flag in the RSCFDnCFDRFSTSx register to 1 (buffer empty). Modify this bit in global operating mode or global test mode. Set this bit to 1 with another instruction after the settings to all bits in the RSCFDnCFDRFCCx register have been done. This bit is cleared to 0 in global reset mode.

23.4.7.2 RSCFDnCFDRFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7)

Access: RSCFDnCFDRFSTSx register can be read/written in 32-bit units
RSCFDnCFDRFSTSxL register can be read/written in 16-bit units
RSCFDnCFDRFSTSxLL register can be read/written in 8-bit units
RSCFDnCFDRFSTSxLH register is the read-only register that can be read in 8-bit units.

Address: RSCFDnCFDRFSTSx: <RSCFDn_base> + $00D8_H$ + (04_H × x)

RSCFDnCFDRFSTSxL: <RSCFDn_base> + $00D8_H$ + (04_H × x)

RSCFDnCFDRFSTSxLL: <RSCFDn_base> + $00D8_H$ + (04_H × x),
RSCFDnCFDRFSTSxLH: <RSCFDn_base> + $00D9_H$ + (04_H × x)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFMC[7:0]								—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.126 RSCFDnCFDRFSTSx Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
15 to 8	RFMC[7:0]	Receive FIFO Unread Message Counter The number of unread messages stored in the receive FIFO buffer is displayed.
7 to 4	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
3	RFIF	Receive FIFO Interrupt Request Flag 0: No receive FIFO interrupt request is present. 1: A receive FIFO interrupt request is present.
2	RFMLT	Receive FIFO Message Lost Flag 0: No receive FIFO message is lost. 1: A receive FIFO message is lost.
1	RFFLL	Receive FIFO Buffer Full Status Flag 0: The receive FIFO buffer is not full. 1: The receive FIFO buffer is full.
0	RFEMP	Receive FIFO Buffer Empty Status Flag 0: The receive FIFO buffer contains unread message. 1: The receive FIFO buffer contains no unread message (buffer empty).

RFMC[7:0] Flag

This flag indicates the number of unread messages in the receive FIFO buffer. This flag becomes 00_H when the RFE bit in the RSCFDnCFDRFCCx register is set to 0.

This flag is 00_H in global reset mode.

RFIF Flag

This flag is set to 1 when the receive FIFO interrupt request generation conditions set by the RFIGCV[2:0] bits and the RFIM bit in the RSCFDnCFDRFCCx register are met. This flag is cleared to 0 in global reset mode or by writing 0 to this flag. Modify this bit in global operating mode or global test mode.
To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFMLT Flag

This flag is set to 1 when an attempt is made to store a new message while the receive FIFO buffer is full. In this case, the new message is discarded.

This flag is cleared to 0 in global reset mode or by writing 0 to this flag.

Modify this bit in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

RFFLL Flag

This flag is set to 1 when the number of messages stored in the receive FIFO buffer matches the FIFO buffer depth set by the RFDC[2:0] bits in the RSCFDnCFDRFCCx register.

If the number of messages stored in the receive FIFO buffer becomes smaller than the FIFO buffer depth set by the RFDC[2:0] bits, this flag is cleared to 0. This flag is also cleared to 0 when the RFE bit in the RSCFDnCFDRFCCx register is set to 0 (no receive FIFO buffer is used) or in global reset mode.

RFEMP Flag

This flag is set to 1 when all messages in the receive FIFO buffer have been read. This flag is also set to 1 when the RFE bit in the RSCFDnCFDRFCCx register is 0 or in global reset mode.

This flag is cleared to 0 when even a single received message has been stored in the receive FIFO buffer.

NOTE

To clear the RFMLT or RFIF flag to 0, use a store instruction to write “0” to the given flag and “1” to the other flags.

23.4.7.3 RSCFDnCFDRFPCTRx — Receive FIFO Buffer Pointer Control Register (x = 0 to 7)

Access: RSCFDnCFDRFPCTRx register can only be written in 32-bit units
RSCFDnCFDRFPCTRxL register can only be written in 16-bit units
RSCFDnCFDRFPCTRxLL register can only be written in 8-bit units

Address: RSCFDnCFDRFPCTRx: <RSCFDn_base> + $00F8_H$ + (04_H × x)

RSCFDnCFDRFPCTRxL: <RSCFDn_base> + $00F8_H$ + (04_H × x)

RSCFDnCFDRFPCTRxLL: <RSCFDn_base> + $00F8_H$ + (04_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.127 RSCFDnCFDRFPCTRx Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be the value after reset.
7 to 0	RFPC[7:0]	Receive FIFO Pointer Control When these bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer.

RFPC[7:0] Bits

When the RFPC[7:0] bits are set to FF_H, the read pointer moves to the next unread message in the receive FIFO buffer. At this time, the RFMC[7:0] (receive FIFO unread message counter) value in the RSCFDnCFDRFSTSx register is decremented by 1. Read the RSCFDnCFDRFIDx, RSCFDnCFDRFPTRx, RSCFDnCFDRFFDSTSx, and RSCFDnCFDRFDFd_x registers to read messages in the receive FIFO buffer, and then write FF_H to the RFPC[7:0] bits.

When writing FF_H to these bits, make sure that the RFE bit in the RSCFDnCFDRFCCx register is set to 1 (receive FIFO buffers are used) and the RFEMP flag in the RSCFDnCFDRFSTSx register is 0 (the receive FIFO buffer contains unread messages).

23.4.7.4 RSCFDnCFDRFIDx — Receive FIFO Buffer Access ID Register (x = 0 to 7)

Access: RSCFDnCFDRFIDx register can be read only in 32-bit units
RSCFDnCFDRFIDxL, RSCFDnCFDRFIDxH registers can be read only in 16-bit units
RSCFDnCFDRFIDxLL, RSCFDnCFDRFIDxLH, RSCFDnCFDRFIDxHL, RSCFDnCFDRFIDxHH registers can be read only in 8-bit units

Address: RSCFDnCFDRFIDx: <RSCFDn_base> + 3000_H + (80_H × x)

RSCFDnCFDRFIDxL: <RSCFDn_base> + 3000_H + (80_H × x),
RSCFDnCFDRFIDxH: <RSCFDn_base> + 3002_H + (80_H × x)

RSCFDnCFDRFIDxLL: <RSCFDn_base> + 3000_H + (80_H × x),
RSCFDnCFDRFIDxLH: <RSCFDn_base> + 3001_H + (80_H × x),
RSCFDnCFDRFIDxHL: <RSCFDn_base> + 3002_H + (80_H × x),
RSCFDnCFDRFIDxHH: <RSCFDn_base> + 3003_H + (80_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFIDE	RFRTR	—	RFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.128 RSCFDnCFDRFIDx Register Contents

Bit Position	Bit Name	Function
31	RFIDE	Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	RFRTR	Receive FIFO Buffer RTR/RRS • When the received message is a classical CAN frame 0: Data frame 1: Remote frame • When the received message is a CAN FD frame The RRS bit value of the received message can be read.
29	Reserved	These bits are read as the value after reset.
28 to 0	RFID[28:0]	Receive FIFO Buffer ID Data The standard ID or extended ID of received message can be read. Read bits b10 to b0 for standard ID. Bits b28 to b11 are read as 0.

RFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the message stored in the receive FIFO buffer.

RFRTR Bit

When the received message is a classical CAN frame, this bit indicates the frame format (data frame or remote frame) of the message stored in the receive FIFO buffer.

When the received message is a CAN FD frame, this bit indicates the RRS bit value in the message.

RFID[28:0] Bits

These bits indicate the ID of the message stored in the receive FIFO buffer.

23.4.7.5 RSCFDnCFDRFPTRx — Receive FIFO Buffer Access Pointer Register (x = 0 to 7)

Access: RSCFDnCFDRFPTRx register can be read only in 32-bit units
RSCFDnCFDRFPTRxL, RSCFDnCFDRFPTRxH registers can be read only in 16-bit units
RSCFDnCFDRFPTRxLL, RSCFDnCFDRFPTRxLH, RSCFDnCFDRFPTRxHL, RSCFDnCFDRFPTRxHH registers can be read only in 8-bit units

Address: RSCFDnCFDRFPTRx: <RSCFDn_base> + 3004_H + (80_H × x)

RSCFDnCFDRFPTRxL: <RSCFDn_base> + 3004_H + (80_H × x),
RSCFDnCFDRFPTRxH: <RSCFDn_base> + 3006_H + (80_H × x)

RSCFDnCFDRFPTRxLL: <RSCFDn_base> + 3004_H + (80_H × x),
RSCFDnCFDRFPTRxLH: <RSCFDn_base> + 3005_H + (80_H × x),
RSCFDnCFDRFPTRxHL: <RSCFDn_base> + 3006_H + (80_H × x),
RSCFDnCFDRFPTRxHH: <RSCFDn_base> + 3007_H + (80_H × x)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDLC[3:0]				RFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.129 RSCFDnCFDRFPTRx Register Contents

Bit Position	Bit Name	Function
31 to 28	RFDLC[3:0]	Receive FIFO Buffer DLC Data

b31	b30	b29	b28	Classical CAN Frame	CAN FD Frame
0	0	0	0	0 data bytes	
0	0	0	1	1 data byte	
0	0	1	0	2 data bytes	
0	0	1	1	3 data bytes	
0	1	0	0	4 data bytes	
0	1	0	1	5 data bytes	
0	1	1	0	6 data bytes	
0	1	1	1	7 data bytes	
1	0	0	0	8 data bytes	
1	0	0	1	8 data bytes	12 data bytes
1	0	1	0		16 data bytes
1	0	1	1		20 data bytes
1	1	0	0		24 data bytes
1	1	0	1		32 data bytes
1	1	1	0		48 data bytes
1	1	1	1		64 data bytes

27 to 16	RFPTR[11:0]	Receive FIFO Buffer Label Data Label information of the received message can be read.
15 to 0	RFTS[15:0]	Receive FIFO Buffer Timestamp Data Timestamp value of the received message can be read.

RFDLC[3:0] Bits

These bits contain the data length of the message stored in the receive FIFO buffer.

RFPTR[11:0] Bits

These bits contain the label information of the message stored in the receive FIFO buffer.

RFTS[15:0] Bits

These bits contain the timestamp value of the message stored in the receive FIFO buffer.

23.4.7.6 RSCFDnCFDRFFDSTSx — Receive FIFO CAN FD Status Register (x = 0 to 7)

Access: RSCFDnCFDRFFDSTSx register can be read only in 32-bit units
RSCFDnCFDRFFDSTSxL register can be read only in 16-bit units
RSCFDnCFDRFFDSTSxLL register can be read only in 8-bit units

Address: RSCFDnCFDRFFDSTSx: <RSCFDn_base> + 3008_H + (80_H × x)

RSCFDnCFDRFFDSTSxL: <RSCFDn_base> + 3008_H + (80_H × x)

RSCFDnCFDRFFDSTSxLL: <RSCFDn_base> + 3008_H + (80_H × x)

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFDF	RFBRS	RFESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.130 RSCFDnCFDRFFDSTSx Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	These bits are read as the value after reset.
2	RFFDF	FDF 0: Classical CAN frame 1: CAN FD frame
1	RFBRS	BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes.
0	RFESI	ESI 0: Error active node 1: Error passive node

RFFDF Bit

This bit indicates the FD format (classical CAN frame or CAN FD frame) of the message stored in the receive FIFO buffer.

RFBRS Bit

When the RFFDF bit is 1, this bit indicates the BRS bit value of the message stored in the receive FIFO buffer. When the RFFDF bit is 0, this bit is always read as 0.

RFESI Bit

When the RFFDF bit is 1, this bit indicates the ESI bit value of the message stored in the receive FIFO buffer. When the RFFDF bit is 0, this bit is always read as 0.

23.4.7.7 RSCFDnCFDRFDFd_x — Receive FIFO Buffer Access Data Field d Register (d = 0 to 15, x = 0 to 7)

Access: RSCFDnCFDRFDFd_x registers can be read only in 32-bit units
RSCFDnCFDRFDFd_xL, RSCFDnCFDRFDFd_xH registers can be read only in 16-bit units
RSCFDnCFDRFDFd_xLL, RSCFDnCFDRFDFd_xLH, RSCFDnCFDRFDFd_xHL, RSCFDnCFDRFDFd_xHH registers can be read only in 8-bit units

Address: RSCFDnCFDRFDFd_x: <RSCFDn_base> + $300C_H$ + (04_H × d) + (80_H × x)

RSCFDnCFDRFDFd_xL: <RSCFDn_base> + $300C_H$ + (04_H × d) + (80_H × x),
RSCFDnCFDRFDFd_xH: <RSCFDn_base> + $300E_H$ + (04_H × d) + (80_H × x)

RSCFDnCFDRFDFd_xLL: <RSCFDn_base> + $300C_H$ + (04_H × d) + (80_H × x),
RSCFDnCFDRFDFd_xLH: <RSCFDn_base> + $300D_H$ + (04_H × d) + (80_H × x),
RSCFDnCFDRFDFd_xHL: <RSCFDn_base> + $300E_H$ + (04_H × d) + (80_H × x),
RSCFDnCFDRFDFd_xHH: <RSCFDn_base> + $300F_H$ + (04_H × d) + (80_H × x)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RFDB4 × d + 3 [7:0]								RFDB4 × d + 2 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RFDB4 × d + 1 [7:0]								RFDB4 × d + 0 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.131 RSCFDnCFDRFDFd_x Register Contents

Bit Position	Bit Name	Function
31 to 24	RFDB4 × d + 3 [7:0]	Receive Buffer Data Byte 4 × d + 3 Receive Buffer Data Byte 4 × d + 2 Receive Buffer Data Byte 4 × d + 1 Receive Buffer Data Byte 4 × d + 0 Data for a message stored in the receive buffer can be read.
23 to 16	RFDB4 × d + 2 [7:0]	
15 to 8	RFDB4 × d + 1 [7:0]	
7 to 0	RFDB4 × d + 0 [7:0]	

When the RFDLC[3:0] value in the RSCFDnCFDRFPTRx register is smaller than the payload storage size of the receive FIFO buffer, data bytes for which no data is set are read as 00_H.

Specify the payload storage size of the receive FIFO buffer by the RFPLS[2:0] bits in the RSCFDnCFDRFCCx register. Do not read or write the RSCFDnCFDRFDFd_x register corresponding to an area larger than the specified size.

23.4.8 Details of Transmit/Receive FIFO Buffer Related Registers

23.4.8.1 RSCFDnCFDCFCCk — Transmit/receive FIFO Buffer Configuration and Control Register k (k = 0 to 8)

Access: RSCFDnCFDCFCCk register can be read/written in 32-bit units
RSCFDnCFDCFCCkL, RSCFDnCFDCFCCkH registers can be read/written in 16-bit units
RSCFDnCFDCFCCkLL, RSCFDnCFDCFCCkLH, RSCFDnCFDCFCCkHL, RSCFDnCFDCFCCkHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCFCCk: <RSCFDn_base> + 0118_H + (04_H × k)

RSCFDnCFDCFCCkL: <RSCFDn_base> + 0118_H + (04_H × k),
RSCFDnCFDCFCCkH: <RSCFDn_base> + $011A_H$ + (04_H × k)

RSCFDnCFDCFCCkLL: <RSCFDn_base> + 0118_H + (04_H × k),
RSCFDnCFDCFCCkLH: <RSCFDn_base> + 0119_H + (04_H × k),
RSCFDnCFDCFCCkHL: <RSCFDn_base> + $011A_H$ + (04_H × k),
RSCFDnCFDCFCCkHH: <RSCFDn_base> + $011B_H$ + (04_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFITT[7:0]								CFTML[3:0]				CFITR	CFITSS	CFM[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFIGCV[2:0]			CFIM	—	CFDC[2:0]			—	CFPLS[2:0]			—	CFTXIE	CFRXIE	CFE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 23.132 RSCFDnCFDCFCCk Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	CFITT[7:0]	Set a message transmission interval. Set Value: 00_H to FF_H
23 to 20	CFTML[3:0]	Transmit Buffer Link Configuration Set the transmit buffer number to be linked to the transmit/receive FIFO buffer.
19	CFITR	Transmit/Receive FIFO Interval Timer Resolution 0: Clock dividing pclk/2 by (ITRCP [15:0] bits) 1: Clock dividing pclk/2 by (ITRCP [15:0] bits × 10)
18	CFITSS	Transmit/Receive FIFO Interval Timer Clock Source Select 0: Interval timer clock source selected by the CFITR bit 1: Interval timer clock source is the nominal bit time clock for the channel to which the FIFO is linked.
17, 16	CFM[1:0]	Transmit/Receive FIFO Mode Select b17 b16 0 0: Receive mode 0 1: Transmit mode 1 0: Gateway mode 1 1: Setting prohibited
15 to 13	CFIGCV[2:0]	Transmit/Receive FIFO Receive Interrupt Request Timing Select b15 b14 b13 0 0 0: When FIFO is 1/8 full. 0 0 1: When FIFO is 2/8 full. 0 1 0: When FIFO is 3/8 full. 0 1 1: When FIFO is 4/8 full. 1 0 0: When FIFO is 5/8 full. 1 0 1: When FIFO is 6/8 full. 1 1 0: When FIFO is 7/8 full. 1 1 1: When FIFO is full.

Table 23.132 RSCFDnCFDCFCCk Register Contents (2/2)

Bit Position	Bit Name	Function
12	CFIM	Transmit/Receive FIFO Interrupt Source Select 0: • Receive mode/gateway mode When the number of received messages has met the condition set by the CFIGCV[2:0] bits, a FIFO receive interrupt request is generated. • Transmit mode/gateway mode When the buffer becomes empty upon completion of message transmission, a FIFO transmit interrupt request is generated. 1: • Receive mode/gateway mode A FIFO receive interrupt request is generated each time a message has been received. • Transmit mode/gateway mode A FIFO transmit interrupt request is generated each time a message has been transmitted.
11	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
10 to 8	CFDC[2:0]	Transmit/Receive FIFO Buffer Depth Configuration b10 b9 b8 0 0 0: 0 messages 0 0 1: 4 messages 0 1 0: 8 messages 0 1 1: 16 messages 1 0 0: 32 messages 1 0 1: 48 messages 1 1 0: 64 messages 1 1 1: 128 messages
7	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
6 to 4	CFPLS[2:0]	Transmit/Receive FIFO Buffer Payload Storage Size Select b6 b5 b4 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes
3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2	CFTXIE	Transmit/Receive FIFO Transmit Interrupt Enable 0: Transmit/receive FIFO transmit interrupt is disabled. 1: Transmit/receive FIFO transmit interrupt is enabled.
1	CFRXIE	Transmit/Receive FIFO Receive Interrupt Enable 0: Transmit/receive FIFO receive interrupt is disabled. 1: Transmit/receive FIFO receive interrupt is enabled.
0	CFE	Transmit/Receive FIFO Buffer Enable 0: No transmit/receive FIFO buffer is used. 1: Transmit/receive FIFO buffers are used.

CFITT[7:0] Bits

These bits are used to set a message transmission interval when transmitting messages continuously from a transmit/receive FIFO buffer whose CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode).

Clear the CFE bit to 0 (no transmit/receive FIFO buffer is used) before modifying the CFITT[7:0] bits.

CFTML[3:0] Bits

These bits are used to set the number of transmit buffer on the channel which will be linked to transmit/receive FIFO buffer k when the CFM[1:0] bits are set to 01_B (transmit mode) or 10_B (gateway mode). There are three transmit/receive FIFO buffers per channel, so channel number m of FIFO buffer k is calculated as m = k/3 (integer division). The actual assigned transmit buffer number p linked to FIFO buffer k will be ((16 × m) + CFTML[3:0]) (see **Table 23.91**).

See **Table 23.89** and **Table 23.90**, as for the relationship between transmit/receive FIFO buffer k and transmit buffer p.

Setting the CFDC[2:0] bits to 001_B or more enables the setting of the CFTML[3:0] bits.

Do not link to any transmit buffer which is already allocated to a transmit queue on the same channel or to another transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFITR Bit

This bit is enabled when the CFITSS bit is 0.

When this bit is 0, the interval timer clock source is the pclk/2 clock divided by the value of the ITRCP[15:0] bits in the RSCFDnCFDGCFG register.

When this bit is 1, the interval timer clock source is the pclk/2 clock divided by (the value of the ITRCP[15:0] bits in the RSCFDnCFDGCFG register × 10).

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFITSS Bit

When this bit is 0, the clock selected by the CFITR bit is the count source of the interval timer.

When this bit is 1, the nominal bit time clock of the channel to which the FIFO is linked is the count source of the interval timer. Use this count source only for the channel does not handle the CAN FD frames.

Modify this bit while the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

CFM[1:0] Bits

These bits are used to select transmit/receive FIFO mode. Modify these bits only in global reset mode.

CFIGCV[2:0] Bits

These bits are used to specify the number of received messages for generating a transmit/receive FIFO receive interrupt request when the CFM[1:0] bits are set to 00_B (receive mode) or 10_B (gateway mode) and the CFIM bit is set to 0 with a fraction for the total number of buffers (the setting of CFDC[2:0]).

When the CFDC[2:0] bits are set to 001_B (4 messages), set the CFIGCV[2:0] bits to 001_B, 011_B, 101_B, or 111_B.

Modify these bits only in global reset mode.

CFIM Bit

This bit is used to select a transmit/receive FIFO interrupt source. Modify this bit only in global reset mode.

CFDC[2:0] Bits

These bits are used to set the number of messages that can be stored in a single transmit/receive FIFO buffer. When these bits are set to 000_B, do not use a transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFPLS[2:0] Bits

These bits are used to select the maximum payload size that can be stored in the transmit/receive FIFO buffer. Modify these bits only in global reset mode.

CFTXIE Bit

When this bit is set to 1 and the CFTXIF flag in the RSCFDnCFDCFSTSk register is set to 1, a transmit/receive FIFO transmit interrupt request is generated.

Modify this bit with the CFE bit set to 0 (no transmit/receive FIFO buffer is used).

CFRXIE Bit

When this bit is set to 1 and the CFRXIF flag in the RSCFDnCFDCFSTSk register is set to 1, a transmit/receive FIFO receive interrupt request is generated.

Modify this bit with the CFE bit set to 0.

CFE Bit

Setting this bit to 1 makes transmit/receive FIFO buffers available.

When this bit is set to 0 in transmit mode or gateway mode, if a message in the transmit/receive FIFO buffer is being transmitted or will be transmitted next, the transmit/receive FIFO buffer becomes empty after completion of transmission of that message, or upon detection of a CAN bus error, or arbitration-lost. In other cases or in receive mode, the transmit/receive FIFO buffer becomes empty immediately.

This bit is cleared to 0 when the following conditions are met.

- Receive mode: Global reset mode
- Transmit mode or gateway mode: Channel reset mode

Modify this bit in the following mode.

- Receive mode: Global operating mode or global test mode
- Transmit mode or gateway mode: Channel communication mode or channel halt mode

After all other bits in the RSCFDnCFDCFCCk register have been set, set this bit to 1 by using another instruction.

23.4.8.2 RSCFDnCFDCFSTSk — Transmit/receive FIFO Buffer Status Register (k = 0 to 8)

Access: RSCFDnCFDCFSTSk register can be read/written in 32-bit units
RSCFDnCFDCFSTSkL register can be read/written in 16-bit units
RSCFDnCFDCFSTSkLL register can be read/written in 8-bit units
RSCFDnCFDCFSTSkLH register is the read-only register that can be read in 8-bit units

Address: RSCFDnCFDCFSTSk: <RSCFDn_base> + 0178_H + (04_H × k)

RSCFDnCFDCFSTSkL: <RSCFDn_base> + 0178_H + (04_H × k)

RSCFDnCFDCFSTSkLL: <RSCFDn_base> + 0178_H + (04_H × k),
RSCFDnCFDCFSTSkLH: <RSCFDn_base> + 0179_H + (04_H × k)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFMC[7:0]								—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.133 RSCFDnCFDCFSTSk Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
15 to 8	CFMC[7:0]	Transmit/Receive FIFO Message Counter The number of messages stored in the transmit/receive FIFO buffer.
7 to 5	Reserved	These bits are read as the value after reset. When writing to these bits, write the value after reset.
4	CFTXIF	Transmit/Receive FIFO Transmit Interrupt Request Flag 0: No transmit/receive FIFO transmit interrupt request is present. 1: A transmit/receive FIFO transmit interrupt request is present.
3	CFRXIF	Transmit/Receive FIFO Receive Interrupt Request Flag 0: No transmit/receive FIFO receive interrupt request is present. 1: A transmit/receive FIFO receive interrupt request is present.
2	CFMLT	Transmit/Receive FIFO Message Lost Flag 0: No transmit/receive FIFO message is lost. 1: A transmit/receive FIFO message is lost.
1	CFFLL	Transmit/Receive FIFO Buffer Full Status Flag 0: The transmit/receive FIFO buffer is not full. 1: The transmit/receive FIFO buffer is full.
0	CFEMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: The transmit/receive FIFO buffer contains messages. 1: The transmit/receive FIFO buffer contains no message (buffer empty).

CFMC[7:0] Bits

The CFMC[7:0] bits indicate the following values that depend on the setting of the CFM[1:0] bits in the RSCFDnCFDCFCCk register.

- When CFM[1:0] value is 01_B (transmit mode): Number of untransmitted messages in the buffer
- When CFM[1:0] value is 00_B (receive mode): Number of unread received messages in the buffer
- When CFM[1:0] value is 10_B (gateway mode): Number of untransmitted received messages in the buffer

These bits are cleared to 0 when any of the following conditions is met.

- When CFM[1:0] value is 00_B: In global reset mode
- When CFM[1:0] value is 01_B or 10_B: In channel reset mode
- When the CFE bit in the RSCFDnCFDCFCCk register is cleared to 0.

CFTXIF Flag

The CFTXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 01_B or 10_B, and the factor selected by the CFIM bit in the RSCFDnCFDCFCCk register occurs

The CFTXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFTXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFRXIF Flag

The CFRXIF flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B, and the factor selected by the CFIM bit in the RSCFDnCFDCFCCk register occurs

The CFRXIF flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFRXIF flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFMLT Flag

The CFMLT flag is set to 1 when any of the following conditions is met.

- When an attempt is made to store a new message while the transmit/receive FIFO buffer is full. In this case, the new message is discarded.

The CFMLT flag is cleared to 0 when any of the following conditions is met.

- When 0 is written to the CFMLT flag
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

Write 0 to this flag in global operating mode or global test mode

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared. When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

CFFLL Flag

The CFFLL flag is set to 1 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer matches the FIFO buffer depth set by the CFDC[2:0] bits in the RSCFDnCFDCFCCk register.

The CFFLL flag is cleared to 0 when any of the following conditions is met.

- When the number of messages stored in the transmit/receive FIFO buffer becomes smaller than the FIFO buffer depth set by the CFDC[2:0] bits.
- When the CFE bit in the RSCFDnCFDCFCCk register is 0 (no transmit/receive FIFO buffer is used): When not in the transmit abort
- When the CFM[1:0] bits are set to 00_B: In global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: In channel reset mode

CFEMP Flag

The CFEMP flag is set to 1 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B: All messages have been read, or in global reset mode
- When the CFM[1:0] bits are set to 01_B or 10_B: All messages have been transmitted, or in channel reset mode
- When the CFE bit is 0 (no transmit/receive FIFO buffer is used): Not in the transmit abort

The CFEMP flag is cleared to 0 when any of the following conditions is met.

- When the CFM[1:0] bits are set to 00_B or 10_B: At least one received message has been stored in the transmit/receive FIFO buffer.
- When the CFM[1:0] bits are set to 01_B: A value of FF_H has been written to the RSCFDnCFDCFPCTRk register after data was written to the RSCFDnCFDCFIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDnCFDCFDFd_k registers.

NOTE

To clear CFTXIF, CFRXIF, or CFMLT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

23.4.8.3 RSCFDnCFDCFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register (k = 0 to 8)

Access: RSCFDnCFDCFPCTRk register can only be written in 32-bit units
RSCFDnCFDCFPCTRkL register is a write-only register that can only be written in 16-bit units
RSCFDnCFDCFPCTRkLL register is a write-only register that can only be written in 8-bit units

Address: RSCFDnCFDCFPCTRk: <RSCFDn_base> + $01D8_H$ + (04_H × k)

RSCFDnCFDCFPCTRkL: <RSCFDn_base> + $01D8_H$ + (04_H × k)

RSCFDnCFDCFPCTRkLL: <RSCFDn_base> + $01D8_H$ + (04_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.134 RSCFDnCFDCFPCTRk Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be the value after reset.
7 to 0	CFPC[7:0]	Transmit/Receive FIFO Pointer Control • Receive mode: Writing FF_H to these bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. • Transmit mode: Writing FF_H to these bits moves the write pointer to the next stage of the transmit/receive FIFO buffer. • Gateway mode: Setting prohibited

CFPC[7:0] Bits

- Receive mode (CFM[1:0] value in the RSCFDnCFDCFCCk register is 00B):
  Writing FFH to the CFPC[7:0] bits moves the read pointer to the next unread message in the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value (transmit/receive FIFO message counter) in the RSCFDnCFDCFSTSk register is decremented by 1. Read the RSCFDnCFDCFIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDnCFDCFDFd_k registers to read messages from the transmit/receive FIFO buffer, and then write FFH to the CFPC[7:0] bits.
  When writing FF_H to these bits, make sure that the CFE bit in the RSCFDnCFDCFCCk register is set to 1 (transmit/receive FIFO buffers are used) and the CFEMP flag in the RSCFDnCFDCFSTSk register is 0 (the transmit/receive FIFO buffer contains messages).

- Transmit mode (CFM[1:0] value in the RSCFDnCFDCFCCk register is 01B):
  Writing FFH to the CFPC[7:0] bits stores the data written to the RSCFDnCFDCFIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDnCFDCFDFd_k registers in the transmit/receive FIFO buffer and moves the write pointer to the next stage of the transmit/receive FIFO buffer. At this time, the CFMC[7:0] value is incremented by 1. Write transmit messages to the RSCFDnCFDCFIDk, RSCFDnCFDCFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDnCFDCFDFd_k registers before writing FFH to the CFPC[7:0] bits.
  When writing FF_H to these bits, make sure that the CFE bit in the RSCFDnCFDCFCCk register is set to 1 and the CFFLL flag in the RSCFDnCFDCFSTSk register is 0 (the transmit/receive FIFO buffer is not full).
- Gateway mode (CFM[1:0] value in the RSCFDnCFDCFCCk register is 10_B):
  Setting prohibited

23.4.8.4 RSCFDnCFDCFIDk — Transmit/receive FIFO Buffer Access ID Register (k = 0 to 8)

Access: RSCFDnCFDCFIDk register can be read/written in 32-bit units
RSCFDnCFDCFIDkL, RSCFDnCFDCFIDkH registers can be read/written in 16-bit units
RSCFDnCFDCFIDkLL, RSCFDnCFDCFIDkLH, RSCFDnCFDCFIDkHL, RSCFDnCFDCFIDkHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCFIDk: <RSCFDn_base> + 3400_H + (80_H × k)

RSCFDnCFDCFIDkL: <RSCFDn_base> + 3400_H + (80_H × k),
RSCFDnCFDCFIDkH: <RSCFDn_base> + 3402_H + (80_H × k)

RSCFDnCFDCFIDkLL: <RSCFDn_base> + 3400_H + (80_H × k),
RSCFDnCFDCFIDkLH: <RSCFDn_base> + 3401_H + (80_H × k),
RSCFDnCFDCFIDkHL: <RSCFDn_base> + 3402_H + (80_H × k),
RSCFDnCFDCFIDkHH: <RSCFDn_base> + 3403_H + (80_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFIDE	CFRTR	THLEN	CFID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.135 RSCFDnCFDCFIDk Register Contents

Bit Position	Bit Name	Function
31	CFIDE	Transmit/Receive FIFO Buffer IDE 0: Standard ID 1: Extended ID
30	CFRTR	Transmit/Receive FIFO Buffer RTR/RRS • When the CFM[1:0] value is 01_B (transmit mode) – When the transmit message is a classical CAN frame 0: Data frame 1: Remote frame – When the transmit message is a CAN FD frame Write 0 to this bit. • When the CFM[1:0] value is 00_B (receive mode) – When the received message is a classical CAN frame 0: Data frame 1: Remote frame – When the received message is a CAN FD frame The RRS bit value of the received message can be read.
29	THLEN	Transmit History Data Store Enable This bit is valid only when the CFM[1:0] value is 01_B (transmit mode). 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	CFID[28:0]	Transmit/Receive FIFO Buffer ID Data • When CFM[1:0] value is 01_B (transmit mode): Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11. • When CFM[1:0] value is 00_B (receive mode): Standard ID or extended ID in the received message can be read. For standard ID, read bits 10 to 0. Bits 28 to 11 are read as 0.

This register is writable only when the CFM[1:0] value in the RSCFDnCFDCFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). This RSCFDnCFDCFIDk register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFIDE Bit

This bit indicates the ID format (standard ID or extended ID) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, these bits are used to set the ID format of the message to be transmitted from the transmit/receive FIFO buffer.

CFRTR Bit

If the received message is a classical CAN frame, this bit indicates the data format (data frame or remote frame) of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. If the received message is a CAN FD frame, this bit indicates the RRS bit value of the received message.

When the CFM[1:0] value is 01_B, this bit is used to set the data format of the message to be transmitted from the transmit/receive FIFO buffer.

When the CFFDF bit in the RSCFDnCFDCFFDCSTSk register is 1 (CAN FD frame), set this bit to 0.

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, buffer type, and timestamp) of transmit messages is stored in the transmit history buffer after transmission is completed.

This bit is enabled when the CFM[1:0] value is 01_B (transmit mode).

CFID[28:0] Bits

These bits contain the ID of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

When the CFM[1:0] value is 01_B, this bit is used to set the ID of the message to be transmitted from the transmit/receive FIFO buffer.

23.4.8.5 RSCFDnCFDCFPTRk — Transmit/receive FIFO Buffer Access Pointer Register (k = 0 to 8)

Access: RSCFDnCFDCFPTRk register can be read/written in 32-bit units
RSCFDnCFDCFPTRkL, RSCFDnCFDCFPTRkH registers can be read/written in 16-bit units
RSCFDnCFDCFPTRkLL, RSCFDnCFDCFPTRkLH, RSCFDnCFDCFPTRkHL, RSCFDnCFDCFPTRkHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCFPTRk: <RSCFDn_base> + 3404_H + (80_H × k)

RSCFDnCFDCFPTRkL: <RSCFDn_base> + 3404_H + (80_H × k),
RSCFDnCFDCFPTRkH: <RSCFDn_base> + 3406_H + (80_H × k)

RSCFDnCFDCFPTRkLL: <RSCFDn_base> + 3404_H + (80_H × k),
RSCFDnCFDCFPTRkLH: <RSCFDn_base> + 3405_H + (80_H × k),
RSCFDnCFDCFPTRkHL: <RSCFDn_base> + 3406_H + (80_H × k),
RSCFDnCFDCFPTRkHH: <RSCFDn_base> + 3407_H + (80_H × k)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDLC[3:0]				CFPTR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.136 RSCFDnCFDCFPTRk Register Contents

Bit Position	Bit Name	Function					
31 to 28	CFDLC[3:0]	Transmit/Receive FIFO Buffer DLC Data					
		b31	b30	b29	b28	Classical CAN Frame	CAN FD Frame
		0	0	0	0	0 data bytes	
		0	0	0	1	1 data byte	
		0	0	1	0	2 data bytes	
		0	0	1	1	3 data bytes	
		0	1	0	0	4 data bytes	
		0	1	0	1	5 data bytes	
		0	1	1	0	6 data bytes	
		0	1	1	1	7 data bytes	
		1	0	0	0	8 data bytes	
		1	0	0	1	8 data bytes	12 data bytes
		1	0	1	0		16 data bytes
		1	0	1	1		20 data bytes
		1	1	0	0		24 data bytes
		1	1	0	1		32 data bytes
		1	1	1	0		48 data bytes
		1	1	1	1		64 data bytes

Table 23.136 RSCFDnCFDCFPTRk Register Contents

Bit Position	Bit Name	Function
27 to 16	CFPTR[11:0]	Transmit/Receive FIFO Buffer Label Data • When CFM[1:0] value is 01_B (transmit mode): Set the label information to be stored in the transmit history buffer. Only bits CFPTR[7:0] are valid. • When CFM[1:0] value is 00_B (receive mode): The label information of the received message can be read.
15 to 0	CFTS[15:0]	Transmit/Receive FIFO Buffer Timestamp Data These bits are valid only when the CFM[1:0] value is 00_B (receive mode). The timestamp value of the received message can be read.

This register is writable only when the CFM[1:0] value in the RSCFDnCFDCFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

CFDLC[3:0] Bits

These bits indicate the data length of the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B.

When the CFM[1:0] value is 01_B, these bits are used to set the data length of the message to be transmitted from the transmit/receive FIFO buffer.

When the CFDLC[3:0] bits are set to 1001_B or more while the CFFDF bit in the RSCFDnCFDCFFDCSTSk register is 0 (CAN frame), 8-byte data is transmitted actually. When the CFFDF bit is 1 (CAN FD frame), the settable value range varies depending on the settings of the TMME bit in the RSCFDnCFDCmFDCFG register and the CFPLS[2:0] bits in the RSCFDnCFDCFCCk register.

- When TMME bit = 0 (transmit buffer merge mode disabled):
  A value of 0000_B to 1111_B is settable. If the specified data length exceeds the payload storage size specified by the CFPLS[2:0] bits, excessive payloads are padded by CCH.
- When TMME bit = 1 (transmit buffer merge mode enabled):
  Set the data length within the payload storage size specified by the CFPLS[2:0] bits.

CFPTR[11:0] Bits

These bits indicate the label information attached to the received message stored in the transmit/receive FIFO buffer when the CFM[1:0] value is 00_B. When the CFM[1:0] value is 01_B, the CFPTR[7:0] value is stored in the transmit history buffer when message transmission has been completed.

CFTS[15:0] Bits

These bits indicate the timestamp value of the message stored in the transmit/receive FIFO buffer.

These bits are valid when the CFM[1:0] value is 00_B.

23.4.8.6 RSCFDnCFDCFFDCSTSk — Transmit/Receive FIFO CAN FD Configuration/Status Register (k = 0 to 8)

Access: RSCFDnCFDCFFDCSTSk register can be read/written in 32-bit units
RSCFDnCFDCFFDCSTSkL register can be read/written in 16-bit units
RSCFDnCFDCFFDCSTSkLL register can be read/written in 8-bit units

Address: RSCFDnCFDCFFDCSTSk: <RSCFDn_base> + 3408_H + (80_H × k)

RSCFDnCFDCFFDCSTSkL: <RSCFDn_base> + 3408_H + (80_H × k)

RSCFDnCFDCFFDCSTSkLL: <RSCFDn_base> + 3408_H + (80_H × k)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	CFFDF	CFBRS	CFESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 23.137 RSCFDnCFDCFFDCSTSk Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	These bits are read as the value after reset The write value should be the value after reset.
2	CFFDF	FDF 0: Classical CAN frame 1: CAN FD frame
1	CFBRS	BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes.
0	CFESI	ESI 0: Error active node 1: Error passive node

This register is writable only when the CFM[1:0] value in the RSCFDnCFDCFCCk register is 01_B (transmit mode). This register is readable only when the CFM[1:0] value is 00_B (receive mode). Do not read or write this register when the CFM[1:0] value is 10_B (gateway mode).

CFFDF Bit

When the CFM[1:0] value is 00_B, this bit indicates the FD format (classical CAN frame or CAN FD frame) of the message stored in the transmit/receive FIFO buffer. When the CFM[1:0] value is 01_B, this bit is used to set the FD format of the message to be transmitted from the transmit/receive FIFO buffer.

CFBRS Bit

When the CFM[1:0] value is 00_B, if the CFFDF bit is 1, this bit indicates the BRS bit value of the message stored in the transmit/receive FIFO buffer. If the CFFDF bit is 0, this bit is always read as 0.

When the CFM[1:0] value is 01_B, if the CFFDF bit is 1, this bit is used to set the BRS bit value of the message to be transmitted from the transmit/receive FIFO buffer. If the CFFDF bit is 0, write 0 to this bit.

CFESI Bit

When the CFM[1:0] value is 00B, if the CFFDF bit is 1, this bit indicates the ESI bit value of the message stored in the transmit/receive FIFO buffer. If the CFFDF bit is 0, this bit is always read as 0.

When the CFM[1:0] value is 01B, if the CFFDF bit is 1, this bit is used to set the ESI bit value of the message to be transmitted from the transmit/receive FIFO buffer. The set value is transmitted when the ESIC bit in the RSCFDnCFDCmFDCFG register is 1 and the channel is in the error active state. When the channel is in the error passive state, the ESI bit value that shows an error passive node is transmitted regardless of this bit value. When the CFFDF bit is 0, write 0 to this bit.

23.4.8.7 RSCFDnCFDCFDFd_k — Transmit/receive FIFO Buffer Access Data Field d Register (d = 0 to 15, k = 0 to 8)

Access: RSCFDnCFDCFDFd_k register can be read/written in 32-bit units
RSCFDnCFDCFDFd_kL, RSCFDnCFDCFDFd_kH registers can be read/written in 16-bit units
RSCFDnCFDCFDFd_kLL, RSCFDnCFDCFDFd_kLH, RSCFDnCFDCFDFd_kHL, RSCFDnCFDCFDFd_kHH registers can be read/written in 8-bit units

Address: RSCFDnCFDCFDFd_k: <RSCFDn_base> + $340C_H$ + (04_H × d) + (80_H × k)

RSCFDnCFDCFDFd_kL: <RSCFDn_base> + $340C_H$ + (04_H × d) + (80_H × k),
RSCFDnCFDCFDFd_kH: <RSCFDn_base> + $340E_H$ + (04_H × d) + (80_H × k)

RSCFDnCFDCFDFd_kLL: <RSCFDn_base> + $340C_H$ + (04_H × d) + (80_H × k),
RSCFDnCFDCFDFd_kLH: <RSCFDn_base> + $340D_H$ + (04_H × d) + (80_H × k),
RSCFDnCFDCFDFd_kHL: <RSCFDn_base> + $340E_H$ + (04_H × d) + (80_H × k),
RSCFDnCFDCFDFd_kHH: <RSCFDn_base> + $340F_H$ + (04_H × d) + (80_H × k)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CFDB4 × d + 3 [7:0]								CFDB4 × d + 2 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFDB4 × d + 1 [7:0]								CFDB4 × d + 0 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.138 RSCFDnCFDCFDFd_k Register Contents

Bit Position	Bit Name	Function
31 to 24	CFDB4 × d + 3 [7:0]	Transmit/Receive FIFO Buffer Data Byte 4 × d + 3 Transmit/Receive FIFO Buffer Data Byte 4 × d + 2 Transmit/Receive FIFO Buffer Data Byte 4 × d + 1 Transmit/Receive FIFO Buffer Data Byte 4 × d + 0 • When CFM[1:0] value is 01B (transmit mode): Set the transmit/receive FIFO buffer data. • When CFM[1:0] value is 00B (receive mode): The message data stored in the transmit/receive FIFO buffer can be read.
23 to 16	CFDB4 × d + 2 [7:0]	
15 to 8	CFDB4 × d + 1 [7:0]	
7 to 0	CFDB4 × d + 0 [7:0]	

This register is writable only when the CFM[1:0] value in the RSCFDnCFDCFCCk register is 01_B (transmit mode).

This register is readable only when the CFM[1:0] value is 00_B (receive mode). When the CFDLC[3:0] value in the RSCFDnCFDCFPTRk register is smaller than the payload storage size of the transmit/receive FIFO buffer, data bytes for which no data is set are read as 00_H.

Specify the payload storage size of the transmit/receive FIFO buffer by the CFPLS[2:0] bits in the RSCFDnCFDCFCCk register. Do not read or write the RSCFDnCFDCFDFd_k register corresponding to an area larger than the specified size.

This register should not be read or written when the CFM[1:0] value is 10_B (gateway mode).

23.4.9 Details of FIFO Status-related Registers

23.4.9.1 RSCFDnCFDFESTS — FIFO Empty Status Register

Access: RSCFDnCFDFESTSregisters can be read only in 32-bit units
RSCFDnCFDFESTSL, RSCFDnCFDFESTSH registers can be read only in 16-bit units
RSCFDnCFDFESTSLL, RSCFDnCFDFESTSLH, RSCFDnCFDFESTSHL registers can be read only in 8-bit units

Address: RSCFDnCFDFESTS: <RSCFDn_base> + 0238_H

RSCFDnCFDFESTSL: <RSCFDn_base> + 0238_H,
RSCFDnCFDFESTSH: <RSCFDn_base> + $023A_H$

RSCFDnCFDFESTSLL: <RSCFDn_base> + 0238_H,
RSCFDnCFDFESTSLH: <RSCFDn_base> + 0239_H,
RSCFDnCFDFESTSHL: <RSCFDn_base> + $023A_H$

Value after reset: 03FF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF8EMP
Value after reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7EMP	CF6EMP	CF5EMP	CF4EMP	CF3EMP	CF2EMP	CF1EMP	CF0EMP	RF7EMP	RF6EMP	RF5EMP	RF4EMP	RF3EMP	RF2EMP	RF1EMP	RF0EMP
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.139 RSCFDnCFDFESTS Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	These bits are read as the value after reset.
16	CF8EMP	Transmit/Receive FIFO Buffer Empty Status Flag 0: Transmit/receive FIFO buffer k contains a message. 1: Transmit/receive FIFO buffer k contains no message. (k = 0 to 8)
15	CF7EMP	
14	CF6EMP	
13	CF5EMP	
12	CF4EMP	
11	CF3EMP	
10	CF2EMP	
9	CF1EMP	
8	CF0EMP	
7	RF7EMP	Receive FIFO Buffer Empty Status Flag 0: Receive FIFO buffer x contains an unread message. 1: Receive FIFO buffer x contains no unread message (buffer empty). (x = 0 to 7)
6	RF6EMP	
5	RF5EMP	
4	RF4EMP	
3	RF3EMP	
2	RF2EMP	
1	RF1EMP	
0	RF0EMP	

The RSCFDnCFDFESTS register is set to 03FF $FFFF_H$ in global reset mode.

CFkEMP Flag (k = 0 to 8)

The CFkEMP flag is set to 1 when the CFEMP flag in the RSCFDnCFDCFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)). When the CFEMP flag is cleared to 0 (the transmit/receive FIFO buffer contains messages), the CFkEMP flag is cleared to 0.

RFxEMP Flag (x = 0 to 7)

The RFxEMP flag is set to 1 when the RFEMP flag in the RSCFDnCFDRFSTSx register is set to 1 (the receive FIFO buffer contains no unread message). When the RFEMP flag is cleared to 0 (the receive FIFO buffer contains unread messages), the RFxEMP flag is cleared to 0.

23.4.9.2 RSCFDnCFDFFSTS — FIFO Full Status Register

Access: RSCFDnCFDFFSTS register can be read only in 32-bit units
RSCFDnCFDFFSTSL, RSCFDnCFDFFSTSH registers can be read only in 16-bit units
RSCFDnCFDFFSTSLL, RSCFDnCFDFFSTSLH, RSCFDnCFDFFSTSHL registers can be read only in 8-bit units

Address: RSCFDnCFDFFSTS: <RSCFDn_base> + $023C_H$

RSCFDnCFDFFSTSL: <RSCFDn_base> + $023C_H$,
RSCFDnCFDFFSTSH: <RSCFDn_base> + $023E_H$

RSCFDnCFDFFSTSLL: <RSCFDn_base> + $023C_H$,
RSCFDnCFDFFSTSLH: <RSCFDn_base> + $023D_H$,
RSCFDnCFDFFSTSHL: <RSCFDn_base> + $023E_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF8FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7FLL	CF6FLL	CF5FLL	CF4FLL	CF3FLL	CF2FLL	CF1FLL	CF0FLL	RF7FLL	RF6FLL	RF5FLL	RF4FLL	RF3FLL	RF2FLL	RF1FLL	RF0FLL
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.140 RSCFDnCFDFFSTS Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	These bits are read as the value after reset.
16	CF8FLL	Transmit/Receive FIFO Buffer Full Status Flag 0: Transmit/receive buffer k is not full. 1: Transmit/receive buffer k is full. (k = 0 to 8)
15	CF7FLL	
14	CF6FLL	
13	CF5FLL	
12	CF4FLL	
11	CF3FLL	
10	CF2FLL	
9	CF1FLL	
8	CF0FLL	
7	RF7FLL	Receive FIFO Buffer Full Status Flag 0: Receive FIFO buffer x is not full. 1: Receive FIFO buffer x is full. (x = 0 to 7)
6	RF6FLL	
5	RF5FLL	
4	RF4FLL	
3	RF3FLL	
2	RF2FLL	
1	RF1FLL	
0	RF0FLL	

The RSCFDnCFDFFSTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkFLL Flag (k = 0 to 8)

The CFkFLL flag is set to 1 when the CFFLL flag in the RSCFDnCFDCFSTSk register is set to 1 (the transmit/receive FIFO buffer is full).
When the CFFLL flag is cleared to 0 (the transmit/receive FIFO buffer is not full), the CFkFLL flag is cleared to 0.

RFxFLL Flag (x = 0 to 7)

The RFxFLL flag is set to 1 when the RFFLL flag in the RSCFDnCFDRFSTSx register is set to 1 (the receive FIFO buffer is full). When the RFFLL flag is cleared to 0 (the receive FIFO buffer is not full), the RFxFLL flag is cleared to 0.

23.4.9.3 RSCFDnCFDFMSTS — FIFO Message Lost Status Register

Access: RSCFDnCFDFMSTS register can be read only in 32-bit units
RSCFDnCFDFMSTSL, RSCFDnCFDFMSTSH registers can be read only in 16-bit units
RSCFDnCFDFMSTSLL, RSCFDnCFDFMSTSLH, RSCFDnCFDFMSTSHL registers can be read only in 8-bit units

Address: RSCFDnCFDFMSTS: <RSCFDn_base> + 0240_H

RSCFDnCFDFMSTSL: <RSCFDn_base> + 0240_H,
RSCFDnCFDFMSTSH: <RSCFDn_base> + 0242_H

RSCFDnCFDFMSTSLL: <RSCFDn_base> + 0240_H,
RSCFDnCFDFMSTSLH: <RSCFDn_base> + 0241_H,
RSCFDnCFDFMSTSHL: <RSCFDn_base> + 0242_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CF8MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CF7MLT	CF6MLT	CF5MLT	CF4MLT	CF3MLT	CF2MLT	CF1MLT	CF0MLT	RF7MLT	RF6MLT	RF5MLT	RF4MLT	RF3MLT	RF2MLT	RF1MLT	RF0MLT
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.141 RSCFDnCFDFMSTS Register Contents

Bit Position	Bit Name	Function
31 to 17	Reserved	These bits are read as the value after reset.
16	CF8MLT	Transmit/Receive FIFO Buffer Message Lost Status Flag 0: No transmit/receive FIFO buffer k message is lost. 1: A transmit/receive FIFO buffer k message is lost. (k = 0 to 8)
15	CF7MLT	
14	CF6MLT	
13	CF5MLT	
12	CF4MLT	
11	CF3MLT	
10	CF2MLT	
9	CF1MLT	
8	CF0MLT	
7	RF7MLT	Receive FIFO Buffer Message Lost Status Flag 0: No receive FIFO buffer x message is lost. 1: A receive FIFO buffer x message is lost. (x = 0 to 7)
6	RF6MLT	
5	RF5MLT	
4	RF4MLT	
3	RF3MLT	
2	RF2MLT	
1	RF1MLT	
0	RF0MLT	

The RSCFDnCFDFMSTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkMLT Flag (k = 0 to 8)

The CFkMLT flag is set to 1 when the CFMLT flag in the RSCFDnCFDCFSTSk register is set to 1 (a transmit/receive FIFO message is lost). When the CFMLT flag is cleared to 0, the CFkMLT flag is cleared to 0.

RFxMLT Flag (x = 0 to 7)

The RFxMLT flag is set to 1 when the RFMLT flag in the RSCFDnCFDRFSTSx register is set to 1 (a receive FIFO message is lost). When the RFMLT flag is cleared to 0, the RFxMLT flag is cleared to 0.

23.4.9.4 RSCFDnCFDRFISTS — Receive FIFO Buffer Interrupt Flag Status Register

Access: RSCFDnCFDRFISTS register can be read only in 32-bit units
RSCFDnCFDRFISTSL register is a read-only register that can be read only in 16-bit units
RSCFDnCFDRFISTSLL register is a read-only register that can be read only in 8-bit units

Address: RSCFDnCFDRFISTS: <RSCFDn_base> + 0244_H

RSCFDnCFDRFISTSL: <RSCFDn_base> + 0244_H

RSCFDnCFDRFISTSLL: <RSCFDn_base> + 0244_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	RF7IF	RF6IF	RF5IF	RF4IF	RF3IF	RF2IF	RF1IF	RF0IF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.142 RSCFDnCFDRFISTS Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	These bits are read as the value after reset.
7	RF7IF	Receive FIFO Buffer Interrupt Request Status Flag 0: No receive FIFO buffer x interrupt request is present. 1: A receive FIFO buffer x interrupt request is present. (x = 0 to 7)
6	RF6IF	
5	RF5IF	
4	RF4IF	
3	RF3IF	
2	RF2IF	
1	RF1IF	
0	RF0IF	

The RSCFDnCFDRFISTS register is cleared to $0000\ 0000_H$ in global reset mode.

RFxIF Flag (x = 0 to 7)

The RFxIF flag is set to 1 when the RFIF flag in the RSCFDnCFDRFSTSx register is set to 1 (a receive FIFO interrupt request is present). When the RFIF flag is cleared to 0, the RFxIF flag is cleared to 0.

23.4.9.5 RSCFDnCFDCFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register

Access: RSCFDnCFDCFRISTS register can be read only in 32-bit units
RSCFDnCFDCFRISTSL register is a read-only register that can be read only in 16-bit units
RSCFDnCFDCFRISTSLL, RSCFDnCFDCFRISTSLH registers are the read-only registers that can be read only in 8-bit units

Address: RSCFDnCFDCFRISTS: <RSCFDn_base> + 0248_H

RSCFDnCFDCFRISTSL: <RSCFDn_base> + 0248_H

RSCFDnCFDCFRISTSLL: <RSCFDn_base> + 0248_H,
RSCFDnCFDCFRISTSLH: <RSCFDn_base> + 0249_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CF8RXIF	CF7RXIF	CF6RXIF	CF5RXIF	CF4RXIF	CF3RXIF	CF2RXIF	CF1RXIF	CF0RXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.143 RSCFDnCFDCFRISTS Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are read as the value after reset.
8	CF8RXIF	Transmit/Receive FIFO Buffer Receive Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k receive interrupt request is present. 1: A transmit/receive FIFO buffer k receive interrupt request is present. (k = 0 to 8)
7	CF7RXIF	
6	CF6RXIF	
5	CF5RXIF	
4	CF4RXIF	
3	CF3RXIF	
2	CF2RXIF	
1	CF1RXIF	
0	CF0RXIF	

The RSCFDnCFDCFRISTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkRXIF Flag (k = 0 to 8)

The CFkRXIF flag is set to 1 when the CFRXIF flag in the RSCFDnCFDCFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). When the CFRXIF flag is cleared to 0, the CFkRXIF flag is cleared to 0.

23.4.9.6 RSCFDnCFDCFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register

Access: RSCFDnCFDCFTISTS register can be read only in 32-bit units
RSCFDnCFDCFTISTSL register is a read-only register that can be read only in 16-bit units
RSCFDnCFDCFTISTSLL, RSCFDnCFDCFTISTSLH registers are the read-only registers that can be read only in 8-bit units

Address: RSCFDnCFDCFTISTS: <RSCFDn_base> + $024C_H$

RSCFDnCFDCFTISTSL: <RSCFDn_base> + $024C_H$

RSCFDnCFDCFTISTSLL: <RSCFDn_base> + $024C_H$,
RSCFDnCFDCFTISTSLH: <RSCFDn_base> + $024D_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	CF8TXIF	CF7TXIF	CF6TXIF	CF5TXIF	CF4TXIF	CF3TXIF	CF2TXIF	CF1TXIF	CF0TXIF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.144 RSCFDnCFDCFTISTS Register Contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are read as the value after reset.
8	CF8TXIF	Transmit/Receive FIFO Buffer Transmit Interrupt Request Status Flag 0: No transmit/receive FIFO buffer k transmit interrupt request is present. 1: A transmit/receive FIFO buffer k transmit interrupt request is present. (k = 0 to 8)
7	CF7TXIF	
6	CF6TXIF	
5	CF5TXIF	
4	CF4TXIF	
3	CF3TXIF	
2	CF2TXIF	
1	CF1TXIF	
0	CF0TXIF	

The RSCFDnCFDCFTISTS register is cleared to $0000\ 0000_H$ in global reset mode.

CFkTXIF Flag (k = 0 to 8)

The CFkTXIF flag is set to 1 when the CFTXIF flag in the RSCFDnCFDCFSTSk register is set to 1 (a transmit/receive FIFO transmit interrupt request is present). When the CFTXIF flag is cleared to 0, the CFkTXIF flag is cleared to 0.

23.4.10 Details of Transmit Buffer-related Registers

23.4.10.1 RSCFDnCFDTMCp — Transmit Buffer Control Register (p = 0 to 47)

Access: RSCFDnCFDTMCp register can be read/written in 8-bit units

Address: RSCFDnCFDTMCp: <RSCFDn_base> + 0250_H + (01_H × p)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W*1	R/W*1

Note 1. The only effective value for writing to this bit is 1, which sets the bit. Otherwise writing to the bit results in retention of its state.

Table 23.145 RSCFDnCFDTMCp Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2	TMOM	One-Shot Transmission Enable 0: One-shot transmission is disabled. 1: One-shot transmission is enabled.
1	TMTAR	Transmit Abort Request 0: Transmit abort is not requested. 1: Transmit abort is requested.
0	TMTR	Transmit Request 0: Transmission is not requested. 1: Transmission is requested.

When the RSCFDnCFDTMCp register meets any of the following conditions, set it to 00_H.

- The RSCFDnCFDTMCp register corresponds to the transmit buffer number selected by the CFTML[3:0] bits in the RSCFDnCFDCFCCk register (p = m × 16 + the value of CFTML[3:0] bits).
- The RSCFDnCFDTMCp register corresponds to the transmit buffer allocated to the transmit queue by the TXQDC[3:0] bits in the RSCFDnCFDTXQCCm (m = 0 to 2) register (p = (m × 16 + 15) to (m × 16 + 15 – the value of TXQDC[3:0] bits)).
- RSCFDnCFDTMCp register (p = (m × 16) + 1, (m × 16) + 2, (m × 16) + 4, or (m × 16) + 5) corresponding to the transmit buffer allocated as a payload storage area when the TMME bit in the RSCFDnCFDCmFDCFG register is 1 (transmit buffer merge mode) Bits in the RSCFDnCFDTMCp register are all cleared to 0 in channel reset mode. Modify the RSCFDnCFDTMCp register in channel communication mode or channel halt mode.

TMOM Bit

Setting this bit to 1 enables one-shot transmission. When transmission fails, retransmission defined in the CAN protocol is not performed.

Modify the TMOM bit when the TMTRM flag in the RSCFDnCFDTMSTSp register is set to 0. Set the TMOM bit to 1 together with the TMTR bit.

TMTAR Bit

Setting this bit to 1 generates a transmit abort request for the message stored in the transmit buffer. However, a message that is being transmitted or one that will be transmitted next cannot be aborted.

The TMTAR bit can be set to 1 when TMTR bit is 1.

The TMTAR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed.
- An error or arbitration-lost has been detected.

If this bit becomes 0 at the same time as the program writes 1 to this bit, this bit becomes 0.

TMTR Bit

Setting this bit to 1 transmits the message stored in the transmit buffer.

The TMTR bit is cleared to 0 when any of the following conditions is met, but cannot be cleared by the program writing 0 to the bit.

- Transmission has been completed.
- Transmit abort has been completed after the TMTAR bit was set to 1.
- An error or arbitration-lost has been detected with the TMOM bit set to 1.

Set the TMTR bit to 1 when the value of TMTRF[1:0] in the RSCFDnCFDTMSTSp register is 00_B.

23.4.10.2 RSCFDnCFDTMSTSp — Transmit Buffer Status Register (p = 0 to 47)

Access: RSCFDnCFDTMSTSp register can be read/written in 8-bit units

Address: RSCFDnCFDTMSTSp: <RSCFDn_base> + $02D0_H$ + (01_H × p)

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R

Table 23.146 RSCFDnCFDTMSTSp Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
4	TMTARM	Transmit Buffer Transmit Abort Request Status Flag 0: No transmit abort request is present. 1: A transmit abort request is present.
3	TMTRM	Transmit Buffer Transmit Request Status Flag 0: No transmit request is present. 1: A transmit request is present.
2, 1	TMTRF[1:0]	Transmit Buffer Transmit Result Status Flag b2 b1 0 0: Transmission is in progress or no transmit request is present. 0 1: Transmit abort has been completed. 1 0: Transmission has been completed (without transmit abort request). 1 1: Transmission has been completed (with transmit abort request).
0	TMTSTS	Transmit Buffer Transmit Status Flag 0: Transmission is not in progress. 1: Transmission is in progress.

The RSCFDnCFDTMSTSp register is cleared to all 0 in channel reset mode.

TMTARM Flag

The TMTARM flag is set to 1 when the TMTAR bit in the RSCFDnCFDTMCp register is set to 1.

The TMTARM flag is set to 0 when the TMTAR bit in the RSCFDnCFDTMCp register is set to 0.

TMTRM Flag

The TMTRM flag is set to 1 when the TMTR bit in the RSCFDnCFDTMCp register is set to 1.

The TMTRM flag is set to 0 when the TMTR bit in the RSCFDnCFDTMCp register is set to 0.

TMTRF[1:0] Flag

This flag indicates the result of transmission from the transmit buffer.

00_B: Transmission is in progress or no transmit request is present.

01_B: Transmission from the transmit buffer was aborted.

10_B: Transmission has been completed with the TMTAR bit in the RSCFDnCFDTMCp register set to 0 (transmit abort is not requested).

11_B: Transmission has been completed with the TMTAR bit in the RSCFDnCFDTMCp register set to 1 (transmit abort is requested).

Write 00_B to the TMTRF[1:0] flag in channel communication mode or channel halt mode. Do not write any value other than 00_B to this flag.

TMTSTS Flag

This flag is set to 1 when transmission from the transmit buffer starts, and is cleared to 0 when transmission from the transmit buffer has been completed or terminated due to a bus error or arbitration lost.

23.4.10.3 RSCFDnCFDTMIDp — Transmit Buffer ID Register (p = 0 to 47)

Access: RSCFDnCFDTMIDp register can be read/written in 32-bit units
RSCFDnCFDTMIDpL, RSCFDnCFDTMIDpH registers can be read/written in 16-bit units
RSCFDnCFDTMIDpLL, RSCFDnCFDTMIDpLH, RSCFDnCFDTMIDpHL, RSCFDnCFDTMIDpHH registers can be read/written in 8-bit units

Address: RSCFDnCFDTMIDp: <RSCFDn_base> + 4000_H + (20_H × p)

RSCFDnCFDTMIDpL: <RSCFDn_base> + 4000_H + (20_H × p),
RSCFDnCFDTMIDpH: <RSCFDn_base> + 4002_H + (20_H × p)

RSCFDnCFDTMIDpLL: <RSCFDn_base> + 4000_H + (20_H × p),
RSCFDnCFDTMIDpLH: <RSCFDn_base> + 4001_H + (20_H × p),
RSCFDnCFDTMIDpHL: <RSCFDn_base> + 4002_H + (20_H × p),
RSCFDnCFDTMIDpHH: <RSCFDn_base> + 4003_H + (20_H × p)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIDE	TMRTR	THLEN	TMID[28:16]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMID[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.147 RSCFDnCFDTMIDp Register Contents

Bit Position	Bit Name	Function
31	TMIDE	Transmit Buffer IDE 0: Standard ID 1: Extended ID
30	TMRTR	Transmit Buffer RTR/RRS • When the transmit message is a classical CAN frame 0: Data frame 1: Remote frame • When the transmit message is a CAN FD frame Write 0 to this bit.
29	THLEN	Transmit History Data Store Enable 0: Transmit history data is not stored in the buffer. 1: Transmit history data is stored in the buffer.
28 to 0	TMID[28:0]	Transmit Buffer ID Data Set standard ID or extended ID. For standard ID, write an ID to bits 10 to 0 and write 0 to bits 28 to 11.

Modify this register when the TMTRM bit in the corresponding RSCFDnCFDTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write data to this register. If this register is allocated to the transmit queue, only write data to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMIDE Bit

This bit is used to set the ID format of the message to be transmitted from the transmit buffer.

TMRTR Bit

This bit is used to set the data format of the message to be transmitted from the transmit buffer.

Set this bit to 0 when the TMFDF bit in the RSCFDnCFDTMFDCTRp register is 1 (CAN FD frame).

THLEN Bit

When this bit is set to 1, the transmit history data (label information, buffer number, buffer type, and timestamp) of transmit messages is stored in the transmit history buffer after transmission is completed.

TMID[28:0] Bits

These bits are used to set the ID of the message to be transmitted from the transmit buffer.

23.4.10.4 RSCFDnCFDTMPTRp — Transmit Buffer Pointer Register (p = 0 to 47)

Access: RSCFDnCFDTMPTRp register can be read/written in 32-bit units
RSCFDnCFDTMPTRpH register can be read/written in 16-bit units
RSCFDnCFDTMPTRpHL, RSCFDnCFDTMPTRpHH registers can be read/written in 8-bit units

Address: RSCFDnCFDTMPTRp: <RSCFDn_base> + 4004_H + (20_H × p)

RSCFDnCFDTMPTRpH: <RSCFDn_base> + 4006_H + (20_H × p)

RSCFDnCFDTMPTRpHL: <RSCFDn_base> + 4006_H + (20_H × p),
RSCFDnCFDTMPTRpHH: <RSCFDn_base> + 4007_H + (20_H × p)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDLC[3:0]				—	—	—	—	TMPTR[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.148 RSCFDnCFDTMPTRp Register Contents

Bit Position	Bit Name	Function
31 to 28	TMDLC[3:0]	Transmit Buffer DLC Data
27 to 24	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
23 to 16	TMPTR[7:0]	Transmit Buffer Label Data Set the label information to be stored in the transmit history buffer.
15 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

b31	b30	b29	b28	Classical CAN Frame	CAN FD Frame
0	0	0	0	0 data bytes	
0	0	0	1	1 data byte	
0	0	1	0	2 data bytes	
0	0	1	1	3 data bytes	
0	1	0	0	4 data bytes	
0	1	0	1	5 data bytes	
0	1	1	0	6 data bytes	
0	1	1	1	7 data bytes	
1	0	0	0	8 data bytes	
1	0	0	1	8 data bytes	12 data bytes
1	0	1	0		16 data bytes
1	0	1	1		20 data bytes
1	1	0	0		24 data bytes
1	1	0	1		32 data bytes
1	1	1	0		48 data bytes
1	1	1	1		64 data bytes

Modify this register when the TMTRM bit in the corresponding RSCFDnCFDTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not

write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

TMDLC[3:0] Bits

These bits are used to set the data length of the message to be transmitted from the transmit buffer when the TMRTR bit in the RSCFDnCFDTMIDp register is set to 0 (data frame).

When the TMDLC[3:0] bits are set to 1001B or more while the TMFDF bit in the RSCFDnCFDTMFDCTRp register is 0 (classical CAN frame), 8-byte data is transmitted actually. When the TMFDF bit is 1 (CAN FD frame), the settable value range varies depending on the setting of the TMME bit in the RSCFDnCFDCmFDCFG register.

- When the TMME bit = 0 (transmit buffer merge mode disabled):
  A value of 0000_B to 1111_B is settable. If a value larger than 1100_B is set, payloads exceeding 20 bytes are padded by CCH.
- When the TMME bit = 1 (transmit buffer merge mode enabled):
  When the corresponding transmit buffer number p = (m × 16) + 0 or (m × 16) + 3, a value of 0000_B to 1111_B is settable. In other cases, set a value of 0000_B to 1011_B (20 data bytes).

When the TMRTR bit is 1 (remote frame), set the data length of a message to be requested.

TMPTR[7:0] Bits

When message transmission has been completed, the TMPTR[7:0] value is stored in the transmit history buffer.

23.4.10.5 RSCFDnCFDTMFDCTRp — Transmit Buffer CAN FD Configuration Register (p = 0 to 47)

Access: RSCFDnCFDTMFDCTRp register can be read/written in 32-bit units
RSCFDnCFDTMFDCTRpL register can be read/written in 16-bit units
RSCFDnCFDTMFDCTRpLL register can be read/written in 8-bit units

Address: RSCFDnCFDTMFDCTRp: <RSCFDn_base> + 4008_H + (20_H × p)

RSCFDnCFDTMFDCTRpL: <RSCFDn_base> + 4008_H + (20_H × p)

RSCFDnCFDTMFDCTRpLL: <RSCFDn_base> + 4008_H + (20_H × p)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TMFDF	TMBRS	TMESI
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 23.149 RSCFDnCFDTMFDCTRp Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	These bits are read as the value after reset. The write value should be the value after reset.
2	TMFDF	FDF 0: Classical CAN frame 1: CAN FD frame
1	TMBRS	BRS 0: The bit rate in the data area does not change. 1: The bit rate in the data area changes.
0	TMESI	ESI 0: Error active node 1: Error passive node

Modify this register when the TMTRM bit in the corresponding RSCFDnCFDTMSTSp register is 0 (transmission not requested). When this register is linked to the transmit/receive FIFO buffer, do not write data to this register. When this register is allocated to the transmit queue, write data only to transmit buffer p (p = m × 16 + 15) of the corresponding channel.

TMFDF Bit

This bit is used to set the FD format of the message to be transmitted from the transmit buffer.

TMBRS Bit

When this bit is set to 1 while the TMFDF bit is 1, the data area of a transmit message is transmitted at the data bit rate. When the TMFDF bit is 0, write 0 to this bit.

TMESI Bit

This bit is used to set the ESI bit value of the message to be transmitted from the transmit buffer when the TMFDF bit is 1. The set value is transmitted when the ESIC bit in the RSCFDnCFDCmFDCFG

register is 1 and the channel is in the error active state. When the channel is in the error passive state, the ESI bit value that shows an error passive node is transmitted regardless of this bit value. When the TMFDF bit is 0, write 0 to this bit.

23.4.10.6 RSCFDnCFDTMDFb_p — Transmit Buffer Data Field b Register (b = 0 to 4, p = 0 to 47)

Access: RSCFDnCFDTMDFb_p register can be read/written in 32-bit units
RSCFDnCFDTMDFb_pL, RSCFDnCFDTMDFb_pH registers can be read/written in 16-bit units
RSCFDnCFDTMDFb_pLL, RSCFDnCFDTMDFb_pLH, RSCFDnCFDTMDFb_pHL, RSCFDnCFDTMDFb_pHH registers can be read/written in 8-bit units

Address: RSCFDnCFDTMDFb_p: <RSCFDn_base> + $400C_H$ + (04_H × b) + (20_H × p)

RSCFDnCFDTMDFb_pL: <RSCFDn_base> + $400C_H$ + (04_H × b) + (20_H × p),
RSCFDnCFDTMDFb_pH: <RSCFDn_base> + $400E_H$ + (04_H × b) + (20_H × p)

RSCFDnCFDTMDFb_pLL: <RSCFDn_base> + $400C_H$ + (04_H × b) + (20_H × p),
RSCFDnCFDTMDFb_pLH: <RSCFDn_base> + $400D_H$ + (04_H × b) + (20_H × p),
RSCFDnCFDTMDFb_pHL: <RSCFDn_base> + $400E_H$ + (04_H × b) + (20_H × p),
RSCFDnCFDTMDFb_pHH: <RSCFDn_base> + $400F_H$ + (04_H × b) + (20_H × p)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMDB4 × b + 3 [7:0]								TMDB4 × b + 2 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMDB4 × b + 1 [7:0]								TMDB4 × b + 0 [7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.150 RSCFDnCFDTMDFb_p Register Contents

Bit Position	Bit Name	Function
31 to 24	TMDB4 × b + 3 [7:0]	Transmit Buffer Data Byte 4 × b + 3 Transmit Buffer Data Byte 4 × b + 2 Transmit Buffer Data Byte 4 × b + 1 Transmit Buffer Data Byte 4 × b + 0 Set the transmit buffer data.
23 to 16	TMDB4 × b + 2 [7:0]	
15 to 8	TMDB4 × b + 1 [7:0]	
7 to 0	TMDB4 × b + 0 [7:0]	

Modify this register when the TMTRM bit in the corresponding RSCFDnCFDTMSTSp register is set to 0 (no transmit request is present). If this register is linked to a transmit/receive FIFO buffer, do not write to this register. If this register is allocated to the transmit queue, only write to a transmit buffer p (p = m × 16 + 15) for the corresponding channel.

23.4.10.7 RSCFDnCFDTMIECy — Transmit Buffer Interrupt Enable Configuration Register (y = 0, 1)

Access: RSCFDnCFDTMIECy register can be read/written in 32-bit units
RSCFDnCFDTMIECyL, RSCFDnCFDTMIECyH registers can be read/written in 16-bit units
RSCFDnCFDTMIECyLL, RSCFDnCFDTMIECyLH, RSCFDnCFDTMIECyHL, RSCFDnCFDTMIECyHH registers can be read/written in 8-bit units

Address: RSCFDnCFDTMIECy: <RSCFDn_base> + 0390_H + (04_H × y)

RSCFDnCFDTMIECyL: <RSCFDn_base> + 0390_H + (04_H × y),
RSCFDnCFDTMIECyH: <RSCFDn_base> + 0392_H + (04_H × y)

RSCFDnCFDTMIECyLL: <RSCFDn_base> + 0390_H + (04_H × y),
RSCFDnCFDTMIECyLH: <RSCFDn_base> + 0391_H + (04_H × y),
RSCFDnCFDTMIECyHL: <RSCFDn_base> + 0392_H + (04_H × y),
RSCFDnCFDTMIECyHH: <RSCFDn_base> + 0393_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMIEp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMIEp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.151 RSCFDnCFDTMIECy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMIEp	Transmit Buffer Interrupt Enable p (p = y × 32 + 31 to y × 32 + 16) 0: Transmit buffer interrupt is disabled 1: Transmit buffer interrupt is enabled
15 to 0	TMIEp	Transmit Buffer Interrupt Enable p (p = y × 32 + 15 to y × 32 + 0) 0: Transmit buffer interrupt is disabled. 1: Transmit buffer interrupt is enabled.

TMIEp Bits (p = 0 to 47)

When any of these bits is set to 1 and the corresponding transmission has been completed, a transmit buffer interrupt request is generated.

Modify these bits when the TMTRM flag in the corresponding RSCFDnCFDTMSTSp register is 0 (no transmit request is present).

Write 0 to bits corresponding to transmit buffers linked to transmit/receive FIFO buffers or transmit buffers allocated to the transmit queue.

When the TMME bit in the RSCFDnCFDCmFDCFG register is 1 (transmit buffer merge mode enable), set the bit corresponding to the transmit buffer allocated as a payload storage area to 0.

Table 23.152 shows the bit assignment.

Table 23.152 TMIEp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

23.4.11 Details of Transmit Buffer Status-related Registers

23.4.11.1 RSCFDnCFDTMTRSTSy — Transmit Buffer Transmit Request Status Register (y = 0, 1)

Access: RSCFDnCFDTMTRSTSy register can be read only in 32-bit units
RSCFDnCFDTMTRSTSyL, RSCFDnCFDTMTRSTSyH registers can be read only in 16-bit units
RSCFDnCFDTMTRSTSyLL, RSCFDnCFDTMTRSTSyLH, RSCFDnCFDTMTRSTSyHL, RSCFDnCFDTMTRSTSyHH registers can be read only in 8-bit units

Address: RSCFDnCFDTMTRSTSy: <RSCFDn_base> + 0350_H + (04_H × y)

RSCFDnCFDTMTRSTSyL: <RSCFDn_base> + 0350_H + (04_H × y),
RSCFDnCFDTMTRSTSyH: <RSCFDn_base> + 0352_H + (04_H × y)

RSCFDnCFDTMTRSTSyLL: <RSCFDn_base> + 0350_H + (04_H × y),
RSCFDnCFDTMTRSTSyLH: <RSCFDn_base> + 0351_H + (04_H × y),
RSCFDnCFDTMTRSTSyHL: <RSCFDn_base> + 0352_H + (04_H × y),
RSCFDnCFDTMTRSTSyHH: <RSCFDn_base> + 0353_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTRSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTRSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.153 RSCFDnCFDTMTRSTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmit request is present. 1: A transmit request is present.
15 to 0	TMTRSTSp	Transmit Buffer Transmit Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit request is present. 1: A transmit request is present.

TMTRSTSp Flags (p = 0 to 47)

These flags indicate the status of the TMTR bit in the RSCFDnCFDTMCp register.

When the TMTR bit is set to 1 (transmission is requested), the corresponding TMTRSTSp flag is set to 1.

The corresponding TMTRSTSp flag is cleared to 0 when the TMTR bit is set to 0 (transmission is not requested) or in channel reset mode.

Table 23.154 shows the bit assignment.

Table 23.154 TMTRSTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

23.4.11.2 RSCFDnCFDTMTARSTSy — Transmit Buffer Transmit Abort Request Status Register (y = 0, 1)

Access: RSCFDnCFDTMTARSTSy register can be read only in 32-bit units
RSCFDnCFDTMTARSTSyL, RSCFDnCFDTMTARSTSyH registers can be read only in 16-bit units
RSCFDnCFDTMTARSTSyLL, RSCFDnCFDTMTARSTSyLH, RSCFDnCFDTMTARSTSyHL, RSCFDnCFDTMTARSTSyHH registers can be read only in 8-bit units

Address: RSCFDnCFDTMTARSTSy: <RSCFDn_base> + 0360_H + (04_H × y)

RSCFDnCFDTMTARSTSyL: <RSCFDn_base> + 0360_H + (04_H × y),
RSCFDnCFDTMTARSTSyH: <RSCFDn_base> + 0362_H + (04_H × y)

RSCFDnCFDTMTARSTSyLL: <RSCFDn_base> + 0360_H + (04_H × y),
RSCFDnCFDTMTARSTSyLH: <RSCFDn_base> + 0361_H + (04_H × y),
RSCFDnCFDTMTARSTSyHL: <RSCFDn_base> + 0362_H + (04_H × y),
RSCFDnCFDTMTARSTSyHH: <RSCFDn_base> + 0363_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTARSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTARSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.155 RSCFDnCFDTMTARSTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: No transmit abort request is present. 1: A transmit abort request is present.
15 to 0	TMTARSTSp	Transmit Buffer Transmit Abort Request Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: No transmit abort request is present. 1: A transmit abort request is present.

TMTARSTSp Flags (p = 0 to 47)

These flags indicate the status of the TMTAR bit in the RSCFDnCFDTMCp register.

When the TMTAR bit is set to 1 (transmit abort is requested), the corresponding TMTARSTSp flag is set to 1.

The corresponding TMTARSTSp flag is cleared to 0 when the TMTAR bit is set to 0 (transmit abort is not requested) or in channel reset mode.

Table 23.156 shows the bit assignment.

Table 23.156 TMTARSTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

23.4.11.3 RSCFDnCFDTMTCSTSy — Transmit Buffer Transmit Complete Status Register (y = 0, 1)

Access: RSCFDnCFDTMTCSTSy register can be read only in 32-bit units
RSCFDnCFDTMTCSTSyL, RSCFDnCFDTMTCSTSyH registers can be read only in 16-bit units
RSCFDnCFDTMTCSTSyLL, RSCFDnCFDTMTCSTSyLH, RSCFDnCFDTMTCSTSyHL, RSCFDnCFDTMTCSTSyHH registers can be read only in 8-bit units

Address: RSCFDnCFDTMTCSTSy: <RSCFDn_base> + 0370_H + (04_H × y)

RSCFDnCFDTMTCSTSyL: <RSCFDn_base> + 0370_H + (04_H × y),
RSCFDnCFDTMTCSTSyH: <RSCFDn_base> + 0372_H + (04_H × y)

RSCFDnCFDTMTCSTSyLL: <RSCFDn_base> + 0370_H + (04_H × y),
RSCFDnCFDTMTCSTSyLH: <RSCFDn_base> + 0371_H + (04_H × y),
RSCFDnCFDTMTCSTSyHL: <RSCFDn_base> + 0372_H + (04_H × y),
RSCFDnCFDTMTCSTSyHH: <RSCFDn_base> + 0373_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTCSTSp (p = y × 32 + 31 to y × 32 + 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTCSTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.157 RSCFDnCFDTMTCSTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission has not been completed. 1: Transmission has been completed.
15 to 0	TMTCSTSp	Transmit Buffer Transmit Complete Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission has not been completed. 1: Transmission has been completed.

TMTCSTSp Flags (p = 0 to 47)

When the TMTRF[1:0] flag in the RSCFDnCFDTMSTSp register is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)), the corresponding TMTCSTSp flag is set to 1.

A TMTCSTSp flag is cleared to 0 when the corresponding TMTRF[1:0] flag is set to 00_B or in channel reset mode.

Table 23.158 shows the bit assignment.

Table 23.158 TMTCSTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

23.4.11.4 RSCFDnCFDTMTASTSy — Transmit Buffer Transmit Abort Status Register (y = 0, 1)

Access: RSCFDnCFDTMTASTSy register can be read only in 32-bit units
RSCFDnCFDTMTASTSyL, RSCFDnCFDTMTASTSyH registers can be read only in 16-bit units
RSCFDnCFDTMTASTSyLL, RSCFDnCFDTMTASTSyLH, RSCFDnCFDTMTASTSyHL, RSCFDnCFDTMTASTSyHH registers can be read only in 8-bit units

Address: RSCFDnCFDTMTASTSy: <RSCFDn_base> + 0380_H + (04_H × y)

RSCFDnCFDTMTASTSyL: <RSCFDn_base> + 0380_H + (04_H × y),
RSCFDnCFDTMTASTSyH: <RSCFDn_base> + 0382_H + (04_H × y)

RSCFDnCFDTMTASTSyLL: <RSCFDn_base> + 0380_H + (04_H × y),
RSCFDnCFDTMTASTSyLH: <RSCFDn_base> + 0381_H + (04_H × y),
RSCFDnCFDTMTASTSyHL: <RSCFDn_base> + 0382_H + (04_H × y),
RSCFDnCFDTMTASTSyHH: <RSCFDn_base> + 0383_H + (04_H × y)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTASTSp (p = y × 32 + 31 to y × 32 × 16 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMTASTSp (p = y × 32 + 15 to y × 32 + 0 (y = 0, 1))															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.159 RSCFDnCFDTMTASTSy Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 31 to y × 32 + 16) 0: Transmission is not aborted 1: Transmission is aborted
15 to 0	TMTASTSp	Transmit Buffer Transmit Abort Status Flag p (p = y × 32 + 15 to y × 32 + 0) 0: Transmission is not aborted. 1: Transmission is aborted.

TMTASTSp Flags (p = 0 to 47)

When the TMTRF[1:0] flag in the RSCFDnCFDTMSTSp register is set to 01_B (transmit abort has been completed), the corresponding TMTASTSp flag is set to 1.

A TMTASTSp flag is cleared to 0 when the corresponding TMTRF[1:0] flag is set to 00_B or in channel reset mode.

Table 23.160 shows the bit assignment.

Table 23.160 TMTASTSp Bit Assignment

Bit	Channel	Transmit Buffer Number
0	0	0
1	0	1
. .	. .	. .
15	0	15
16	1	0
. .	. .	. .
30	1	14
31	1	15
32	2	0
33	2	1
. .	. .	. .
47	2	15

23.4.12 Details of Transmit Queue-related Registers

23.4.12.1 RSCFDnCFDTXQCCm — Transmit Queue Configuration and Control Register (m = 0 to 2)

Access: RSCFDnCFDTXQCCm register can be read/written in 32-bit units
RSCFDnCFDTXQCCmL register can be read/written in 16-bit units
RSCFDnCFDTXQCCmLL, RSCFDnCFDTXQCCmLH registers can be read/written in 8-bit units

Address: RSCFDnCFDTXQCCm: <RSCFDn_base> + $03A0_H$ + (04_H × m)

RSCFDnCFDTXQCCmL: <RSCFDn_base> + $03A0_H$ + (04_H × m)

RSCFDnCFDTXQCCmLL: <RSCFDn_base> + $03A0_H$ + (04_H × m),
RSCFDnCFDTXQCCmLH: <RSCFDn_base> + $03A1_H$ + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TXQIM	TXQIE	TXQDC[3:0]				—	—	—	—	—	—	—	TXQE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 23.161 RSCFDnCFDTXQCCm Register Contents

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
13	TXQIM	Transmit Queue Interrupt Source Select 0: When the transmit queue becomes empty upon completion of message transmission, a transmit queue interrupt request is generated. 1: A transmit queue interrupt request is generated each time a message has been transmitted.
12	TXQIE	Transmit Queue Interrupt Enable 0: Transmit queue interrupt is disabled. 1: Transmit queue interrupt is enabled.
11 to 8	TXQDC[3:0]	Transmit Queue Depth Configuration Setting these bits to g (g = 2 to 15) makes the (g + 1) transmit queue available. Setting these bits to 0 disables the transmit queue. Setting these bits to 1 is prohibited. For transmit buffer merge mode, set g to 2 to 9.
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	TXQE	Transmit Queue Enable 0: The transmit queue is not used. 1: The transmit queue is used.

TXQIM Bit

This bit is used to select a transmit queue interrupt source. Modify this bit in channel reset mode.

TXQIE Bit

When the TXQIE bit is set to 1 and the source selected by the TXQIM bit occurs, an interrupt request is generated.

Set the TXQE bit to 0 before modifying the TXQIE bit.

TXQDC[3:0] Bits

These bits are used to specify the number of transmit buffers to be allocated to the transmit queues.

Transmit buffers are allocated to transmit queues in descending order of buffer number, that is, from (m × 16 + 15) to (m × 16 + 0). (See **Table 23.92**.) For examples of how buffer allocation is done, see Figure 23.9.

When the TMME bit in the RSCFDnCFDCmFDCFG register is 1 (transmit buffer merge mode), transmit buffers (m × 16 + 5) to (m × 16 + 0) are merged and cannot be allocated to the transmit queue. Therefore, do not set TXQDC[3:0] bits to 10 to 15.

Modify these bits only in channel reset mode.

TXQE Bit

Setting this bit to 1 makes the transmit queue available. Modify this bit in channel communication mode or channel halt mode. This bit is cleared to 0 in channel reset mode.

Before setting the TXQE bit to 1, set the TXQDC[3:0] bits to 0010_B or more.

23.4.12.2 RSCFDnCFDTXQSTSm — Transmit Queue Status Register (m = 0 to 2)

Access: RSCFDnCFDTXQSTSm register can be read/written in 32-bit units
RSCFDnCFDTXQSTSmL register can be read/written in 16-bit units
RSCFDnCFDTXQSTSmLL register can be read/written in 8-bit units

Address: RSCFDnCFDTXQSTSm: <RSCFDn_base> + $03C0_H$ + (04_H × m)

RSCFDnCFDTXQSTSmL: <RSCFDn_base> + $03C0_H$ + (04_H × m)

RSCFDnCFDTXQSTSmLL: <RSCFDn_base> + $03C0_H$ + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQIF	TXQFLL	TXQEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.162 RSCFDnCFDTXQSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 8	Reserved	When read, the undefined value is returned. When writing to these bits, write the value after reset.
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	TXQIF	Transmit Queue Interrupt Request Flag 0: No transmit queue interrupt request is present. 1: A transmit queue interrupt request is present.
1	TXQFLL	Transmit Queue Full Status Flag 0: The transmit queue is not full. 1: The transmit queue is full.
0	TXQEMP	Transmit Queue Empty Status Flag 0: The transmit queue contains messages. 1: The transmit queue contains no message (transmit queue empty).

TXQIF Flag

The TXQIF flag is set to 1 when the event specified by the TXQIM bit in the RSCFDnCFDTXQCCm register has occurred.

The TXQIF flag is cleared to 0 in channel reset mode or by writing 0 to this flag. This flag is not cleared to 0 by setting the TXQE bit in the RSCFDnCFDTXQCCm register to 0 (the transmit queue is not used).

TXQFLL Flag

The TXQFLL flag is set to 1 when the number of messages set for the transmit queue matches the transmit queue depth set by the TXQDC[3:0] bits in the RSCFDnCFDTXQCCm register.

This flag is cleared to 0 in any of the following cases.

- The number of messages set for the transmit queue is smaller than the transmit queue depth set by the TXQDC[3:0] bits.
- In channel reset mode

TXQEMP Flag

The TXQEMP flag is cleared to 0 when even a single message is set for the transmit queue.

This flag is set to 1 in any of the following cases.

- The TXQE bit is set to 0 (the transmit queue is not used).
- The transmit queue becomes empty.
- In channel reset mode

23.4.12.3 RSCFDnCFDTXQPCTRm — Transmit Queue Pointer Control Register (m = 0 to 2)

Access: RSCFDnCFDTXQPCTRm registers can only be written in 32-bit units
RSCFDnCFDTXQPCTRmL register is a write-only register that can only be written in 16-bit units
RSCFDnCFDTXQPCTRmLL register is a write-only register that can only be written in 8-bit units

Address: RSCFDnCFDTXQPCTRm: <RSCFDn_base> + $03E0_H$ + (04_H × m)

RSCFDnCFDTXQPCTRmL: <RSCFDn_base> + $03E0_H$ + (04_H × m)

RSCFDnCFDTXQPCTRmLL: <RSCFDn_base> + $03E0_H$ + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.163 RSCFDnCFDTXQPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	The write value should be the value after reset.
7 to 0	TXQPC[7:0]	Transmit Queue Pointer Control Writing FF_H to these bits moves the write pointer of the transmit queue to the next queue buffer.

TXQPC[7:0] Bits

Writing FF_H to the TXQPC[7:0] bits moves the write pointer to the next transmit queue buffer and generates a transmit request of the message. Write transmit messages to the RSCFDnCFDTMIDp, RSCFDnCFDTMPTRp, RSCFDnCFDTMFDCTRp, and RSCFDnCFDTMDFb_p registers (p = 15, 31, and 47) before writing FF_H to the TXQPC[7:0] bits.

When writing FF_H to these bits, make sure that the TXQE bit in the RSCFDnCFDTXQCCm register is set to 1 (the transmit queue is used) and the TXQFLL flag in the RSCFDnCFDTXQSTSm register is 0 (he transmit queue is not full).

23.4.13 Details of Transmission History-related Registers

23.4.13.1 RSCFDnCFDTHLCCm — Transmit History Configuration and Control Register (m = 0 to 2)

Access: RSCFDnCFDTHLCCm register can be read/written in 32-bit units
RSCFDnCFDTHLCCmL register can be read/written in 16-bit units
RSCFDnCFDTHLCCmLL, RSCFDnCFDTHLCCmLH registers can be read/written in 8-bit units

Address: RSCFDnCFDTHLCCm: <RSCFDn_base> + 0400_H + (04_H × m)

RSCFDnCFDTHLCCmL: <RSCFDn_base> + 0400_H + (04_H × m)

RSCFDnCFDTHLCCmLL: <RSCFDn_base> + 0400_H + (04_H × m),
RSCFDnCFDTHLCCmLH: <RSCFDn_base> + 0401_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

Table 23.164 RSCFDnCFDTHLCCm Register Contents

Bit Position	Bit Name	Function
31 to 11	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
10	THLDTE	Transmit History Target Buffer Select 0: Entry from transmit/receive FIFO buffers and transmit queue 1: Entry from transmit buffers, transmit/receive FIFO buffers, and transmit queue
9	THLIM	Transmit History Interrupt Source Select 0: When 12 sets of data have been stored in the transmit history buffer 1: When a single set of transmit history data has been stored
8	THLIE	Transmit History Interrupt Enable 0: Transmit history interrupt is disabled. 1: Transmit history interrupt is enabled.
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	THLE	Transmit History Buffer Enable 0: Transmit history buffer is not used. 1: Transmit history buffer is used.

THLDTE Bit

When this bit is set to 0, the transmit history data of messages transmitted from transmit/receive FIFO buffers and the transmit queue is stored in the transmit history buffer. When this bit is set to 1, the transmit history data of messages transmitted from transmit buffers, transmit/receive FIFO buffers, and the transmit queue is stored in the transmit history buffer.

Modify this bit only in channel reset mode.

THLIM Bit

This bit is used to select a transmit history interrupt source.

Modify this bit only in channel reset mode.

THLIE Bit

When the THLIE bit is set to 1 and the source selected by the THLIM bit has occurred, a transmit history interrupt request is generated. Modify the THLIE bit only when the THLE bit set to 0.

THLE Bit

Setting this bit to 1 makes the transmit history buffer available. When data transmission from the buffer selected by the THLDTE bit has been completed, the transmit history data of transmit messages is stored in the transmit history buffer.

Modify this bit in channel communication mode or channel halt mode.

This bit is cleared to 0 in channel reset mode.

23.4.13.2 RSCFDnCFDTHLSTSm — Transmit History Status Register (m = 0 to 2)

Access: RSCFDnCFDTHLSTSm register can be read/written in 32-bit units
RSCFDnCFDTHLSTSmL register can be read/written in 16-bit units
RSCFDnCFDTHLSTSmLL register can be read/written in 8-bit units
RSCFDnCFDTHLSTSmLH register is a read-only register that can be read in 8-bit units

Address: RSCFDnCFDTHLSTSm: <RSCFDn_base> + 0420_H + (04_H × m)

RSCFDnCFDTHLSTSmL: <RSCFDn_base> + 0420_H + (04_H × m)

RSCFDnCFDTHLSTSmLL: <RSCFDn_base> + 0420_H + (04_H × m),
RSCFDnCFDTHLSTSmLH: <RSCFDn_base> + 0421_H + (04_H × m)

Value after reset: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	THLMC[4:0]					—	—	—	—	THLIF	THLELT	THLFLL	THLEMP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W*1	R/W*1	R	R

Note 1. The only effective value for writing to this flag bit is 0, which clears the bit. Otherwise writing to the bit results in retention of its state.

Table 23.165 RSCFDnCFDTHLSTSm Register Contents

Bit Position	Bit Name	Function
31 to 13	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 8	THLMC[4:0]	Transmit History Buffer Unread Data Counter These bits indicate the number of unread data sets stored in the transmit history buffer.
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3	THLIF	Transmit History Interrupt Request Flag 0: No transmit history interrupt request is present. 1: A transmit history interrupt request is present.
2	THLELT	Transmit History Buffer Overflow Flag 0: Transmit history buffer overflow has not occurred. 1: Transmit history buffer overflow has occurred.
1	THLFLL	Transmit history Buffer Full Status Flag 0: Transmit history buffer is not full. 1: Transmit history buffer is full.
0	THLEMP	Transmit History Buffer Empty Status Flag 0: Transmit history buffer contains unread data. 1: Transmit history buffer contains no unread data (buffer empty).

THLMC[4:0] Bits

These bits indicate the number of unread data sets stored in the transmit history buffer. These bits are cleared to 0 in channel reset mode.

THLIF Flag

The THLIF flag is set to 1 when the interrupt source specified with the THLIM bit in the RSCFDnCFDTHLCCm register occurs.

This flag is cleared to 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLELT Flag

The THLELT flag is set to 1 when an attempt is made to store new transmit history data while the transmit history buffer is full. In this case, the new data is discarded. This flag becomes 0 in channel reset mode or by the program writing 0 to this flag.

To clear the flags of the register to 0, the program must write 0 to the corresponding flag to be cleared.

When writing 0, using store instruction, set the bit to be set to “0” to “0”, and the bits not to be set to “0” to “1”.

THLFLL Flag

The THLFLL flag is set to 1 when 16 data sets have been stored in the transmit history buffer, and is cleared to 0 when the number of data sets stored in the transmit history buffer has decreased to less than 16. This bit is also cleared to 0 in channel reset mode or when the THLE bit in the RSCFDnCFDTHLCCm register is set to 0 (transmit history buffer is not used).

THLEMP Flag

The THLEMP flag is cleared to 0 when even a single set of transmit history data has been stored in the transmit history buffer.

This flag is set to 1 when all the data in the transmit history buffer has been read. This flag is also set to 1 in channel reset mode or when the THLE bit in the RSCFDnCFDTHLCCm register is set to 0 (transmit history buffer is not used).

NOTE

To clear THLIF or THLELT flag to 0, the program must write 0. When writing, use a store instruction to write “0” to the given flag and “1” to other flags.

23.4.13.3 RSCFDnCFDTHLPCTRm — Transmit History Pointer Control Register (m = 0 to 2)

Access: RSCFDnCFDTHLPCTRm register can only be written in 32-bit units
RSCFDnCFDTHLPCTRmL register is a write-only register that can only be written in 16-bit units
RSCFDnCFDTHLPCTRmLL register is a write-only register that can only be written in 8-bit units

Address: RSCFDnCFDTHLPCTRm: <RSCFDn_base> + 0440_H + (04_H × m)

RSCFDnCFDTHLPCTRmL: <RSCFDn_base> + 0440_H + (04_H × m)

RSCFDnCFDTHLPCTRmLL: <RSCFDn_base> + 0440_H + (04_H × m)

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	THLPC[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 23.166 RSCFDnCFDTHLPCTRm Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When writing to these bits, write the value after reset.
7 to 0	THLPC[7:0]	Transmit History List Pointer Control Writing FF_H to these bits moves the read pointer to the next unread data in the transmit history buffer.

THLPC[7:0] Bits

When the THLPC[7:0] bits are set to FF_H, the read pointer moves to the next data in the transmit history buffer. At this time, the THLMC[4:0] (transmit history buffer unread data counter) value in the RSCFDnCFDTHLSTSm register is decremented by 1. Write FF_H to the THLPC[7:0] bits after reading from the RSCFDnCFDTHLACCm register.

When writing FF_H to these bits, make sure that the THLE bit in the RSCFDnCFDTHLCCm register is set to 1 (transmit history buffer is used) and the THLEMP flag in the RSCFDnCFDTHLSTSm register is 0.

23.4.13.4 RSCFDnCFDTHLACCm — Transmit History Access Register (m = 0 to 2)

Access: RSCFDnCFDTHLACCm register can be read only in 32-bit units
RSCFDnCFDTHLACCmL, RSCFDnCFDTHLACCmH registers can be read only in 16-bit units
RSCFDnCFDTHLACCmLL, RSCFDnCFDTHLACCmLH, RSCFDnCFDTHLACCmHL,
RSCFDnCFDTHLACCmHH registers can be read only in 8-bit units

Address: RSCFDnCFDTHLACCm: <RSCFDn_base> + 6000_H + (04_H × m)

RSCFDnCFDTHLACCmL: <RSCFDn_base> + 6000_H + (04_H × m),
RSCFDnCFDTHLACCmH: <RSCFDn_base> + 6002_H + (04_H × m)

RSCFDnCFDTHLACCmLL: <RSCFDn_base> + 6000_H + (04_H × m),
RSCFDnCFDTHLACCmLH: <RSCFDn_base> + 6001_H + (04_H × m),
RSCFDnCFDTHLACCmHL: <RSCFDn_base> + 6002_H + (04_H × m),
RSCFDnCFDTHLACCmHH: <RSCFDn_base> + 6003_H + (04_H × m)

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMTS[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TID[7:0]								—	BN[3:0]				BT[2:0]		
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 23.167 RSCFDnCFDTHLACCm Register Contents

Bit Position	Bit Name	Function
31 to 16	TMTS[15:0]	Timestamp Data The timestamp data of stored data can be read.
15 to 8	TID[7:0]	Label Data The label information of stored data can be read.
7	Reserved	When read, the value after reset is returned.
6 to 3	BN[3:0]	Buffer Number Data The buffer number of transmit source (transmit buffer, transmit/receive FIFO or transmit queue) can be read.
2 to 0	BT[2:0]	Buffer Type Data b2 b1 b0 0 0 1: Transmit buffer 0 1 0: Transmit/receive FIFO buffer 1 0 0: Transmit queue

TMTS[15:0] Bits

Timestamp values in transmit history data stored in the transmit history buffer are displayed.

TID[7:0] Bits

These bits indicate the label information of transmit history data stored in the transmit history buffer.

BN[3:0] Bits

These bits indicate the transmit source buffer number in the transmit history data stored in the transmit history buffer.

BT[2:0] Bits

These bits indicate the type of the transmit source buffer in the transmit history data stored in the transmit history buffer.

23.4.14 Details of Test-related Registers

23.4.14.1 RSCFDnCFDGTSTCFG — Global Test Configuration Register

Access: RSCFDnCFDGTSTCFG register can be read/written in 32-bit units
RSCFDnCFDGTSTCFGL, RSCFDnCFDGTSTCFGH registers can be read/written in 16-bit units
RSCFDnCFDGTSTCFGLL, RSCFDnCFDGTSTCFGHL registers can be read/written in 8-bit units

Address: RSCFDnCFDGTSTCFG: <RSCFDn_base> + 0468_H

RSCFDnCFDGTSTCFGL: <RSCFDn_base> + 0468_H,
RSCFDnCFDGTSTCFGH: <RSCFDn_base> + $046A_H$

RSCFDnCFDGTSTCFGLL: <RSCFDn_base> + 0468_H,
RSCFDnCFDGTSTCFGHL: <RSCFDn_base> + $046A_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	RTMPS[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	C2ICBCE	C1ICBCE	C0ICBCE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 23.168 RSCFDnCFDGTSTCFG Register Contents

Bit Position	Bit Name	Function
31 to 23	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
22 to 16	RTMPS[6:0]	RAM Test Page Configuration Set a value within a range of page 0 (00_H) to page 41 (29_H).
15 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	C2ICBCE	CAN2 Inter-channel Communication Test Enable 0: CAN2 inter-channel communication test is disabled 1: CAN2 inter-channel communication test is enabled.
1	C1ICBCE	CAN1 Inter-Channel Communication Test Enable 0: CAN1 inter-channel communication test is disabled. 1: CAN1 inter-channel communication test is enabled.
0	C0ICBCE	CAN0 Inter-Channel Communication Test Enable 0: CAN0 inter-channel communication test is disabled. 1: CAN0 inter-channel communication test is enabled.

Modify the RSCFDnCFDGTSTCFG register only in global test mode.

RTMPS[6:0] Bits

These bits are used to set the RAM test target page number for RAM test. Set a value in the range of 00_H to 29_H, inclusive.

In CAN FD mode, do not access more than 160 bytes in the last page (RTMPS[6:0] = 29_H) during RAM test.

C2ICBCE Bit

Setting this bit to 1 enables the channel 2 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C1ICBCE Bit

Setting this bit to 1 enables the channel 1 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

C0ICBCE Bit

Setting this bit to 1 enables the channel 0 inter-channel communication test.

This bit is cleared to 0 in global reset mode.

23.4.14.2 RSCFDnCFDGTSTCTR — Global Test Control Register

Access: RSCFDnCFDGTSTCTR register can be read/written in 32-bit units
RSCFDnCFDGTSTCTRL register can be read/written in 16-bit units
RSCFDnCFDGTSTCTRLL register can be read/written in 8-bit units

Address: RSCFDnCFDGTSTCTR: <RSCFDn_base> + $046C_H$

RSCFDnCFDGTSTCTRL: <RSCFDn_base> + $046C_H$

RSCFDnCFDGTSTCTRLL: <RSCFDn_base> + $046C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	ICBCTME
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W

Table 23.169 RSCFDnCFDGTSTCTR Register Contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2	RTME	RAM Test Enable 0: RAM test is disabled. 1: RAM test is enabled.
1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	ICBCTME	Communication Test between Channels Enable 0: Communication test between channels disabled 1: Communication test between channels enabled

RTME Bit

Setting this bit to 1 enables the RAM test. Modify this bit only in global test mode.

This bit is cleared to 0 in global reset mode (See Figure 23.37, RAM Test Setting Procedure).

1. Set the GMDC[1:0] bits in the RSCFDnCFDGCTR register to 10B (Global test mode).
2. Set the RTME bit to 1.
3. Check that the RTME bit is set to 1.

ICBCTME Bit

When this bit is set to 1, a communication test is enabled between the channels for which the CmICBCE bit (m = 0 to 2) in the RSCFDnCFDGTSTCFG register has been set to 1. Modify the ICBCTME bit only in global test mode.

This bit is cleread to 0 in global reset mode.

23.4.14.3 RSCFDnCFDGLOCKK — Global Lock Key Register

Access: RSCFDnCFDGLOCKK register is a write-only register that can be write only in 32-bit units.
RSCFDnCFDGLOCKKL register is a write-only register that can be write only in 16-bit units.

Address: RSCFDnCFDGLOCKK: <RSCFDn_base> + $047C_H$

RSCFDnCFDGLOCKKL: <RSCFDn_base> + $047C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LOCK[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1	W*1

Note 1. Writing to these bits is effective only when the RS-CANFD module is in global test mode.

Table 23.170 RSCFDnCFDGLOCKK Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When writing these bits, write the value after reset.
15 to 0	LOCK[15:0]	Lock Key These bits are key bits to release protection of test mode.

The RSCFDnCFDGLOCKK register releases protection of special test bits and is write only.

For the protection release data, see Section 23.11.4.2, Procedure for Releasing the Protection.

LOCK[15:0] Bits

Writing the protection release data to the LOCK[15:0] bits in succession enables writing 1 to the RTME bit in the RSCFDnCFDGTSTCTR register.

After the protection has been released, writing to the I/O register area (<RSCFDn_base> + 0000_H to <RSCFDn_base> + $05FF_H$) of the CAN (except the RAM) enables the protection again.

Reading from the I/O register area of the CAN or reading from/writing to other areas does not enable the protection.

23.4.14.4 RSCFDnCFDRPGACCr — RAM Test Page Access Register (r = 0 to 63)

Access: RSCFDnCFDRPGACCr register can be read/written in 32-bit units
RSCFDnCFDRPGACCrL, RSCFDnCFDRPGACCrH registers can be read/written in 16-bit units
RSCFDnCFDRPGACCrLL, RSCFDnCFDRPGACCrLH, RSCFDnCFDRPGACCrHL,
RSCFDnCFDRPGACCrHH registers can be read/written in 8-bit units

Address: RSCFDnCFDRPGACCr: <RSCFDn_base> + 6400_H + (04_H × r)

RSCFDnCFDRPGACCrL: <RSCFDn_base> + 6400_H + (04_H × r)),
RSCFDnCFDRPGACCrH: <RSCFDn_base> + 6402_H + (04_H × r))

RSCFDnCFDRPGACCrLL: <RSCFDn_base> + 6400_H + (04_H × r)),
RSCFDnCFDRPGACCrLH: <RSCFDn_base> + 6401_H + (04_H × r)),
RSCFDnCFDRPGACCrHL: <RSCFDn_base> + 6402_H + (04_H × r)),
RSCFDnCFDRPGACCrHH: <RSCFDn_base> + 6403_H + (04_H × r))

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDTA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDTA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.171 RSCFDnCFDRPGACCr Register Contents

Bit Position	Bit Name	Function
31 to 0	RDTA [31:0]	RAM Data Test Access Data can be read and written in RSCAN RAM.

Modify the RSCFDnCFDRPGACCr register in global test mode with the RTME bit in the RSCFDnCFDGTSTCTR register set to 1 (RAM test is enabled).

The RSCFDnCFDRPGACCr register is readable and writable when the RTME bit is set to 1.

23.5 Interrupt Sources

The RS-CANFD module has 11 interrupts that are grouped into global interrupts and channel interrupts.

- Global interrupts (2 sources):
  Receive FIFO interrupt
  Global error interrupt
- Channel interrupts (3 sources/channel):
  CANm transmit interrupt (m = 0 to 2)
  - CANm transmit complete interrupt
  - CANm transmit abort interrupt
  - CANm transmit/receive FIFO transmit complete interrupt (in transmit mode, gateway mode)
  - CANm transmit history interrupt
  - CANm transmit queue Interrupt

  CANm transmit/receive FIFO receive complete interrupt (in receive mode, gateway mode)
  CANm error interrupt

When an interrupt request is generated, the corresponding interrupt request flag is set to 1 (interrupt request present). In that case, when the interrupt enable bit is set to 1 (enabling interrupts), an interrupt request is output from the RS-CANFD module. (Generation of interrupts also depends on the interrupt control register settings of the interrupt controller.)

Setting the interrupt request flag to 0 (no interrupt request present) or setting the interrupt enable bit to 0 (disabling interrupts) clears the current interrupt request. The current interrupt request is still output until the interrupt request flag is cleared.

Table 23.172 lists the CAN interrupt sources. **Figure 23.2** shows the CAN global interrupt block diagram.

Table 23.172 List of CAN Interrupt Sources (1/2)

	Interrupt Source		Corresponding Interrupt Request Flag	Corresponding Interrupt Enable Bit
Global interrupts	Receive FIFO	Receive FIFO 0	RFIF in the RSCFDn(CFD)RFSTS0 register	RFIE in the RSCFDn(CFD)RFCC0 register
		Receive FIFO 1	RFIF in the RSCFDn(CFD)RFSTS1 register	RFIE in the RSCFDn(CFD)RFCC1 register
		Receive FIFO 2	RFIF in the RSCFDn(CFD)RFSTS2 register	RFIE in the RSCFDn(CFD)RFCC2 register
		Receive FIFO 3	RFIF in the RSCFDn(CFD)RFSTS3 register	RFIE in the RSCFDn(CFD)RFCC3 register
		Receive FIFO 4	RFIF in the RSCFDn(CFD)RFSTS4 register	RFIE in the RSCFDn(CFD)RFCC4 register
		Receive FIFO 5	RFIF in the RSCFDn(CFD)RFSTS5 register	RFIE in the RSCFDn(CFD)RFCC5 register
		Receive FIFO 6	RFIF in the RSCFDn(CFD)RFSTS6 register	RFIE in the RSCFDn(CFD)RFCC6 register
		Receive FIFO 7	RFIF in the RSCFDn(CFD)RFSTS7 register	RFIE in the RSCFDn(CFD)RFCC7 register
	Global error		• DEF in the RSCFDn(CFD)GERFL register • MES in the RSCFDn(CFD)GERFL register • THLES in the RSCFDn(CFD)GERFL register • CMPOF in the RSCFDnCFDGERFL register	• DEIE in the RSCFDn(CFD)GCTR register • MEIE in the RSCFDn(CFD)GCTR register • THLEIE in the RSCFDn(CFD)GCTR register • CMPOFIE in the RSCFDnCFDGERFL register

Table 23.172 List of CAN Interrupt Sources (2/2)

	Interrupt Source		Corresponding Interrupt Request Flag	Corresponding Interrupt Enable Bit
Channel interrupts (m = 0 to 2)	CANm transmit	CANm transmit complete	TMTRF[1:0] in the RSCFDn(CFD)TMSTSp register	TMIEp in the RSCFDn(CFD)TMIECy register
		CANm transmit abort	TMTRF[1:0] in the RSCFDn(CFD)TMSTSp register	TAIE in the RSCFDn(CFD)CmCTR register
		CANm transmit/receive FIFO transmit complete	CFTXIF in the RSCFDn(CFD)CFSTSk register	CFTXIE in the RSCFDn(CFD)CFCCk register
		CANm transmit queue	TXQIF in the RSCFDn(CFD)TXQSTSm register	TXQIE in the RSCFDn(CFD)TXQCCm register
		CANm transmit history	THLIF in the RSCFDn(CFD)THLSTSm register	THLIE in the RSCFDn(CFD)THLCCm register
	CANm transmit/receive FIFO receive complete		CFRXIF in the RSCFDn(CFD)CFSTSk register	CFRXIE in the RSCFDn(CFD)CFCCk register
	CANm error		• BEF in the RSCFDn(CFD)CmERFL register • ALF in the RSCFDn(CFD)CmERFL register • BLF in the RSCFDn(CFD)CmERFL register • OVLF in the RSCFDn(CFD)CmERFL register • BORF in the RSCFDn(CFD)CmERFL register • BOEF in the RSCFDn(CFD)CmERFL register • EPF in the RSCFDn(CFD)CmERFL register • EWF in the RSCFDn(CFD)CmERFL register • SOCO in the RSCFDnCFDCmFDSTS register • EOCO in the RSCFDnCFDCmFDSTS register • TDCVF in the RSCFDnCFDCmFDSTS register	• BEIE in the RSCFDn(CFD)CmCTR register • ALIE in the RSCFDn(CFD)CmCTR register • BLIE in the RSCFDn(CFD)CmCTR register • OLIE in the RSCFDn(CFD)CmCTR register • BORIE in the RSCFDn(CFD)CmCTR register • BOEIE in the RSCFDn(CFD)CmCTR register • EPIE in the RSCFDn(CFD)CmCTR register • EWIE in the RSCFDn(CFD)CmCTR register • SOCOIE in the RSCFDnCFDCmCTR register • EOCOIE in the RSCFDnCFDCmCTR register • TDCVFIE in the RSCFDnCFDCmCTR register

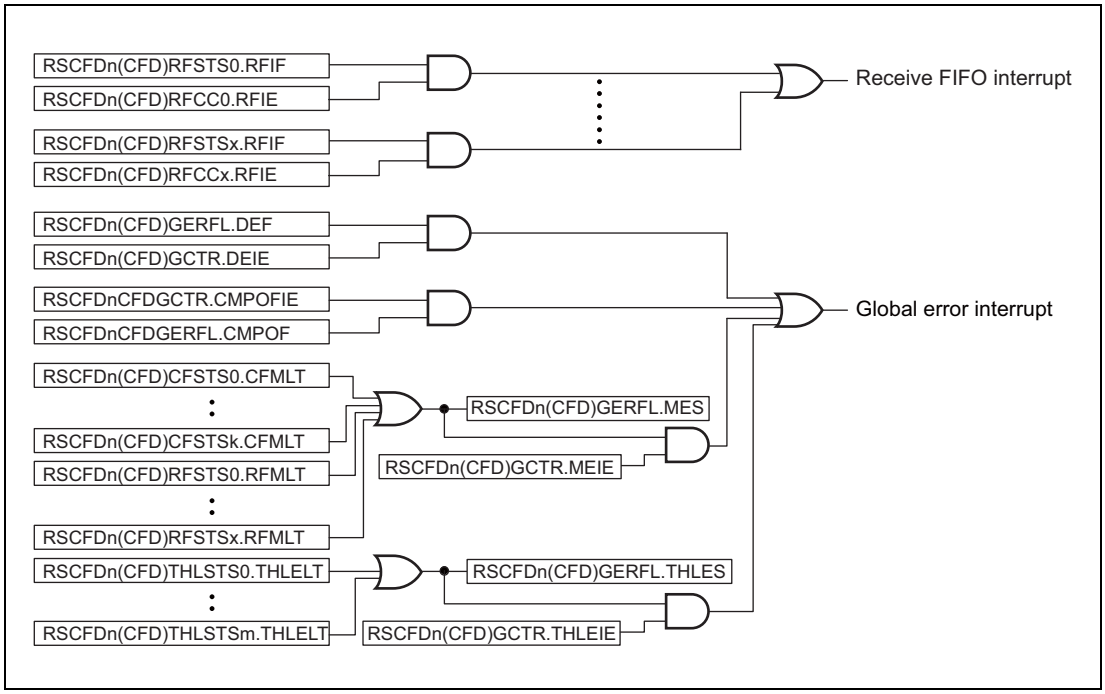


Figure 23.2 CAN Global Interrupt Block Diagram

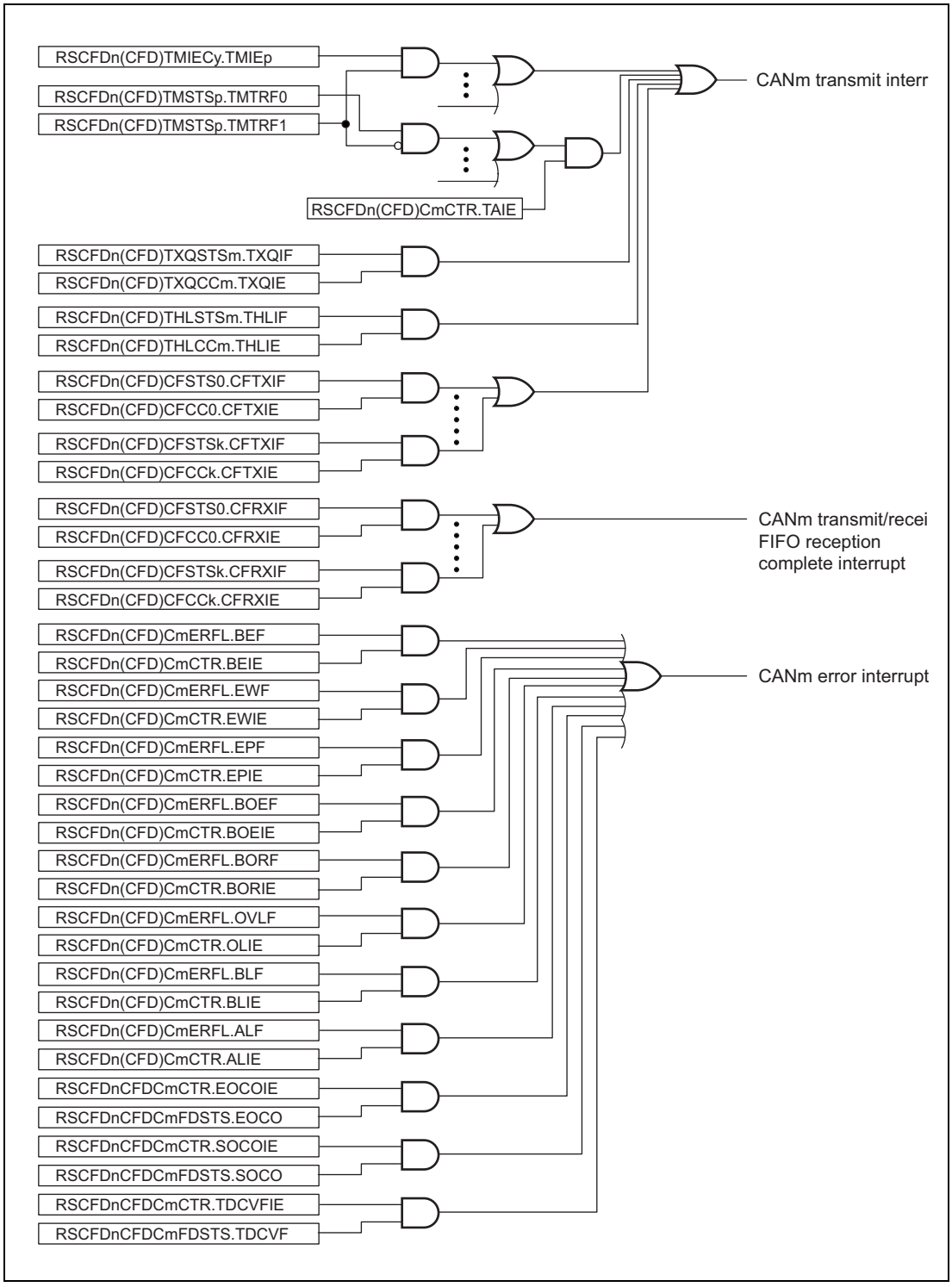


Figure 23.3 CAN Channel Interrupt Block Diagram

23.6 CAN Modes

The RS-CANFD module has four global modes to control the entire RS-CANFD module status and four channel modes to control individual channel status. Details of global modes are described in Section 23.6.1, Global Modes, and details of channel modes are described in **Section 23.6.2, Channel Modes**.

- Global stop mode: Stops the clocks of the entire module to achieve low power consumption.
- Global reset mode: Performs initial settings for the entire module.
- Global test mode: Performs test settings and performs the RAM test.
- Global operating mode: Makes the entire module operable.
- Channel stop mode: Stops the channel clock.
- Channel reset mode: Performs initial settings for the channels.
- Channel halt mode: Stops CAN communication and allows channel testing.
- Channel communication mode: Performs CAN communication.

23.6.1 Global Modes

Figure 23.4 shows the transitions of global modes.

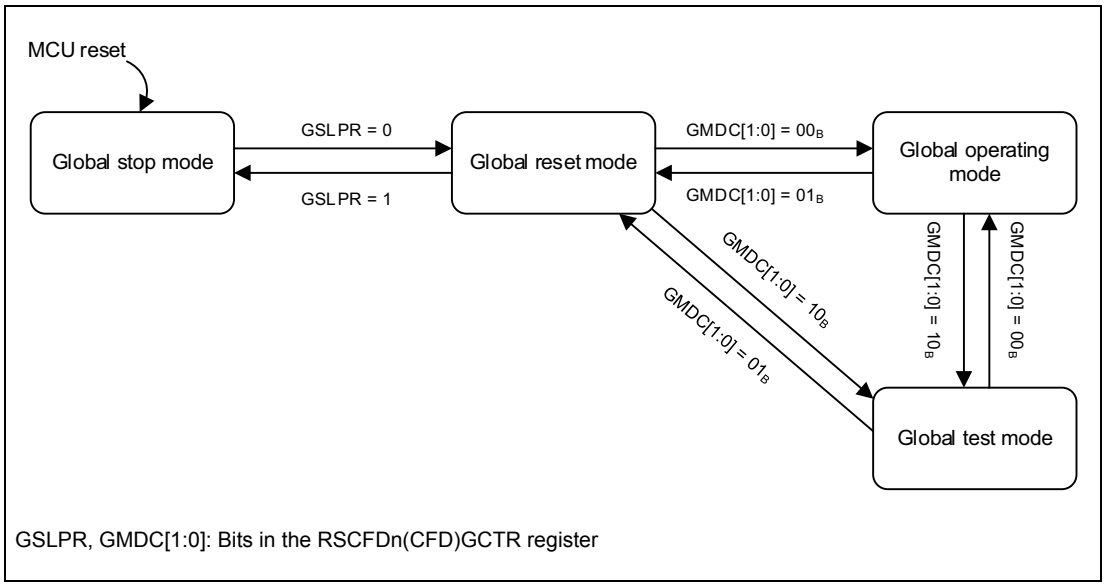


Figure 23.4 Transitions of Global Modes

In some cases, global mode transitions also force channel mode transitions. **Table 23.173** shows the channel mode transitions depending on the global mode setting dictated by the GMDC[1:0] bits and the GSLPR bit.

Table 23.173 Transitions of Channel Modes Depending on Global Mode Setting (GMDC[1:0] and GSLPR Bits)

Channel Mode before Setting	Channel Mode after Setting			
	GMDC[1:0] = 00_B GSLPR = 0 (Global Operation)	GMDC[1:0] = 10_B GSLPR = 0 (Global Test)	GMDC[1:0] = 01_B GSLPR = 0 (Global Reset)	GMDC[1:0] = 01_B GSLPR = 1 (Global Stop)
Channel communication	Channel communication	Channel halt	Channel reset	Transition prohibited
Channel halt	Channel halt	Channel halt	Channel reset	Transition prohibited
Channel reset	Channel reset	Channel reset	Channel reset	Channel stop
Channel stop	Channel stop	Channel stop	Channel stop	Channel stop

Note: GMDC[1:0], GSLPR: Bits in the RSCFDn(CFD)GCTR register

Table 23.174 shows the global mode transition time.

Table 23.174 Global Mode Transition Time

Mode before Transition	Mode after Transition	Maximum Transition Time
Global stop	Global reset	Three pclk cycles
Global reset	Global stop	Three pclk cycles
Global reset	Global test	Ten pclk cycles
Global reset	Global operating	Ten pclk cycles
Global test	Global reset	Two CAN bit times*1,*2
Global test	Global operating	Three pclk cycles
Global operating	Global reset	Two CAN bit times*1,*2
Global operating	Global test	Two CAN frames*1

Note 1. CAN frame time and CAN bit time of the lowest communication speed of the channels in use

Note 2. In CAN FD mode, this time value is the CAN bit time of the nominal bit rate.

23.6.1.1 Global Stop Mode

In global stop mode, clocks of the CAN do not run and therefore power consumption is reduced. CAN registers can be read, but writing data to them is prohibited. Register values are retained. Only the clock used by the CPU for writing to the GSLPR bit runs in this mode.

After the MCU is reset, the CAN module transitions to global stop mode. Setting the GSLPR bit in the RSCFDn(CFD)GCTR register to 1 (in global stop mode) in global reset mode sets the CSLPR bit in each of the RSCFDn(CFD)CmCTR register to 1 (channel stop mode). Afterwards, if all channels are forced to transition to channel stop mode, the CAN module transitions to global stop mode. The GSLPR bit should not be modified in global operating mode or global test mode.

23.6.1.2 Global Reset Mode

In global reset mode, RS-CANFD module settings are performed. When the RS-CANFD module transitions to global reset mode, some registers are initialized. For registers to be initialized, see Table 23.177, Registers Initialized in Global Reset Mode or Channel Reset Mode and Table 23.178, Registers Initialized Only in Global Reset Mode.

Setting the GMDC[1:0] bits in the RSCFDn(CFD)GCTR register to 01_B sets the CHMDC[1:0] bits in each of the RSCFDn(CFD)CmCTR registers (m = 0 to 2) to 01_B (channel reset mode). If all channels are forced to transition to channel reset mode, the CAN module transitions to global reset mode.

Channels that are already in channel reset mode or channel stop mode do not transition (because the CHMDC[1:0] bits have already been set to 01_B).

23.6.1.3 Global Test Mode

In global test mode, settings for test-related registers are performed. When the CAN module transitions to global test mode, all CAN communications are disabled.

Setting the GMDC[1:0] bits in the RSCFDn(CFD)GCTR register to 10_B sets the CHMDC[1:0] bits in each of the RSCFDn(CFD)CmCTR register to 10_B (channel halt mode). If all channels are forced to transition to channel halt mode, the CAN module transitions to global test mode. Channels that are in channel stop mode, channel reset mode, or channel halt mode do not transition.

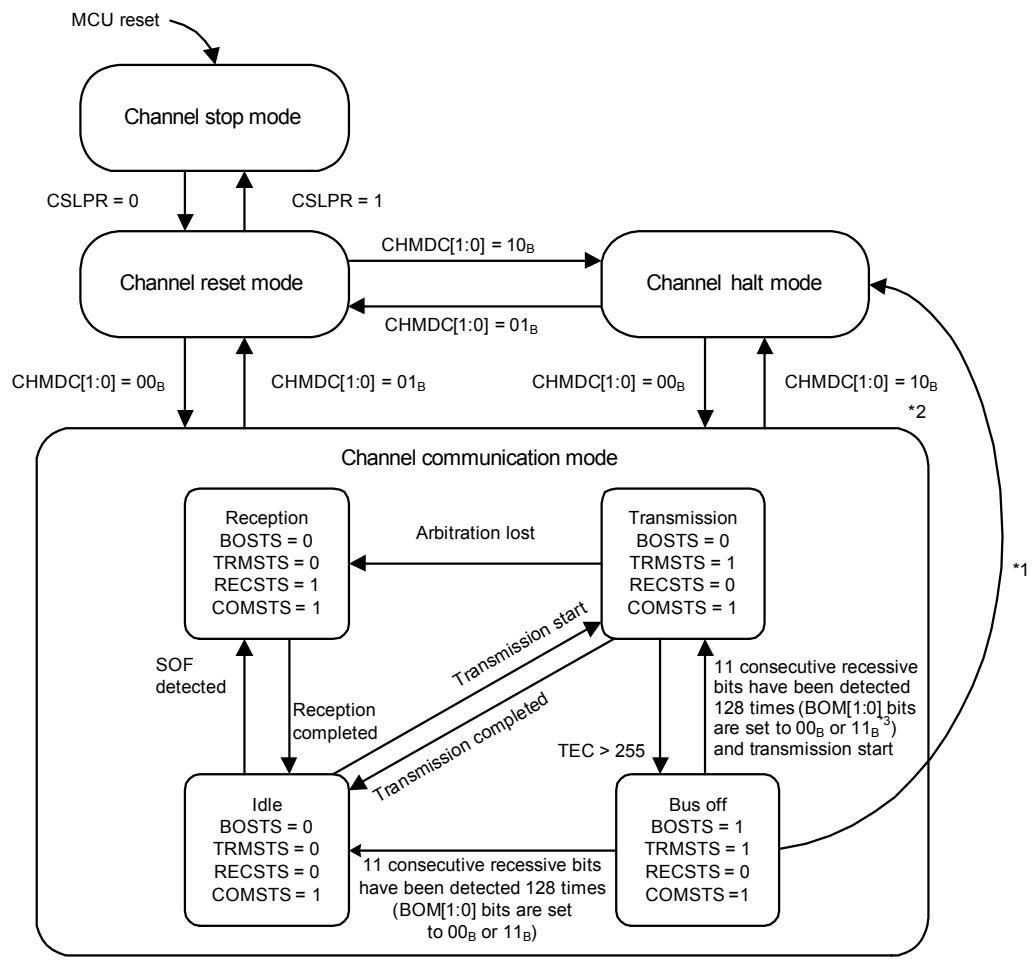
23.6.1.4 Global Operating Mode

The RS-CANFD module operates in global operating mode.

When the GMDC[1:0] bits in the RSCFDn(CFD)GCTR register are set to 00_B, the RS-CANFD module transitions to global operating mode.

23.6.2 Channel Modes

Figure 23.5 shows a channel mode state transition chart. **Table 23.175** shows the channel mode transition time.



Note: CHMDC[1:0], CSLPR, BOM[1:0]: Bits in the RSCFDn(CFD)CmCTR register (m = 0 to 2)
BOSTS, TRMSTS, RECSTS, COMSTS: Bits in the RSCFDn(CFD)CmSTS register

Note 1. Timing of transition from bus off state to channel halt mode
- When BOM[1:0] = 01_B: Transition to channel halt mode when TEC exceeds 255
- When BOM[1:0] = 10_B: Transition to channel halt mode when 11 consecutive recessive bits have been detected 128 times
- When BOM[1:0] = 11_B: Transition to channel halt mode when the CHMDC[1:0] bits are set to 10_B

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode.

Note 3. When 11 consecutive recessive bits are detected 128 times before the CHMDC[1:0] bits are set to 10_B.

Figure 23.5 Channel Mode State Transition Chart

Table 23.175 Channel Mode Transition Time (1/2)

Mode before Transition	Mode after Transition	Maximum Transition Time
Channel stop	Channel reset	Three pclk cycles
Channel reset	Channel stop	Three pclk cycles
Channel reset	Channel halt	Three CANm bit times*1
Channel reset	Channel communication	Four CANm bit times*1
Channel halt	Channel reset	Two CANm bit times*1

Table 23.175 Channel Mode Transition Time (2/2)

Mode before Transition	Mode after Transition	Maximum Transition Time
Channel halt	Channel communication	Four CANm bit times*1
Channel communication	Channel reset	Two CANm bit times*1
Channel communication	Channel halt	Two CANm frames

Note 1. In CAN FD mode, this time value is the CANm bit time of the nominal bit rate.

23.6.2.1 Channel Stop Mode

In channel stop mode, clocks are not supplied to channels and therefore power consumption is reduced. Channel-related registers can be read, but writing data to them is prohibited. Register values are retained.

Each channel enters channel stop mode after the MCU is reset. Channels also transition to channel stop mode when the CSLPR bit in the RSCFDn(CFD)CmCTR register (m = 0 to 2) is set to 1 (channel stop mode) in channel reset mode. The CSLPR bit should not be modified in channel communication mode and channel halt mode.

23.6.2.2 Channel Reset Mode

In channel reset mode, channel settings are performed. When a channel transitions to channel reset mode, some channel-related registers are initialized. For registers to be initialized, see **Table 23.177, Registers Initialized in Global Reset Mode or Channel Reset Mode**.

When the CHMDC[1:0] bits in the RSCFDn(CFD)CmCTR register are set to 01_B (channel reset mode) during CAN communication, communication is terminated before it is completed and the channel transitions to channel reset mode. **Table 23.176** shows the operation when the CHMDC[1:0] bits are set to 01_B (channel reset mode) during CAN communication.

23.6.2.3 Channel Halt Mode

In channel halt mode, settings for test-related registers of channels are performed. When a channel transitions to channel halt mode, CAN communication of the channel stops.

Table 23.176 shows operation when the CHMDC[1:0] bits are set to 10_B (channel halt mode) during CAN communication.

Table 23.176 Operation a Channel Transitions to Channel Reset Mode/Channel Halt Mode

Mode	During Reception	During Transmission	Bus Off State
Channel reset (CHMDC[1:0] = 01_B)	Transitions to channel reset mode before reception is completed.*1	Transitions to channel reset mode before transmission is completed.*1	Transitions to channel reset mode before bus off recovery.
Channel halt*3 (CHMDC[1:0] = 10_B)	Transitions to channel halt mode after reception is completed.*2	Transitions to channel halt mode after transmission is completed.	[When BOM[1:0] = 00_B] Transitions to channel halt mode (CHMDC[1:0] = 10_B) only after bus off recovery. [When BOM[1:0] = 01_B] Transitions to channel halt mode automatically when the condition for transition to bus off state is met. [When BOM[1:0] = 10_B] Transitions to channel halt mode automatically after bus off recovery. [When BOM[1:0] = 11_B] Transitions to channel halt mode immediately after the CHMDC[1:0] bits are set to 10_B before bus off recovery.

Note 1. To allow transition to channel reset mode after communication is completed, set the CHMDC[1:0] bits to 10_B and confirm that communication has been completed and transition to channel halt mode has been made, and then set the CHMDC[1:0] bits to 01_B.

Note 2. While the CAN bus is locked at the dominant level, transition to channel halt mode is not made. In that case, enter channel reset mode. The CAN bus status can be confirmed with the BLF flag of the RSCFDn(CFD)CmERFL register that becomes 1 when dominant lock is detected.

Note 3. In classical CAN mode, when the transition from channel reset mode to channel halt mode is to be made, set the RSCANnCmCFG register in channel reset mode and then shift to channel halt mode. In CAN FD mode, set the RSCFDnCFDCmNCFG register and the RSCFDnCFDCmDCFG register, and then make a transition.

23.6.2.4 Channel Communication Mode

In channel communication mode, CAN communication is performed. Each channel has the following communication states during CAN communication.

- Idle: Neither reception nor transmission is in progress.
- Reception:Receiving a message sent from another node.
- Transmission:Transmitting a message.
- Bus off:Isolated from CAN communication.

When the CHMDC[1:0] bits in the RSCFDn(CFD)CmCTR register are set to 00_B, the channel transitions to channel communication mode. After that, once 11 consecutive recessive bits have been detected, the COMSTS flag in the RSCFDn(CFD)CmSTS register (m = 0 to 2) is set to 1 (communication is ready) and transmission and reception are enabled on the CAN network as an active node. At this time, transmission and reception of messages can be started.

23.6.2.5 Bus Off State

A channel transitions to the bus off state according to the transmit/receive error counter increment/decrement rules of the CAN specifications.

The conditions for returning from the bus off state are determined by the BOM[1:0] bits in the RSCFDn(CFD)CmCTR register.

- When BOM[1:0] = 00_B:
  Bus off recovery is compliant with the CAN specifications. After 11 consecutive recessive bits have been detected 128 times, a channel returns from the bus off state to the CAN communication ready state (error active state). At that time, the TEC[7:0] and REC[7:0] bits in the RSCFDn(CFD)CmSTS register are initialized to 00_H, the BORF flag in the RSCFDn(CFD)CmERFL register is set to 1 (bus off recovery is detected), and a bus off recovery interrupt request is generated. When the CHMDC[1:0] bits in the RSCFDn(CFD)CmCTR register are set to 10_B (channel halt mode) in the bus off state, the channel transitions to channel halt mode after bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times).
- When BOM[1:0] = 01_B:
  When a channel transitions to the bus off state, the CHMDC[1:0] bits are set to 10_B and the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H. The BORF flag is not set to 1, and bus off recovery interrupt request is not generated.
- When BOM[1:0] = 10_B:
  When a channel has transitioned to the bus off state, the CHMDC[1:0] bits are set to 10_B. After bus off recovery has been completed (11 consecutive recessive bits have been detected 128 times),

the channel transitions to channel halt mode. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H, the BORF flag is set to 1, and a bus off recovery interrupt request is generated.

- When BOM[1:0] = 11_B:
  When the CHMDC[1:0] bits are set to 10_B in the bus off state, the channel transitions to channel halt mode before bus off recovery is completed. At that time, the TEC[7:0] and REC[7:0] bits are initialized to 00_H, but the BORF flag is not set to 1. Also, a bus off recovery interrupt is not generated.
  However, the BORF flag becomes 1 and a bus off recovery interrupt request is generated if a CAN module transitions to error active state (by detecting 128 times of 11 consecutive recessive bits) before CHMDC[1:0] bits are set to 10_B.

If the RS-CANFD module causes the channel to transition to channel halt mode simultaneously with a program write to the CHMDC[1:0] bits, the program write takes precedence. An automatic transition to channel halt mode when the BOM[1:0] bits are set to 01_B or 10_B is made only when the CHMDC[1:0] bits are 00_B (channel communication mode).

Furthermore, setting the RTBO bit in the RSCFDn(CFD)CmCTR register to 1 allows a forced return from the bus off state. As soon as the RTBO bit is set to 1, the state changes to the error active state. After 11 consecutive recessive bits have been detected, the CAN module becomes ready for communication. In this case, the BORF flag is not set to 1 and the TEC[7:0] and REC[7:0] bits are initialized to 00_H. Write 1 to the RTBO bit only when the BOM[1:0] value is 00_B. Writing the RTBO bit to 1 in a state other than the bus off state is ignored, and the RTBO bit is immediately set to 0.

23.6.3 Initializing Registers by Transition to CAN Mode

Table 23.177 lists bits and flags to be initialized by a transition to channel reset mode. These bits and flags are also initialized by a transition to global reset mode. Furthermore, **Table 23.178** lists bits and flags to be initialized only by a transition to global reset mode.

Table 23.177 Registers Initialized in Global Reset Mode or Channel Reset Mode (1/2)

Register	Bit / Flag
RSCFDn(CFD)CmCTR register	(ROM), CRCT, CTMS[1:0], CTME, CHMDC[1:0]
RSCFDn(CFD)CmSTS register	CHLTSTS, EPSTS, BOSTS, TRMSTS, RECSTS, COMSTS, (ESIF), REC[7:0], TEC[7:0]
RSCFDn(CFD)CmERFL register	CRCREG[14:0], ADERR, B0ERR, B1ERR, CERR, AERR, FERR, SERR, ALF, BLF, OVLF, BORF, BOEF, EPF, EWF, BEF
RSCFDnCFDCmFDCTR register	EOCCLR, SOCCLR
RSCFDnCFDCmFDSTS register	SOC[7:0], EOC[7:0], SOCO, EOCO, TDCVF, TDCR[6:0]
RSCFDnCFDCmFDCRC register	CRCREG[20:0], SCNT[3:0]
RSCFDn(CFD)CFCCk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFE
RSCFDn(CFD)CFSTSk register	When transmit/receive FIFO buffer is in transmit mode or gateway mode: CFMC[7:0], CFFLL, CFEMP, CFMLT, CFRXIF, CFTXIF
RSCFDn(CFD)CFTISTS register	CFkTXIF
RSCFDn(CFD)TMCp register	TMOM, TMTAR, TMTR
RSCFDn(CFD)TMSTSp register	TMTARM, TMTRM, TMTRF[1:0], TMTSTS
RSCFDn(CFD)TMTRSTSy register	TMTRSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCFDn(CFD)TMTARSTSy register	TMTARSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCFDn(CFD)TMTCSTSy register	TMTCSTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCFDn(CFD)TMTASTSy register	TMTASTSp (Bits of corresponding channel are initialized in channel reset mode.)
RSCFDn(CFD)TXQCCm register	TXQE

Table 23.177 Registers Initialized in Global Reset Mode or Channel Reset Mode (2/2)

Register	Bit / Flag
RSCFDn(CFD)TXQSTSm register	TXQIF, TXQFLL, TXQEMP
RSCFDn(CFD)THLCCm register	THLE
RSCFDn(CFD)THLSTSm register	THLMC[4:0], THLIF, THLELT, THLFLL, THLEMP
RSCFDn(CFD)GTINTSTS0 register	TSIFm, TAIFm, TQIFm, CFTIFm, THIFm (m = 0 to 2)

Note: Bits and flags in parentheses exist only in registers in CAN FD mode.

Table 23.178 Registers Initialized Only in Global Reset Mode

Register	Bit / Flag
RSCFDn(CFD)GSTS register	GHLTSTS
RSCFDn(CFD)GERFL register	EEF0, EEF1, EEF2, EEF3, EEF4, EEF5, (CMPOF), THLES, MES, DEF
RSCFDn(CFD)GTSC register	TS[15:0]
RSCFDn(CFD)RMNDy register	RMNSq
RSCFDn(CFD)RFCCx register	RFE
RSCFDn(CFD)RFSTSx register	RFMC[7:0], RFIF, RFMLT, RFFLL, RFEMP
RSCFDn(CFD)CFCCk register	When transmit/receive FIFO buffer is in receive mode: CFE
RSCFDn(CFD)CFSTSk register	When transmit/receive FIFO buffer is in receive mode: CFMC[7:0], CFFLL, CFEMP, CFTXIF, CFRXIF, CFMLT
RSCFDn(CFD)FESTS register	CFkEMP, RFxEMP
RSCFDn(CFD)FFSTS register	CFkFLL, RFxFLL
RSCFDn(CFD)FMSTS register	CFkMLT, RFxMLT
RSCFDn(CFD)RFISTS register	RFxIF
RSCFDn(CFD)CFRISTS register	CFkRXIF
RSCFDn(CFD)GTSTCFG register	RTMPS[6:0], C0ICBCE, C1ICBCE, C2ICBCE
RSCFDn(CFD)GTSTCTR register	RTME, ICBCTME

Note: Bits and flags in parentheses exist only in registers in CAN FD mode.

23.7 Reception Function

There are two reception types.

- Reception by receive buffers:
  Zero to 48 receive buffers can be shared by all channels. Since messages stored in receive buffers are overwritten at each reception, the latest receive data can always be read.
- Reception by receive FIFO buffers and transmit/receive FIFO buffers (receive mode):
  Eight receive FIFO buffers can be shared by all channels and three dedicated transmit/receive FIFO buffers are provided for each channel. Messages of up to the number of buffer stages specified with the RFDC[2:0] and CFDC[2:0] bits can be stored in FIFO buffers and can be read sequentially from the oldest.

23.7.1 Data Processing Using the Receive Rule Table

Data processing using the receive rule table allows dispatching of selected messages to the specified buffer. Data processing includes acceptance filter processing, DLC filter processing, routing processing, label addition processing, and mirror function processing.

Up to 128 receive rules can be registered per channel and up to (64 × number of channels) total receive rules can be registered in the entire module. (Up to 192 receive rules can be registered in this module

that has three channels.) Set receive rules for each channel. Receive rules cannot be shared with other channels. If receive rules are not set, no messages can be received. **Figure 23.6** illustrates how receive rules are registered.

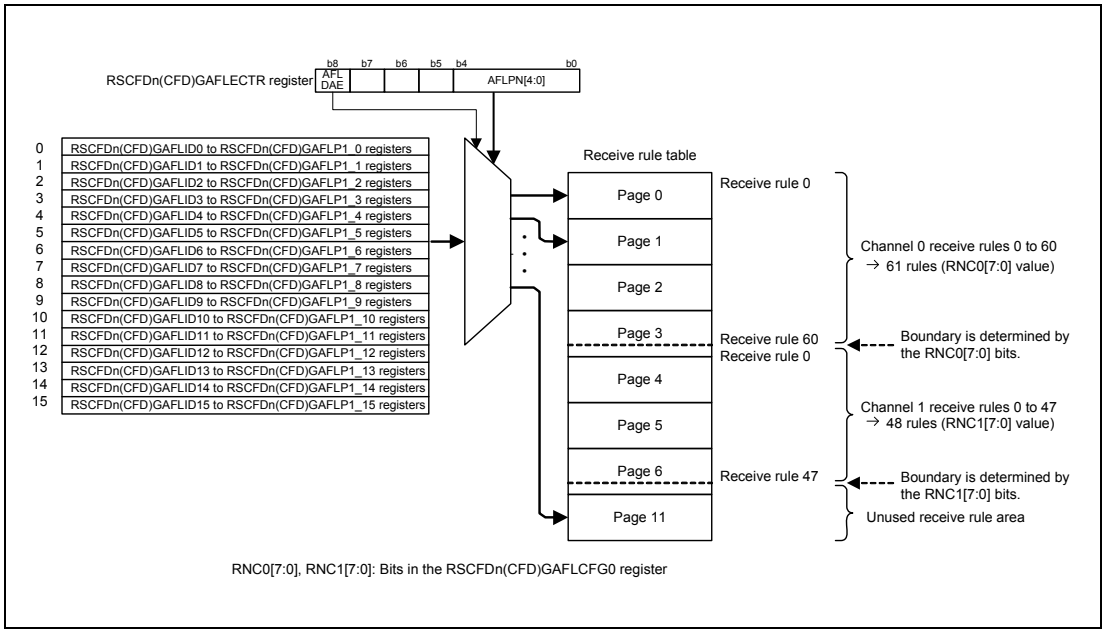


Figure 23.6 Entry of Receive Rules (for Setting Channel 0 and 1)

CAUTION

Receive rules for each channel must be set in contiguous blocks.
Channel 1 rules and channel 0 rules must be set separately.

Each receive rule consists of 16 bytes in the RSCFDn(CFD)GAFLIDj, RSCFDn(CFD)GAFLMj, RSCFDn(CFD)GAFLP0_j, and RSCFDn(CFD)GAFLP1_j registers (j = 0 to 15). The RSCFDn(CFD)GAFLIDj register is used to set GAFLID[28:0], GAFLIDE bit, GAFLRTR bit, and the mirror function, the RSCFDn(CFD)GAFLMj register is used to set mask, the RSCFDn(CFD)GAFLP0_j register is used to set label information to be added, DLC value, and storage receive buffer, and the RSCFDn(CFD)GAFLP1_j register is used to set storage FIFO buffer. Up to 16 receive rules can be set per page.

23.7.1.1 Acceptance Filter Processing

In the acceptance filter processing, the ID data, IDE bit, and RTR bit in a received message are compared with the ID data, IDE bit, and RTR bit set in the receive rule of the corresponding channel. When all these bits match, the message passes through the acceptance filter processing. The ID data, IDE bit, and RTR bit in the received message which correspond to the bits set to 0 (bits are not compared) in the RSCFDn(CFD)GAFLMj register are not compared and are regarded as matched.

Check begins with the receive rule of the minimum number for the corresponding channel. When all the bits to be compared in a received message match the bits set in the receive rule or when all the receive rules are compared without any match, filter processing stops. If there is no matching receive rule, the received message is not stored in the receive buffer or FIFO buffer.

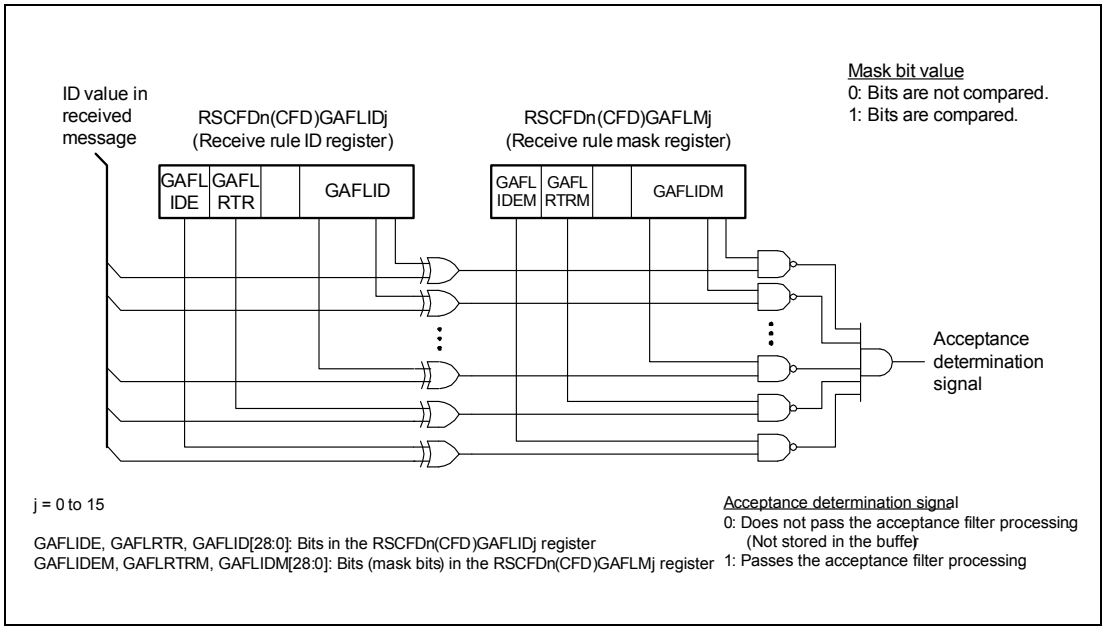


Figure 23.7 Acceptance Filter Function

23.7.1.2 DLC Filter Processing

When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), DLC filter processing is added to messages that passed through the acceptance filter processing. When the DLC value in a message is equal to or larger than the DLC value set in the receive rule, the message passes through the DLC filter processing.

When a message has passed through the DLC filter processing with the DRE bit in the RSCFDn(CFD)GCFG register set to 0 (DLC replacement is disabled), the DLC value in the received message is stored in the buffer. In this case, all the data bytes in the received message are stored in the buffer.

When a message has passed through the DLC filter processing with the DRE bit in the RSCFDn(CFD)GCFG register set to 1 (DLC replacement is enabled), the DLC value in the receive rule is stored in the buffer instead of the DLC value in the received message. In this case, a value of 00_H is stored in each data byte beyond the number of bytes which is indicated by the DLC value in the receive rule.

When the DLC value in the received message is smaller than that in the receive rule, the message does not pass through the DLC filter processing. In this case, the message is not stored in the receive buffer or the FIFO buffer and the DEF flag in the RSCFDn(CFD)GERFL register is set to 1 (a DLC error is present).

23.7.1.3 Routing Processing

Messages that passed through the acceptance filter processing and the DLC filter processing are stored in receive buffers, receive FIFO buffers, or transmit/receive FIFO buffers (set to receive mode or gateway mode). Message storage destination is set by the GAFLRMV and GAFLRMDP[6:0] bits in the RSCFDn(CFD)GAFLP0_j register (j = 0 to 15) and by the RSCFDn(CFD)GAFLP1_j register. Messages that passed through the acceptance filter processing and the DLC filter processing can be stored in up to eight buffers.

In CAN FD mode, if the payload length of the received message exceeds the payload storage size of the storage buffer, the CMPOF flag in the RSCFDnCFDGERFL register is set to 1 (payload overflow) and

the processing is handled according to the CMPOC bit in the RSCFDnCFDGCFG register. When the CMPOC bit is 0, the received message which exceeds the payload storage size is not stored in the buffer. When the CMPOC bit is 1, the received message is stored in the buffer with payloads exceeding the storage size being discarded, and depending on the DRE bit in the RSCFDnCFDGCFG register the received DLC value or the DLC value of the receive rule is stored in the buffer.

23.7.1.4 Label Addition Processing

It is possible to add 12-bit label information to messages that passed through the filter processing and store them in buffers. This label information is set in the GAFLPTR[11:0] bits in the RSCFDn(CFD)GAFLP0_j register.

23.7.1.5 Mirror Function Processing

The mirror function allows the CAN node to receive its own transmitted messages. The mirror function is made available by setting the MME bit in the RSCFDn(CFD)GCFG register to 1 (mirror function is enabled).

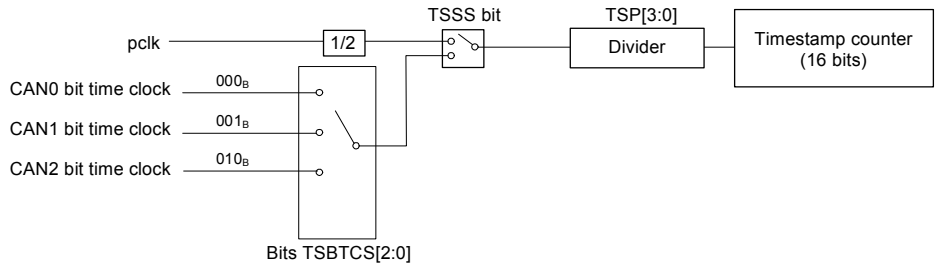
When the mirror function is in use, receive rules for which the GAFLLB bit in the RSCFDn(CFD)GAFLIDj register is set to 0 are used for data processing when receiving messages transmitted from other CAN nodes. When the CAN node is receiving its own transmitted messages, receive rules for which the GAFLLB bit is set to 1 are used for data processing.

23.7.1.6 Timestamp

The timestamp counter is a 16-bit free-running counter used for recording message receive time and transmission time. In classical CAN mode, the timestamp counter value is fetched at the start-of-frame (SOF) timing of a message. In CAN FD mode, the timestamp counter value is fetched at the timing set with the TSCCFG[1:0] bits in the RSCFDnCFDGFDCFG register. Then, the value is stored in a receive buffer or a FIFO buffer together with the message ID and data during data reception. The clock source of the timestamp counter is selected by the TSBTCS[2:0] and TSSS bits in the RSCFDn(CFD)GCFG register. In classical CAN mode, either pclk/2 or the CANm bit time clock (m = 0 to 2). In CAN FD mode, the clock source is selectable from pclk/2 or nominal CANm bit time clock. However, do not select the nominal CANm bit time clock of channels that handle CAN FD frames. The timestamp counter count source is obtained by dividing the selected clock source by the TSP[3:0] value in the RSCFDn(CFD)GCFG register.

When the CANm bit time clock or nominal CANm bit time clock is used as a clock source, the timestamp counter stops when the corresponding channel transitions to channel reset mode or channel halt mode. When the pclk/2 is used as a clock source, the timestamp function is not affected by channel mode.

The timestamp counter value is reset to 0000_H by setting the TSRST bit in the RSCFDn(CFD)GCTR register to 1.



TSBTCS[2:0], TSSS, TSP[3:0]: Bits in the RSCFDn(CFD)GCFG register

Note: When specifying pclk/2 as the timestamp counter count source, set bits TSBTCS[2:0] to 000_B.

Figure 23.8 Timestamp Function Block Diagram

23.8 Transmission Functions

There are three types of transmission. In classical CAN mode, transmittable payload length is 8 bytes in every transmission type. In CAN FD mode, transmittable payload length varies with transmission types.

- Transmission using transmit buffers:
  Each channel has 16 buffers. Transmittable payload length in CAN FD mode is 20 bytes. However, when transmit buffer merge mode is used, four buffers out of 16 buffers are allocated as a payload-only storage area and two buffers are able to transmit payloads with a length of more than 20 bytes.
- Transmission using transmit/receive FIFO buffers (transmit mode):
  Each channel has three FIFO buffers. Up to 128 messages can be contained in a single FIFO buffer. Transmittable payload length in CAN FD mode is 64 bytes. Each FIFO buffer is used with a link to a transmit buffer. Only the message to be transmitted next in a FIFO buffer becomes the target of transmit priority determination. Messages are transmitted sequentially on a first-in, first-out basis.
- Transmission using transmit queues:
  Up to 16 transmit buffers per channel can be allocated to the transmit queues. Transmittable payload length in CAN FD mode is 20 bytes. Transmit buffer ((16 × m) + 15) is used as an access window of a corresponding channel. Transmit buffers are allocated to transmit queues in descending order of buffer number. All messages in transmit queues, which are targets of priority determination, are transmitted in the order of ID number.

Figure 23.9 shows the allocation of transmit queues and transmit/receive FIFO buffer link.

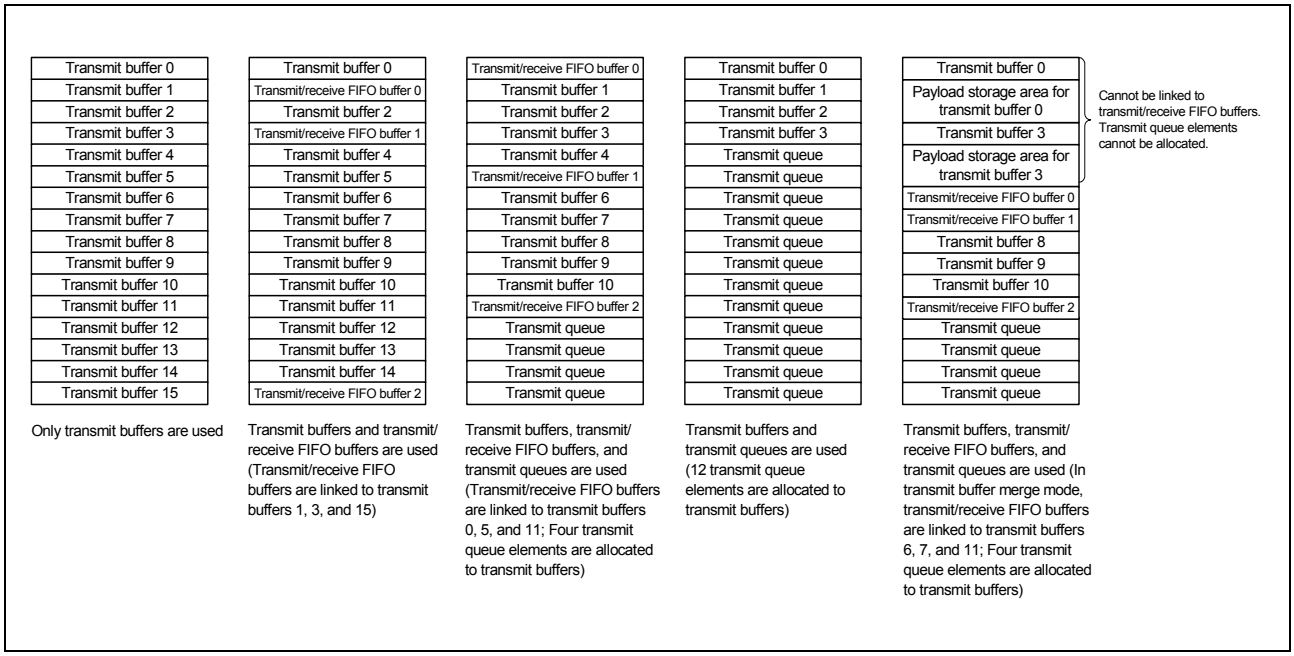


Figure 23.9 Allocation of Transmit Queues and Transmit/Receive FIFO Buffer Links

23.8.1 Transmit Priority Determination

If transmit requests are issued from multiple buffers or from the queue on the same channel, transmit priority is determined using one of the following methods.

- ID priority (TPRI bit = 0)
- Transmit buffer number priority (TPRI bit = 1)

All CAN channels use the setting of the TPRI bit in the RSCFDn(CFD)GCFG register.

When the TPRI bit is set to 0, messages are transmitted according to the priority of stored message IDs. ID priority conforms to the CAN bus arbitration specification defined in the CAN specifications. All IDs of pending transmit messages are targets of priority determination, regardless of whether they are stored in transmit buffers, transmit/receive FIFO buffers (set to transmit mode or gateway mode), or the transmit queue. If even a single transmit queue is used, select ID priority. When transmit/receive FIFO buffers are used, the oldest message in a FIFO buffer becomes the target of priority determination. When a message is being transmitted from a transmit/receive FIFO buffer, the next message in the FIFO buffer becomes the target of priority determination. When a transmit queue is used, all messages in the transmit queue are targets of priority determination. If the same ID is set for two or more buffers, the buffer with the smaller buffer number takes precedence.

When the TPRI bit is set to 1, the message in the transmit buffer with the minimum buffer number among all buffers with a transmit request is transmitted first. When transmit/receive FIFO buffers are linked to transmit buffers, transmit priority is determined according to linked transmit buffer numbers.

When messages are retransmitted due to an arbitration-lost or an error, transmit priority determination is made again according to the TPRI bit. When a 2-bit ECC error is detected in the priority determination processing, no message is transmitted (only when the EEFE bit in the RSCANnGCFG register is 1 in classical CAN mode).

23.8.2 Transmission Using Transmit Buffers

Setting the transmit request bit (TMTR bit in the RSCFDn(CFD)TMCp register) in a transmit buffer to 1 (transmission is requested) allows transmission of data frames or remote frames.

The transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCFDn(CFD)TMSTSp register (p = 0 to 47). When transmit completes successfully, the TMTRF[1:0] flag is set to 10_B (transmission has been completed (without transmit abort request)) or 11_B (transmission has been completed (with transmit abort request)).

23.8.2.1 Transmit Abort Function

With respect to transmit buffers for which the TMTRM bit in the RSCFDn(CFD)TMSTSp register is set to 1 (a transmit request is present), when the TMTAR bit in the RSCFDn(CFD)TMCp register is set to 1 (transmit abort is requested), the transmit request is canceled. When transmit abort is completed, the TMTRF[1:0] flag in the RSCFDn(CFD)TMSTSp register is set to 01_B (transmit abort has been completed) and the transmit request is canceled (clearing the TMTRM bit to 0).

A message that is being transmitted or a message to be transmitted next according to the transmit priority determination cannot be aborted. However, when an arbitration-lost or an error occurs during transmission of a message for which the TMTAR bit is set to 1, retransmission is not performed.

23.8.2.2 One-Shot Transmission Function (Retransmission Disabling Function)

When the TMOM bit in the RSCFDn(CFD)TMCp register is set to 1 (one-shot transmission is enabled), transmission is performed only once. Even if an arbitration-lost or an error occurs, retransmission is not performed.

The one-shot transmit result is shown by the TMTRF[1:0] flag in the corresponding RSCFDn(CFD)TMSTSp register. When one-shot transmission completes successfully, the TMTRF[1:0] flag is set to 10_B or 11_B. When an arbitration-lost or an error occurs, the TMTRF[1:0] flag is set to 01_B (transmit abort has been completed).

23.8.2.3 Transmit Buffer Merge Mode (Only in CAN FD Mode)

Transmit buffers can transmit messages with a payload length of 20 bytes, but can transmit messages with a payload length of up to 64 bytes by merging three transmit buffers in transmit buffer merge mode.

Setting the TMME bit to 1 in the RSCFDnCFDCmFDCFG register enables transmit buffer merge mode. In this mode, six buffers per channel become a merge area and two sets of transmit buffers (16 × m) + 0 to (16 × m) + 2 and transmit buffers (16 × m) + 3 to (16 × m) + 5 are merged. A transmission request is made by the first transmit buffer, and subsequent two buffers are used as a payload storage area. Do not set the transmission request bit (TMTR bit in the RSCFDnCFDTMCp register) and the transmission abort request bit (TMTAR bit in the RSCFDnCFDTMCp register) to 1 for transmit buffers except for the first buffer.

While transmit buffer merge mode is enabled, do not link the transmit/receive FIFO buffer to six merged buffers or allocate it to the transmit queue.

23.8.3 Transmission Using FIFO Buffers

Multiple messages can be stored in a single transmit/receive FIFO buffers, up to the number specified by the FIFO buffer depth, which is set by the CFDC[2:0] bits in the RSCFDn(CFD)CFCCk register (k = 0 to 8). Messages are transmitted sequentially on a first-in, first-out basis.

Each transmit/receive FIFO buffer is linked to a transmit buffer selected by the CFTML[3:0] bits in the RSCFDn(CFD)CFCCk register. When the CFE bit in the RSCFDn(CFD)CFCCk register is set to 1 (transmit/receive FIFO buffers are used), transmit/receive FIFO buffers become targets of transmit priority determination. Priority of only the next transmit message is determined in the FIFO buffer.

When the CFE bit is set to 0 (no transmit/receive FIFO buffer is used), the CFEMP flag is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) at the timing below.

- The transmit/receive FIFO buffer becomes empty immediately if the message in it is not being transmitted or is not to be transmitted next.
- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, arbitration-lost or the transition to channel halt mode in the case that a message in it is being transmitted or to be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages cannot be stored in FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

23.8.3.1 Interval Transmission Function

A message transmission interval time can be set to space the transmission of messages from the same FIFO buffer when using a transmit/receive FIFO buffer set to transmit mode or gateway mode.

Immediately after the first message has been transmitted successfully from the FIFO buffer with the CFE bit in the RSCFDn(CFD)CFCCk register set to 1, the interval timer starts counting (after EOF7 of the CAN protocol). After that, when the interval time has passed, the next message is transmitted. The interval timer stops in channel reset mode or by clearing the CFE bit to 0.

The interval time is set by the CFITT[7:0] bits in the RSCFDn(CFD)CFCCk register. When the interval timer is not used, set the CFITT[7:0] bits to 00_H.

Select an interval timer count source using the CFITR and CFITSS bits in the RSCFDn(CFD)CFCCk register. When the interval timer is not used, set the CFITT[7:0] bits to 00_H. When the CFITR and CFITSS bits are set to 00_B, the count source is obtained by dividing pclk/2 by the value of the ITRCP[15:0] bits. When the CFITR and CFITSS bits are set to 10_B, the count source is obtained by dividing pclk/2 by (the value of the ITRCP[15:0] bits in the RSCFDn(CFD)GCFG register × 10). When the CFITR and CFITSS bits are set to $x1_B$, the CANm bit time clock becomes a count source in classical CAN mode and the nominal CANm bit time clock becomes a count source in CAN FD mode. (Use this count source only for the channel does not handle the CAN FD frames in CAN FD mode.)

The interval time is calculated by the following equations where M is the value of ITRCP[15:0] and N is the value of CFITT[7:0].

- When CFITR and CFITSS = 00_B:

$$\frac{1}{\text{pclk frequency}} \times 2 \times M \times N$$

- When CFITR and CFITSS = 10_B:

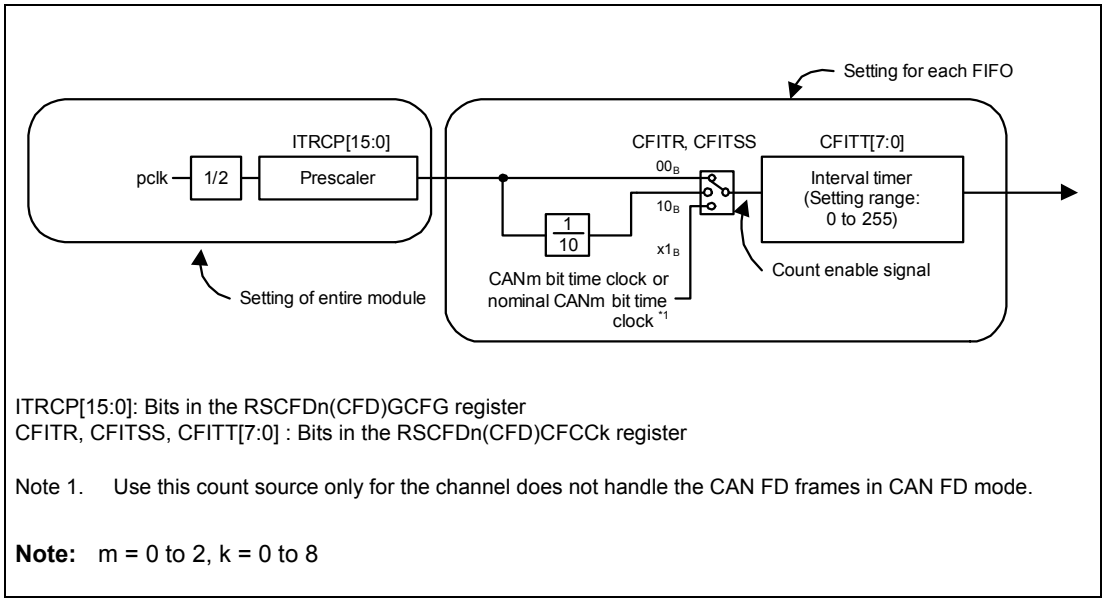
$$\frac{1}{\text{pclk frequency}} \times 2 \times M \times 10 \times N$$

- When CFITR and CFITSS = $x1_B$:

$$\text{Classical CAN mode: } \frac{1}{\text{CANm bit time clock frequency}} \times N$$

$$\text{CAN FD mode: } \frac{1}{\text{Nominal CANm bit time clock frequency}} \times N$$

Figure 23.10 shows the interval timer block diagram.



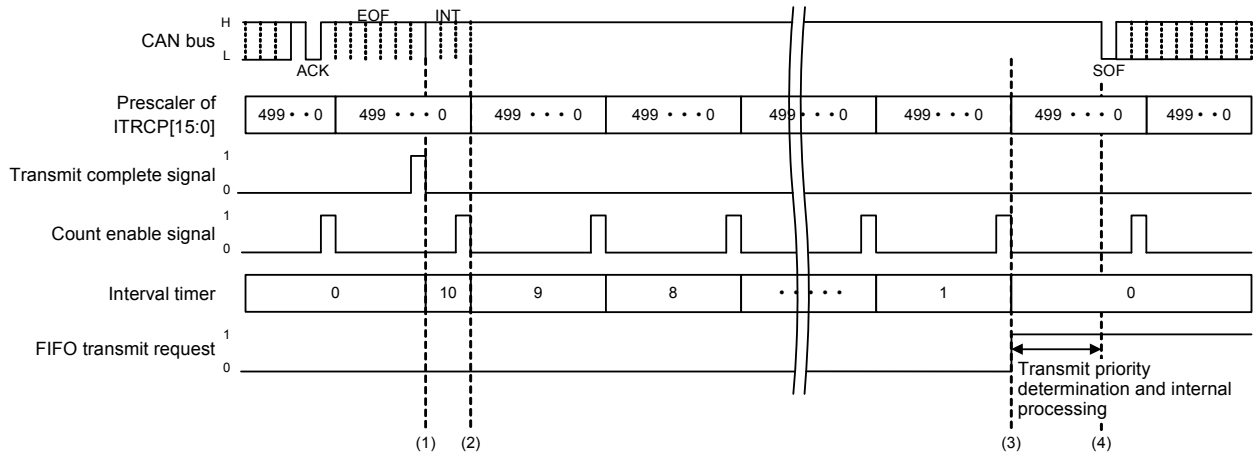
ITRCP[15:0]: Bits in the RSCFDn(CFD)GCFG register
CFITR, CFITSS, CFITT[7:0] : Bits in the RSCFDn(CFD)CFCCk register

Note 1. Use this count source only for the channel does not handle the CAN FD frames in CAN FD mode.

Note: m = 0 to 2, k = 0 to 8

Figure 23.10 Interval Timer Block Diagram

Figure 23.11 shows the interval timer timing diagram.



Interval time (logical value) = 2/fPBA × (set ITRCP[15:0] value) × set CFITT[7:0] value

- fPBA: Frequency of pclk
- ITRCP[15:0]: Bits in the RSCFDn(CFD)GCFG register (The set value is 500 in this figure.)
- CFITT[7:0]: Bits in the RSCFDn(CFD)CFCCk register (The set value is 10 in this figure.)

Figure 23.11 Interval Timer Timing Chart

(1) The interval timer starts counting upon completion of transmission. Since the prescaler is not initialized at the time of transmission completion, the first interval time contains an error of up to one count of the interval timer.

(2) The interval timer is decremented by 1 upon the next count enable signal.

(3) When the interval timer has decreased to 0, the transmit/receive FIFO buffer issues a transmit request.

(4) The transmit/receive FIFO buffer is determined for the next transmission by the priority determination, it starts transmitting data. Transmission starts usually with a delay of three CANm bit time clock cycles or less from the issue of transmit request. If multiple internal processes (such as receive filter processing, message routing, and transmit priority determination) take place in all channels, a delay of up to 774 cycles of the pclk may be generated.

23.8.4 Transmission Using Transmit Queues

Three to sixteen buffers (up to 10 buffers in transmit buffer merge mode) are allocated to a transmit queue for each channel, and transmit buffer ((16 × m) + 15) is used as an access window of a corresponding channel.

All messages in a transmit queue are targets of transmit priority determination and are transmitted in the ID priority order regardless of storage sequence. If two messages having the same ID are stored in a transmit queue, these messages are not always transmitted in the order of their storage in the transmit queue.

Setting the TXQE bit in the RSCFDn(CFD)TXQCCm register to 0 disables transmit queues. When the TXQE bit is set to 0, the TXQEMP flag in the RSCFDn(CFD)TXQSTSm register is set to 1 (the transmit queue contains no messages (transmit queue empty)) at the timing below.

- The transmit queue becomes empty immediately when no message in it is being transmitted or will be transmitted next.

- The transmit queue becomes empty after transmission completion, CAN bus error detection, arbitration-lost, or the transition to channel halt mode when a message in it is being transmitted or will be transmitted next.

When the TXQE bit is cleared to 0, all messages in transmit queues are lost and messages cannot be stored in transmit queues. Confirm that the TXQEMP flag is set to 1 before setting the TXQE bit to 1 again.

23.8.5 Transmit Data Padding (Only in CAN FD Mode)

When the payload length indicated by the set DLC value in a transmit message exceeds the payload storage area size of a buffer to be used for transmission, excessive payloads are padded by CC_H.

This processing is performed in the following cases when the transmit buffer merge mode is disabled (TMME bit in the RSCFDnCFDCmFDCFG register is 0).

- Transmit/receive FIFO set to transmission or gateway mode:
  When the payload length of the transmit DLC exceeds the transmit/receive FIFO payload storage area size set by the CFPLS[2:0] bits in the RSCFDnCFDCFCCk register
- Transmit buffer (including transmit queue):
  When the payload length of the transmit DLC exceeds 20 bytes

When the transmit buffer merge mode is enabled, no transmit data is padded in any transmission using a transmit buffer, transmit/receive FIFO buffer, or transmit queue. At this time, do not set a payload length more than the payload storage size of the buffer for transmitting as the DLC value in the transmit message.

23.8.6 Transmit History Function

Information about transmission-completed messages can be stored in the transmit history buffer. Each channel has a single transmit history buffer that can contain 16 sets of transmit history data.

A message transmit source buffer type can be selected by the THLDTE bit in the RSCFDn(CFD)THLCCm register. The THLEN bit in the RSCFDn(CFD)CFIDk register (k = 0 to 8) determines whether transmit history data is stored for each message.

In classical CAN mode, the TMTSCE bit in the RSCANnGCFG register can be used to set whether to include a timestamp value in the transmit history data. In CAN FD mode, a timestamp value is always included.

The following information on a transmitted message will be stored in the transmit history buffer after the successful completion of transmission.

Storage of the transmit history data after the successful completion of transmission may take up to 134 cycles of pclk in classical CAN mode or 386 cycles of pclk in CAN FD mode.

- Buffer type — 001_B: Transmit buffer
010_B: Transmit/receive FIFO buffer
100_B: Transmit queue
- Buffer number — Number of source transmit buffer, transmit queue, or transmit/receive FIFO buffer. This number depends on buffer types. See **Table 23.179**.
- Label data — Label information of the transmit message
- Timestamp — Timestamp value of the transmit message

(When the TMTSCE bit is 1 in classical CAN mode)

Table 23.179 Transmit History Data Buffer Numbers

Buffer type / Buffer No.	001_B	010_B	100_B
0000_B	Transmit buffer 16 × m + 0	Buffer numbers of the transmit buffer linked to the transmit/receive FIFO buffer by the CFTML[3:0] bits in the RSCFDn(CFD)CFCCk register (k = 0 to 8)	Buffer numbers of the transmit buffer allocated to the transmit queue that performed transmission
0001_B	Transmit buffer 16 × m + 1		
0010_B	Transmit buffer 16 × m + 2		
0011_B	Transmit buffer 16 × m + 3		
0100_B	Transmit buffer 16 × m + 4		
0101_B	Transmit buffer 16 × m + 5		
0110_B	Transmit buffer 16 × m + 6		
0111_B	Transmit buffer 16 × m + 7		
1000_B	Transmit buffer 16 × m + 8		
1001_B	Transmit buffer 16 × m + 9		
1010_B	Transmit buffer 16 × m + 10		
1011_B	Transmit buffer 16 × m + 11		
1100_B	Transmit buffer 16 × m + 12		
1101_B	Transmit buffer 16 × m + 13		
1110_B	Transmit buffer 16 × m + 14		
1111_B	Transmit buffer 16 × m + 15		

Label data is used to identify each message. Unique label data can be added to each message transmitted from a transmit buffer, transmit queue, or transmit/receive FIFO buffer.

The timestamp value is fetched from the timestamp counter at the SOF (start of frame) timing of the message. For details about the timestamp counter, see Section 23.7.1.6, Timestamp.

Transmit history data can be read from the RSCFDn(CFD)THLACCm register. If an attempt is made to store new transmit history data while the buffer is full, the buffer overflows and the new data is discarded.

23.9 Gateway Function

When a transmit/receive FIFO buffer is set to gateway mode, receive messages can be transmitted from an arbitrary channel without CPU intervention.

When the CFM[1:0] bits in the RSCFDn(CFD)CFCCk register are set to 10_B (gateway mode) and the transmit/receive FIFO buffer of a channel transmitting a message is selected in RSCAN0GAFLP1j register, messages that pass through filter processing according to the receive rule are stored in the specified transmit/receive FIFO buffer and are automatically transmitted from the buffer.

Messages stored in a transmit/receive FIFO buffer are transmitted sequentially on a first-in, first-out basis. Only the message to be transmitted next becomes the target of transmit priority determination.

Transmit/receive FIFO buffers in the gateway mode are disabled by setting the CFE bit in the RSCFDn(CFD)CFCCk register to 0 and the CFEMP flag becomes 1 according to the timing below.

- The transmit/receive FIFO buffer becomes empty immediately when the message in it is not being transmitted and will not to be transmitted next.

- The transmit/receive FIFO buffer becomes empty after transmission completion, CAN bus error detection, or arbitration-lost when the message in it is being transmitted or will be transmitted next.

When the CFE bit is cleared to 0, all messages in transmit/receive FIFO buffers are lost and messages can no longer be stored in transmit/receive FIFO buffers. Confirm that the CFEMP flag is set to 1 before setting the CFE bit to 1 again.

23.9.1 CAN-CAN FD Gateway (Only in CAN FD Mode)

When the gateway function is used in CAN FD mode, a frame to be transmitted can be replaced with a classical CAN frame or a CAN FD frame.

Setting the GWEN bit in the RSCFDnCFDCmFDCFG register to 1 enables the CAN-CAN FD gateway. The FDF and BRS bits in the transmit frame can be selected by the GWFDF and GWBRS bits in the RSCFDnCFDCmFDCFG register. When the DLC value of the received CAN frame is 1001_B or more and the GWFDF bit is 1 (CAN FD frame), the DLC value is replaced with 1000_B.

When the CAN-CAN FD gateway is enabled, do not perform routing for the following frames.

- CAN FD frames with a payload length of more than 8 bytes
- Remote frames

When the CAN-CAN FD gateway is enabled, the following frame should be transmitted in the channel by setting of GWFDF.

- When GWFDF bit is set to 0, only classical CAN frame should be transmitted.
- When GWFDF bit is set to 1, only CAN FD frame should be transmitted.

23.10 Test Function

The test function is classified into communication tests and global tests.

- Communication tests: Performed for each channel.
  - Standard test mode
  - Listen-only mode
  - Self-test mode 0 (external loopback mode)
  - Self-test mode 1 (internal loopback mode)
  - Restricted operation mode (only in CAN FD mode)
- Global tests: Performed for the entire module
  - RAM test (read/write test)
  - Inter-channel communication test [CRC error test enabled]

23.10.1 Standard Test Mode

CRC tests are enabled in standard test mode. The CRC value calculated by the RS-CANFD module based on the transmit message or receive message is stored in the register. This CRC value is stored in the CRCREG[14:0] bits in the RSCFDn(CFD)CmERFL register when the message is a classical CAN frame (CRC length = 15 bits) or in the CRCREG[20:0] bits in the RSCFDnCFDCmFDCRC register when the message is a CAN FD frame (CRC length = 17 or 21 bits). Use the inter-channel communication test function for CRC error tests. For details, see Section 23.10.6.1, CRC Error Test.

23.10.2 Listen-Only Mode

Listen-only mode allows reception of data frames and remote frames. Only recessive bits are transmitted on the CAN bus, and the ACK bit, overload flag, and active error flag are not transmitted.

Listen-only mode is available for detecting the communication speed.

Do not make a transmit request from any buffer or queue in listen-only mode.

Figure 23.12 shows the connection when listen-only mode is selected.

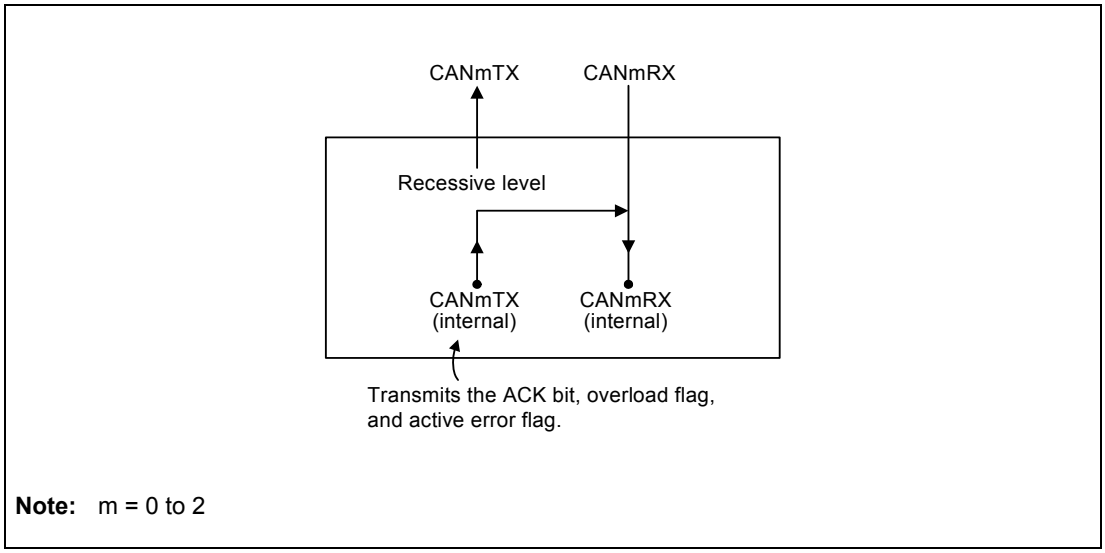


Figure 23.12 Connection when Listen-Only Mode is Selected

23.10.3 Self-Test Mode (Loopback Mode)

In self-test mode, transmitted messages are compared with the receive rule of the own channel and the messages are stored in a buffer if they have passed through the filter processing. Messages transmitted from other CAN nodes are compared only with the receive rule for which the GAFLLB bit in the RSCFDn(CFD)GAFLIDj register (j = 0 to 15) is set to 0 (when a message transmitted from another CAN node is received).

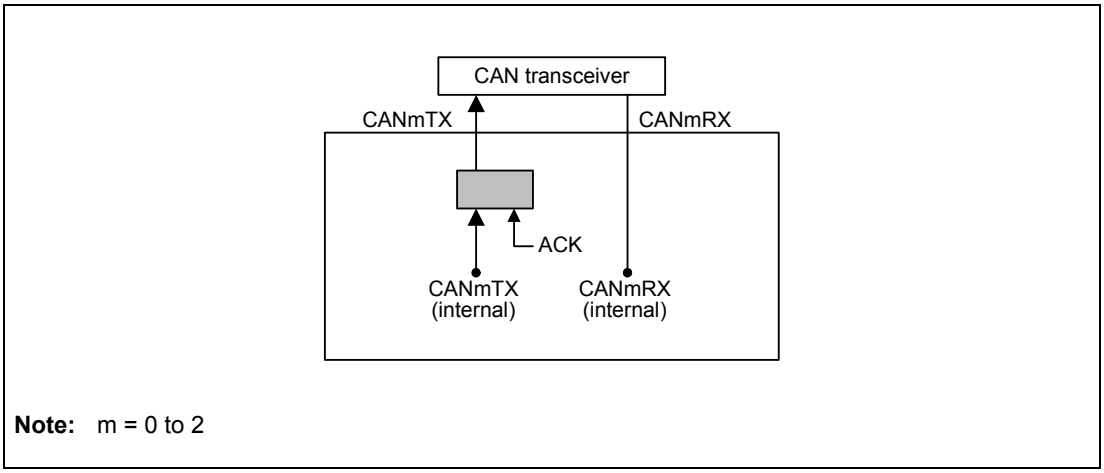
If the mirror function and self-test mode are both enabled, the self-test mode setting takes precedence.

23.10.3.1 Self-Test Mode 0 (External Loopback Mode)

Self-test mode 0 is used to perform a loopback test within a channel including the CAN transceiver.

In self-test mode 0, transmitted messages are handled as messages received through the CAN transceiver and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

Figure 23.13 shows the connection when self-test mode 0 is selected.



Note: m = 0 to 2

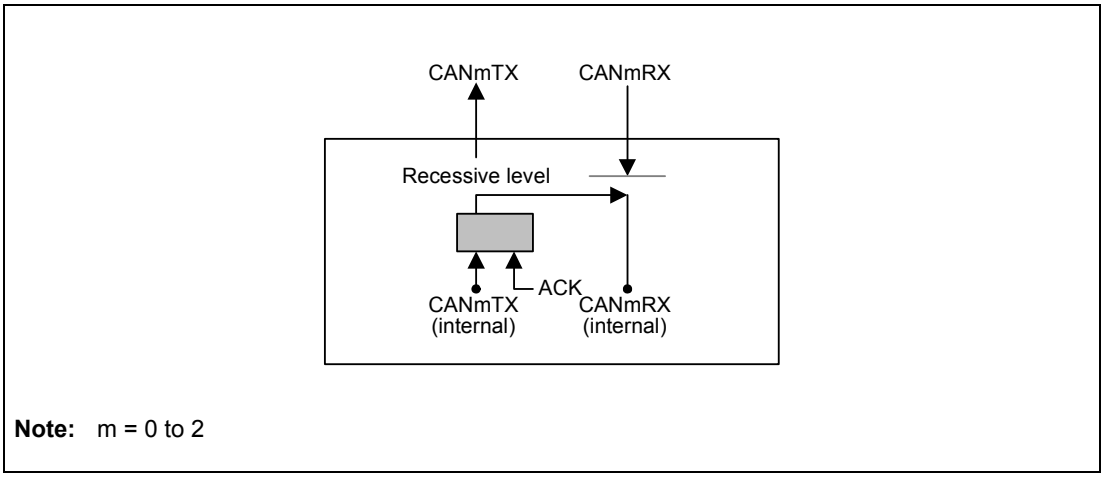
Figure 23.13 Connection when Self-Test Mode 0 is Selected

23.10.3.2 Self-Test Mode 1 (Internal Loopback Mode)

In self-test mode 1, transmitted messages are handled as received messages and are stored in a buffer. An ACK bit is generated to receive messages transmitted from the own CAN node.

In self-test mode 1, internal feedback from the internal CANmTX pin (m = 0 to 2) to the internal CANmRX pin is performed. The external CANmRX pin input is isolated. The external CANmTX pin outputs only recessive bits.

Figure 23.14 shows the connection when self-test mode 1 is selected.



Note: m = 0 to 2

Figure 23.14 Connection when Self-Test Mode 1 is Selected

23.10.4 Restricted Operation Mode (Only in CAN FD Mode)

In restricted operation mode, an ACK bit is generated when a valid data frame and a remote frame have been received, but these frames are not transmitted even if an error frame or an overload frame transmit condition is detected. When a condition is detected, operation is suspended until the bus idle state comes for resynchronization with the CAN communication. The receive error counter (REC) and the transmit error counter (TEC) do not change due to an error.

A desired transmission request can be made for transmission without restrictions.

23.10.5 RAM Test

The RAM test function allows accesses to all CAN RAM addresses.

When the RAM test function is used, the RAM is divided into pages of 256 bytes each. RAM test page is set by the RTMPS[6:0] bits in the RSCFDn(CFD)GTSTCFG register. Data in the set page can be read from and written to the RSCFDn(CFD)RPGACCr register (r = 0 to 63). The available total RAM size is 7680 bytes ($1E00_H$)*1 in classical CAN mode or 10656 bytes ($29A0_H$) in CAN FD mode.

In CAN FD mode, do not access more than 160 bytes in the last page (RTMPS[6:0] = 29_H) during RAM test.

Note 1. In classical CAN mode, 7584 bytes ($1DA0_H$) are used. To shorten the test time, the test of 96 bytes in the last page can be omitted.

23.10.6 Inter-Channel Communication Test

The inter-channel communication test function allows communication test by internally connecting CAN channels to each other. During this test, channels are isolated from the external CAN bus.

Before starting data transmission/reception in channel communication mode, make transmission/reception settings for each channel.

Channels not included in the test must be placed in channel halt mode. Set the channel(s) not participating in test to Channel halt mode.

Figure 23.15 shows the connection for inter-channel communication test.

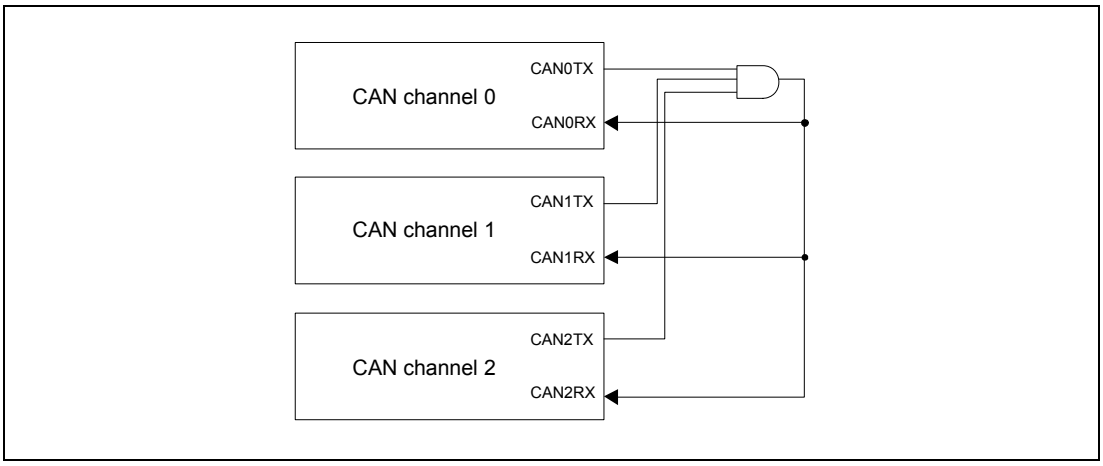


Figure 23.15 Connection for Inter-Channel Communication Test

23.10.6.1 CRC Error Test

A CRC error test is enabled during an inter-channel communication test. The following shows an example of channel 0 CRC error test procedure during a communication test between channel 0 and channel 1.

Preconditions

- Inter-channel communication test is enabled.
- Channel 0 and channel 1 are in standard test mode.

Procedure

1. Make a setting to send a message from the transmit buffer p of channel 1.
2. Set the CRCT bit in the RSCFDn(CFD)C0CTR register to 1 (to enable inversion of the first bit in the received ID field).
3. Set the TMTR bit in the RSCFDn(CFD)TMCp register to 1 (to issue a transmission request to the transmit buffer p of channel 1).
4. Wait for occurrence of a CAN0 error interrupt due to a channel bus error.
5. Read the CRCREG[14:0] bits in the RSCFDn(CFD)CmERFL register or the CRCREG[20:0] bits in the RSCFDnCFDCmFDCRC register of channel 0 and channel 1, and confirm that the CRC values are different on the transmission and the reception side.
6. Confirm that the CERR bit in RSCFDm(CFD)C0ERFL is 1 (CRC error detected).

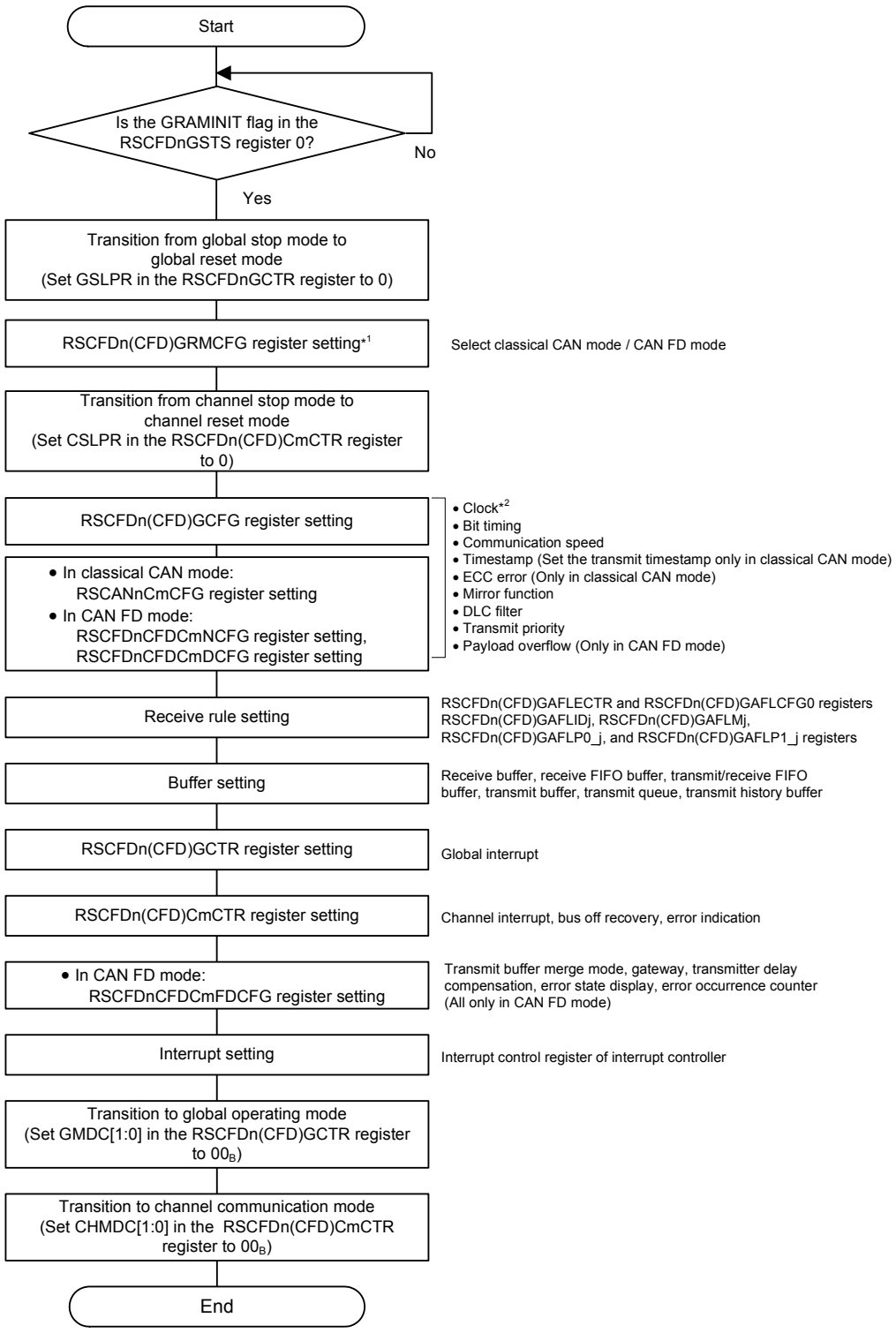
The CRC error test function generates an incorrect CRC value by inverting the first bit in the received ID field. Therefore, note that not a CRC error but a stuff error (continuous 6-bit data of the same level) is detected when a message in which ID's upper 5-bit value is 10000B or ID's upper 6-bit value is 011111B is received.

The CRC generation circuit of the RS-CANFD module is contained in the protocol controller of each channel. Another CRC calculation test is not necessary during transmission because the same circuit is used for both transmission and reception.

23.11 RS-CANFD Setting Procedure

23.11.1 Initial Settings

The RS-CANFD module initializes the CAN RAM after the MCU is reset. The RAM initialization time is 1897 cycles of the pclk. The GRAMINIT flag in the RSCFDn(CFD)GSTS register is set to 1 (CAN RAM initialization is ongoing) during the RAM initialization and is cleared to 0 (CAN RAM initialization is finished) when the initialization is completed. Make CAN settings after the GRAMINIT flag is cleared to 0. **Figure 23.16** shows the CAN setting procedure after the MCU is reset.



Note 1. The RSCANnGRMCFG register and the RSCFDnCFDGRMCFG register are the same register. Therefore, set either one of the registers.

Note 2. For the setting of CAN clock frequency, see Table 23.8, Setting Range example of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/D1L/D1M.

Note: m = 0 to 2, j = 0 to 15

Figure 23.16 CAN Setting Procedure after the MCU is Reset

23.11.1.1 Clock Setting

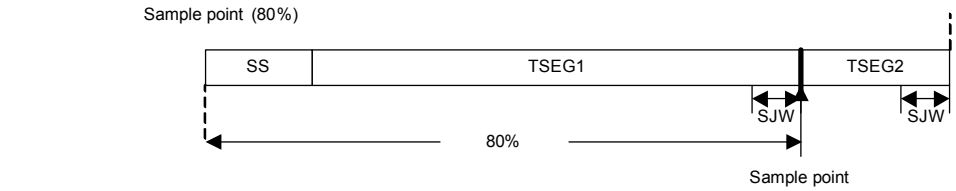
Set the CAN clock (fCAN) as a clock source of the RS-CANFD module. Select the clk_xincan or clkc using the DCS bit in the RSCFDn(CFD)GCFG register.

23.11.1.2 Bit Timing Setting

In the CAN protocol, one bit of a communication frame consists of three segments SS, TSEG1, and TSEG2, of which two segments TSEG1 and TSEG2 can be set by the corresponding register for each channel. In classical CAN mode, set these two segments in the RSCANnCmCFG register. Two bit rates (nominal bit rate and data bit rate) are provided for CAN FD mode. Set the nominal bit rate in the RSCFDnCFDCmNCFG register and set the data bit rate in the RSCFDnCFDCmDCFG register. Sample point timing can be determined by setting these two segments. This timing can be adjusted in units of 1 Time Quantum (hereafter referred to as Tq). A single Tq is the cycle of clock obtained by dividing the clock selected by the DCS bit in the RSCFDn(CFD)GCFG register. Set a division ratio by the BRP[9:0] bits in the RSCANnCmCFG register in classical CAN mode (CANmTq clock), and by the NBRP[9:0] bits in the RSCFDnCFDCmNCFG register and the DBRP[7:0] bits in the RSCFDnCFDCmDCFG register in CAN FD mode (CANmTq(N) clock and CANmTq(D) clock).
Be sure to specify the same value for both NBRP[9:0] and DBRP[7:0].
To specify a different value for the nominal bit rate and the data bit rate, change the values of the RSCFDnCFDCmNCFG.NTSEG1[6:0] and NTSEG2[4:0] bits and RSCFDnCFDCmDCFG.DTSEG1[3:0] and DTSEG2[2:0] bits, respectively.

When the TDCE bit is set to 1(Transmitter delay compensation is enabled) in the RSCFDnCFDCmFDCFG register, set the equal value of 1 or less to the bits NBRP[9:0] and DBRP[7:0].

Figure 23.17 shows the bit timing chart. **Table 23.180** shows an example of bit timing setting.



Classical CAN mode:
SS = 1 Tq fixed
Set TSEG1 to a range of 4 Tq to 16 Tq
Set TSEG2 to a range of 2 Tq to 8 Tq
Set SJW to a range of 1 Tq to 4 Tq
Set SS + TSEG1 + TSEG2 to a range of 8 Tq to 25 Tq
TSEG1 > TSEG2 ≥ SJW

CAN FD mode:
<Nominal bit rate>
SS = 1 Tq fixed
Set TSEG1 to a range of 4 Tq to 128 Tq
Set TSEG2 to a range of 2 Tq to 32 Tq
Set SJW to a range of 1 Tq to 32 Tq
Set SS + TSEG1 + TSEG2 to a range of 8 Tq to 161 Tq
TSEG1 > TSEG2 ≥ SJW

<Data bit rate>
SS = 1 Tq fixed
Set TSEG1 to a range of 2 Tq to 16 Tq
Set TSEG2 to a range of 2 Tq to 8 Tq
Set SJW to a range of 1 Tq to 8 Tq
Set SS + TSEG1 + TSEG2 to a range of 5 Tq to 25 Tq
TSEG1 ≥ TSEG2 ≥ SJW
Set SJW to 1 Tq when TSEG1 = 2 Tq and TSEG2 = 2 Tq

- SS (synchronization segment):
  The SS is a segment that performs synchronization by monitoring the edge from recessive to dominant bits in the Interframe Space.
  Interframe Space consists of Intermission, Suspend Transmission, and Bus Idle. All nodes can start transmission during Bus Idle.
- TSEG1 (Time segment 1):
  TSEG1 is a segment that absorbs physical delay on the CAN network. The physical delay on the CAN network is twice the total of the delay on the CAN bus, the delay in the input comparator, and the delay in the output driver.
- TSEG2 (Time segment 2):
  TSEG2 is a segment that compensates phase error due to an error in frequency.
- SJW (Resynchronization jump width)
  The SJW is a length to extend or reduce the time segment to compensate for an error in phase due to phase error.

Figure 23.17 Bit Timing Chart

Table 23.180 Example of Bit Timing Setting

1 Bit	Set Value (Tq)				Sample Point (%) Note: See Figure 23.17.
	SS	TSEG1	TSEG2	SJW	
5 Tq*1	1	2	2	1	60.00
8 Tq	1	4	3	1	62.50
	1	5	2	1	75.00
10 Tq	1	6	3	1	70.00
	1	7	2	1	80.00
16 Tq	1	10	5	1	68.75
	1	11	4	1	75.00
20 Tq	1	12	7	1	65.00
	1	13	6	1	70.00
50 Tq*1	1	39	10	4	80.00

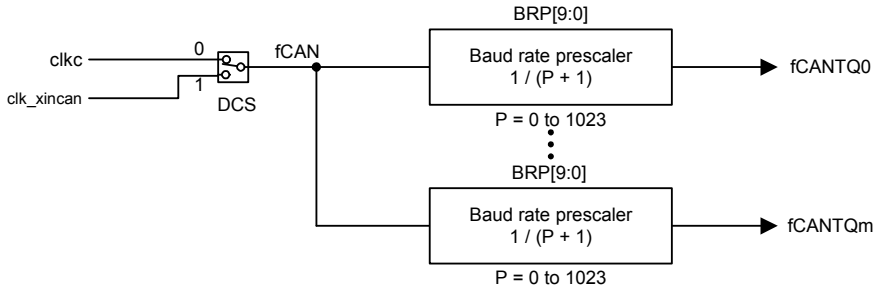
Note 1. Only in CAN FD mode

23.11.1.3 Communication Speed Setting

Set the CAN communication speed for each channel using the fCAN, baud rate prescaler division value, and Tq count per bit time. For CAN FD mode, set two types of transmission rate (arbitration phase and data phase) for each channel.

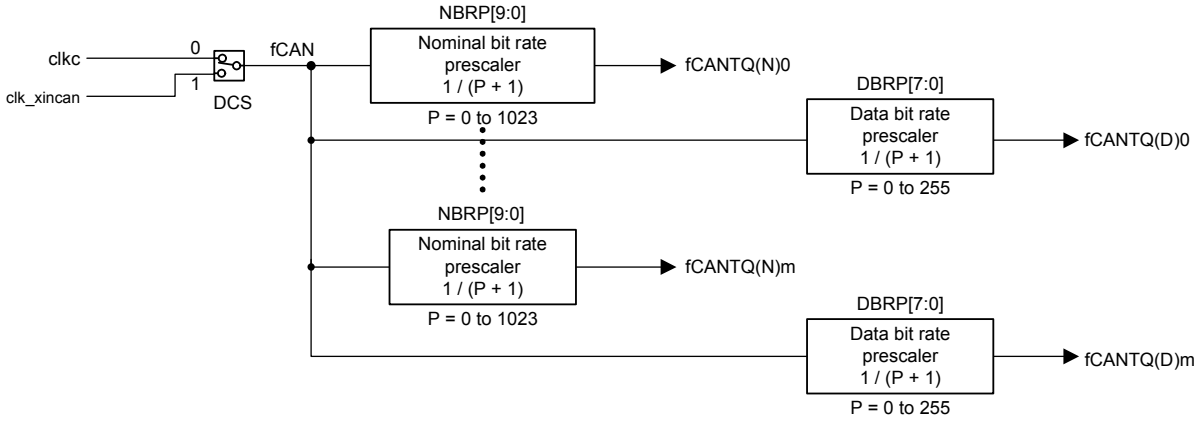
Figure 23.18 shows the CAN clock control block diagram, and **Table 23.181** shows an example of the communication speed setting.

- Classical CAN mode



$$\text{Communication speed} = \frac{\text{fCAN}}{\text{Baud rate prescaler division value} \times (\text{Tq count of 1 bit time})}$$

- CAN FD mode



$$\text{Nominal bit rate} = \frac{\text{fCAN}}{\text{Nominal bit rate prescaler division value} \times (\text{Tq count of 1 nominal bit time})}$$

$$\text{Data bit rate} = \frac{\text{fCAN}}{\text{Data bit rate prescaler division value} \times (\text{Tq count of 1 data bit time})}$$

Note: m = 0 to 2
DCS: Bit in the RSCFDn(CFD)GCFG register
BRP[9:0]: Bits in the RSCANnCmCFG register
NBRP[9:0]: Bits in the RSCFDnCFDCmNCFG register
DBRP[7:0]: Bits in the RSCFDnCFDCmDCFG register
fCAN: CAN clock
fCANTQm: CANmTq clock
fCANTQ(N)m: CANmTq(N) clock
fCANTQ(D)m: CANmTq(D) clock

Figure 23.18 CAN Clock Control Block Diagram

Table 23.181 Example of Communication Speed Setting (Classical CAN mode)

fCAN / Communication speed	40 MHz	20 MHz	16 MHz	8 MHz
1 Mbps	8 Tq (5) 20 Tq (2)	10 Tq (2) 20 Tq (1)	8 Tq (2) 16 Tq (1)	8 Tq (1)
500 Kbps	8 Tq (10) 20 Tq (4)	10 Tq (4) 20 Tq (2)	8 Tq (4) 16 Tq (2)	8 Tq (2) 16 Tq (1)
250 Kbps	8 Tq (20) 20 Tq (8)	10 Tq (8) 20 Tq (4)	8 Tq (8) 16 Tq (4)	8 Tq (4) 16 Tq (2)
125 Kbps	8 Tq (40) 20 Tq (16)	10 Tq (16) 20 Tq (8)	8 Tq (16) 16 Tq (8)	8 Tq (8) 16 Tq (4)

Note: Values in ( ) are baud rate prescaler division values.

Table 23.182 Example of Transmission Rate Setting (Nominal Bit Rate and Data Bit Rate in CAN FD Mode)

fCAN / Communiation rate	40 MHz	20 MHz
Nominal bit rate 1 Mbps Data bit rate 5 Mbps	Nominal bit rate 40 Tq (1) Data bit rate 8 Tq (1)	None
Nominal bit rate 500 kbps Data bit rate 2 Mbps	Nominal bit rate 80 Tq (1) Data bit rate 20 Tq (1)	Nominal bit rate 40 Tq (1) Data bit rate 10 Tq (1)

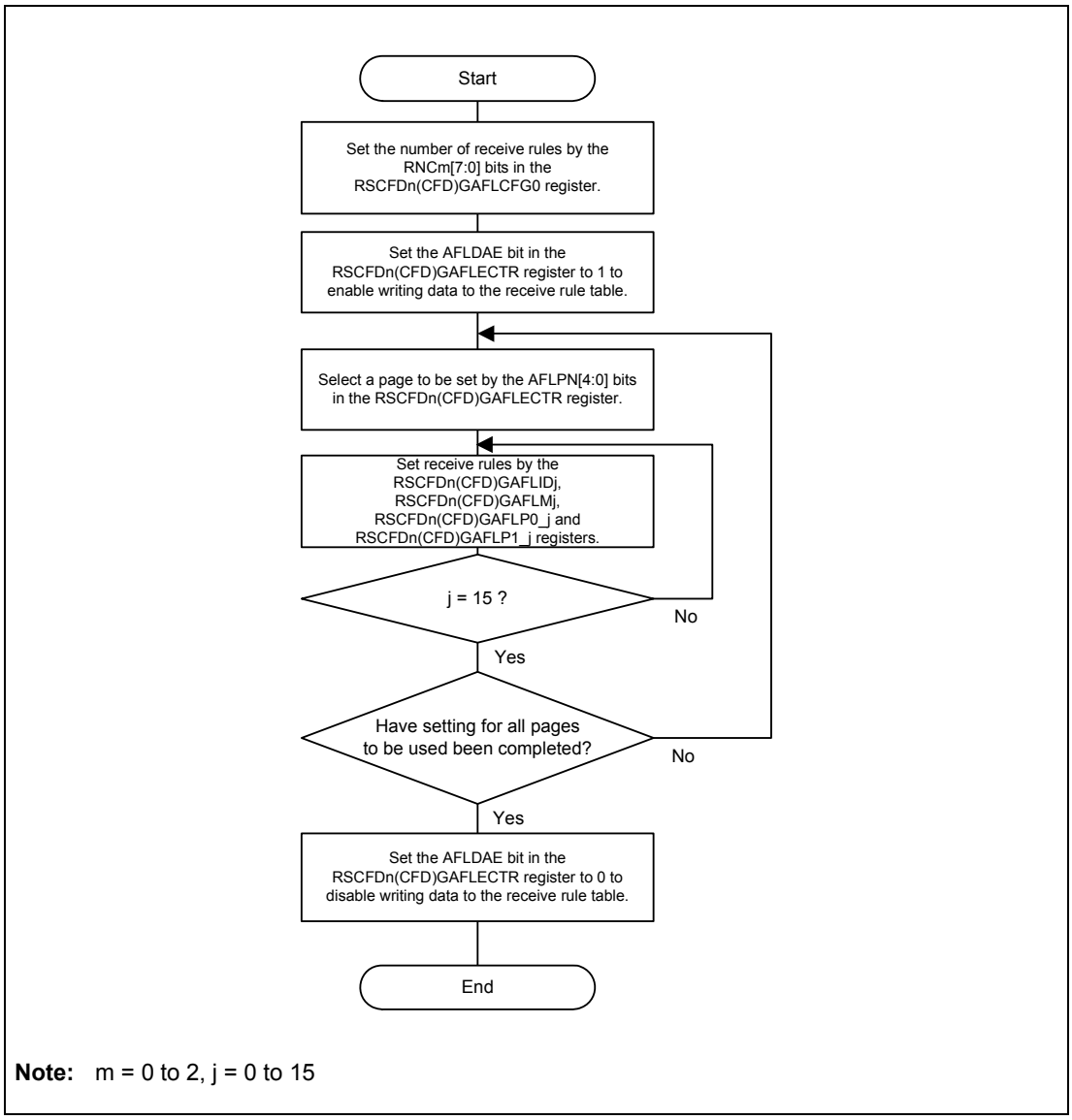
Note: Values in ( ) are nominal bit rate prescaler / data bit rate prescaler division values.

23.11.1.4 Receive Rule Setting

Receive rules can be set using receive rule-related registers.

Up to 16 receive rules can be registered per page. Specify pages 0 to 11 (for 3-channel unit) by the AFLPN[4:0] bits in the RSCFDn(CFD)GAFLECTR register. Set receive rule table write enable/disable using the AFLDAE bit.

Figure 23.19 shows the receive rule setting procedure.



Note: m = 0 to 2, j = 0 to 15

Figure 23.19 Receive Rule Setting Procedure

23.11.1.5 Buffer Setting

Set the number of buffers to be used (number of messages to be stored) and interrupt sources. Also set the payload storage size for CAN FD mode. For transmit/receive FIFO buffers that are set to transmit mode, set transmit buffers to be linked.

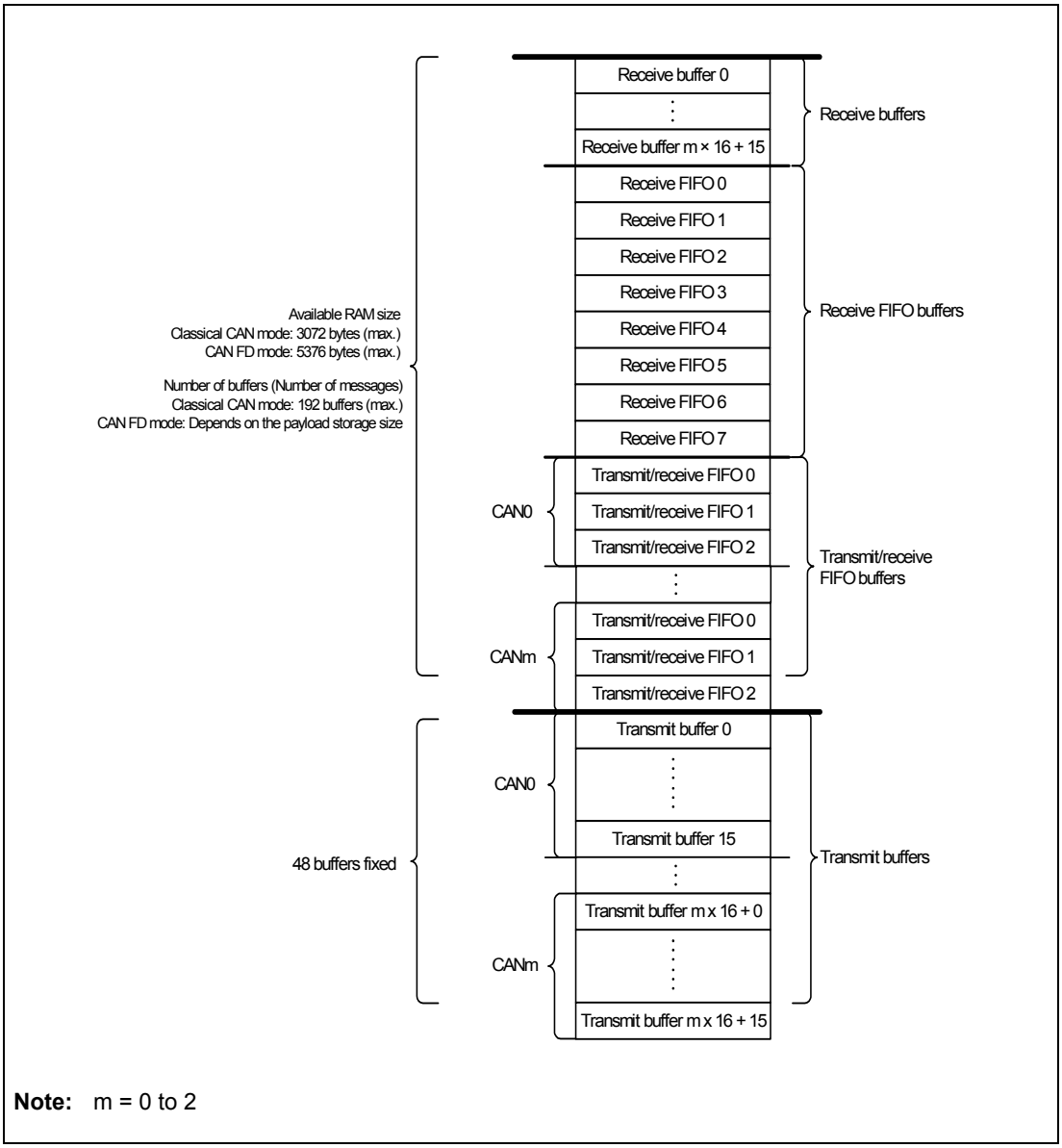
In classical CAN mode, up to 3072 bytes of the RAM can be used in receive buffers and FIFO buffers. Up to 192 buffers are available, and 16 bytes are used per buffer. Configure the buffers so that the following conditions are met.

Number of receive buffers + total number of depth of receive FIFO buffers x +
total number of depth of transmit/receive FIFO buffers k ≤ 192 buffers

In CAN FD mode, up to 5376 bytes of the RAM can be used in receive buffers and FIFO buffers. Configure the number of buffers so that the following conditions are met.

Number of receive buffers × (12 + payload storage size) + total of (number of depth × (12 + payload storage size)) of receive FIFO buffers x + total of (number of depth × (12 + payload storage size)) of transmit/receive FIFO buffers k ≤ 5376 bytes

Figure 23.20 shows the buffer configuration. **Figure 23.21** shows the buffer setting procedure.

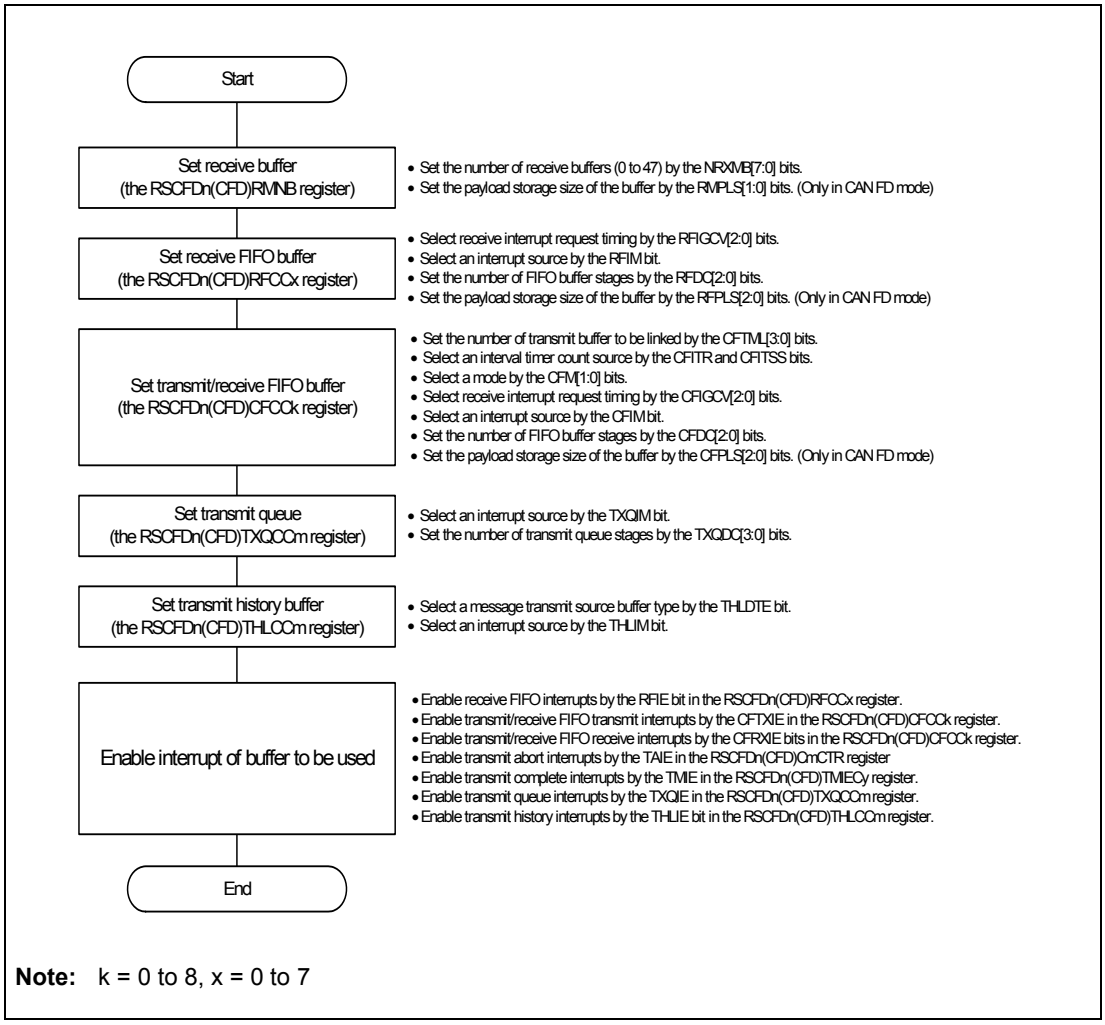


Note: m = 0 to 2

Figure 23.20 Buffer Configuration

CAUTION

Receive buffers, receive FIFO buffers, transmit/receive FIFO buffers, and transmit buffers are located in succession.



Note: k = 0 to 8, x = 0 to 7

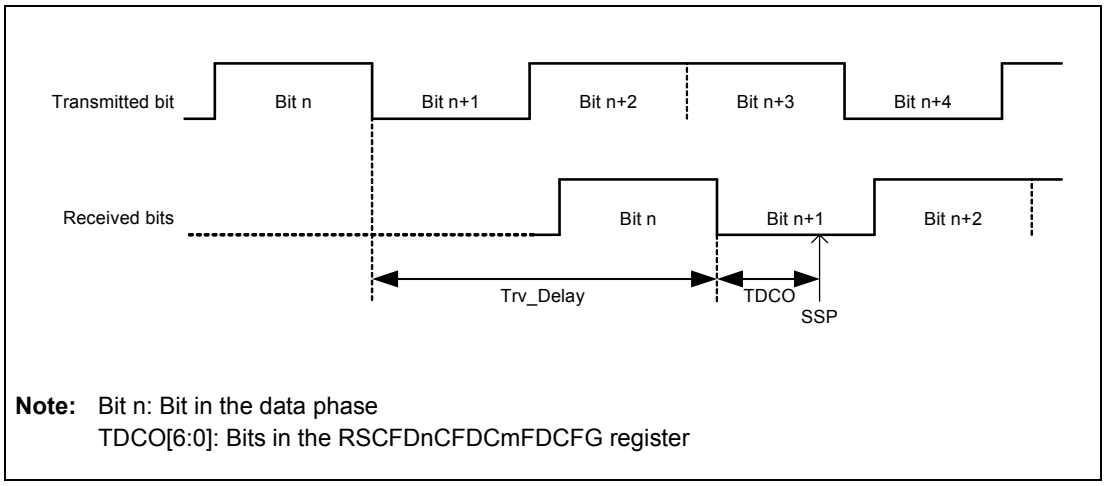
Figure 23.21 Buffer Setting Procedure

23.11.1.6 Transceiver delay compensation (Only in CAN FD Mode)

A high baud rate is used in CAN FD mode. Transmitter delay compensation is provided as a function to accept propagation delay in this case.

To use this function, set the TDCE bit in the RSCFDnCFDCmFDCFG register to 1. Also set the secondary sample point (SSP) timing used in the data phase by the TDCOC bit and TDCO[6:0] bits in the RSCFDnCFDCmFDCFG register.

When the TDCOC bit is 0, the SSP timing equals the total value of the delay measured by the RS-CANFD module and the TDCO[6:0] value. (This value is rounded off to the nearest integer of Tq.) Usually, the TDCO[6:0] value must be equal to SS + TSEG1, the sample point timing.



Note: Bit n: Bit in the data phase
TDCO[6:0]: Bits in the RSCFDnCFDCmFDCFG register

Figure 23.22 SSP timing

When the TDCOC bit is 1, the SSP timing is determined only by the TDCO[6:0] value. (When the DBRP[7:0] value in the RSCFDnCFDCmDCFG register is larger than 0, the TDCO[6:0] value is also rounded off to the nearest integer of Tq.)

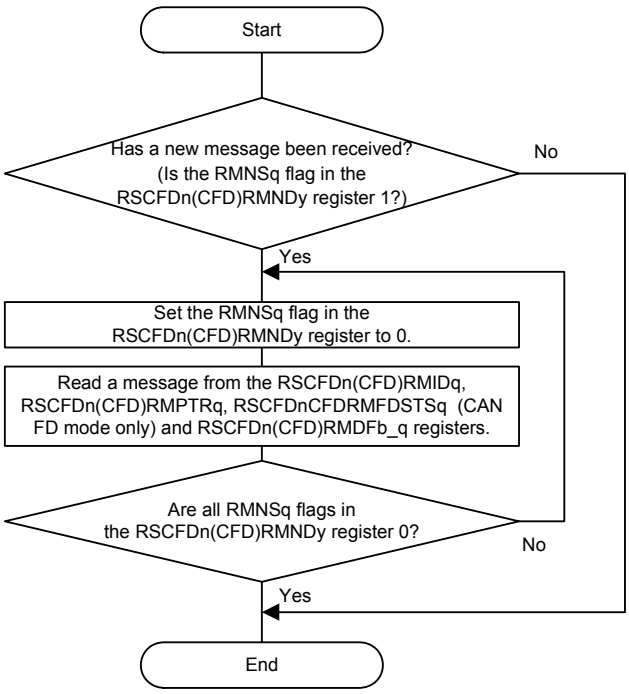
The RS-CANFD module compensates a delay up to (3 CANm bit time - 2 fCAN). (CANm bit time is data bit rate values.)

When the TDCE bit is set to 1(Transmitter delay compensation is enabled) in the RSCFDnCFDCmFDCFG register, set the equal value of 1 or less to the bits NBRP[9:0] and DBRP[7:0].

23.11.2 Reception Procedure

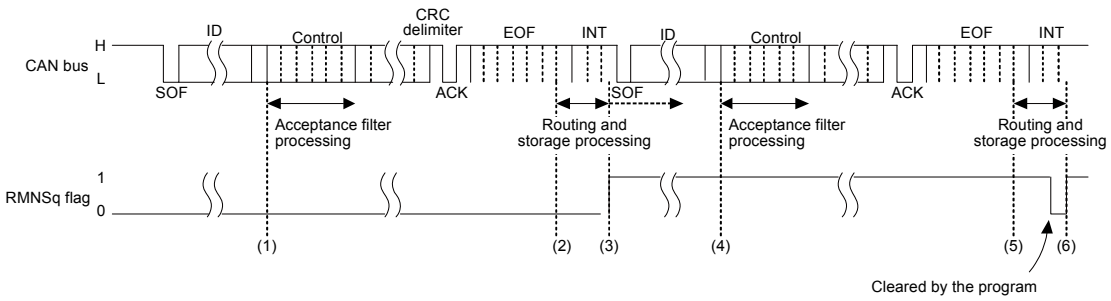
23.11.2.1 Receive Buffer Reading Procedure

When the processing to store received messages in a receive buffer starts, the RMNSq flag in the RSCFDn(CFD)RMNDy register (y = 0, 1, q = 0 to 47) is set to 1 (receive buffer q contains a new message). Messages can be read from registers RSCFDn(CFD)RMIDq, RSCFDn(CFD)RMPTRq, RSCFDnCFDRMFDSTSq (only in CAN FD mode), and RSCFDn(CFD)RMDFb_q (b = 0 or 1 in classical CAN mode, b = 0 to 4 in CAN FD mode). If the next message is received before reading the message out of buffer, the message will be overwritten. **Figure 23.23** shows the receive buffer reading procedure. This procedure ensures the consistency of messages read from registers RSCFDn(CFD)RMIDq, RSCFDn(CFD)RMPTRq, RSCFDnCFDRMFDSTSq, and RSCFDn(CFD)RMDFb_q.



Note: y = 0, 1, q = 0 to 47, b = 0 or 1 (classical CAN mode), 0 to 4 (CAN FD mode)

Figure 23.23 Receive Buffer Reading Procedure



Note: y = 0, 1, q = 0 to 47
RMNSq: Flag in the RSCFDn(CFD)RMNDy register

Figure 23.24 Receive Buffer Reception Timing Chart

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing, the processing to store the message in the specified receive buffer starts.
When the message storage processing starts, the RMNSq flag in the corresponding

RSCFDn(CFD)RMNDy register is set to 1 (the receive buffer contains a new message). If other channels are performing filter processing or transmit priority determination processing, the routing processing and the storage processing may be delayed.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(6) When the corresponding RMNSq flag is cleared to 0 (the receive buffer contains no new message), this flag is set to 1 again when the message storage processing starts. Even if the RMNSq flag remains 1, a new message is overwritten to the receive buffer. The RMNSq flag should not be cleared to 0 during storage of messages.

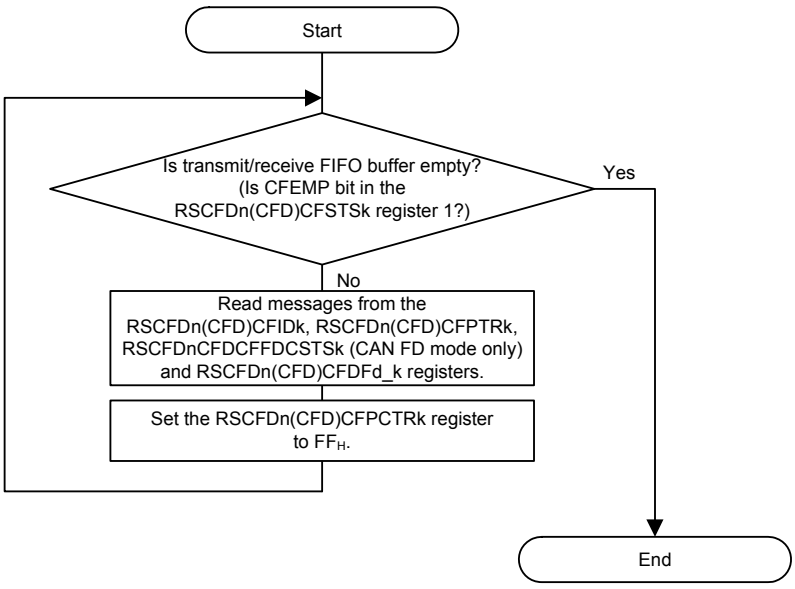
23.11.2.2 FIFO Buffer Reading Procedure

When received messages have been stored in one or more receive FIFO buffers or a transmit/receive FIFO buffer that is set to receive mode or gateway mode, the corresponding message count display counter (RFMC[7:0] bits in the RSCFDn(CFD)RFSTSx register (x = 0 to 7) or CFMC[7:0] bits in the RSCFDn(CFD)CFSTSk register (k = 0 to 8)) is incremented by 1. At this time, when the RFIE bit (receive FIFO interrupt is enabled) in the RSCFDn(CFD)RFCCx register or the CFRXIE bit (transmit/receive FIFO receive interrupt is enabled) in the RSCFDn(CFD)CFCCk register is set to 1, an interrupt request is generated. Received messages can be read from the RSCFDn(CFD)RFIDx, RSCFDn(CFD)RFPTRx, RSCFDnCFDRFFDSTSx (only in CAN FD mode), and RSCFDn(CFD)RFDFd_x (d = 0 or 1 in classical CAN mode, d = 0 to 15 in CAN FD mode) registers for receive FIFO buffers, or from the RSCFDn(CFD)CFIDk, RSCFDn(CFD)CFPTRk, RSCFDnCFDCFFDCSTSk (only in CAN FD mode), and RSCFDn(CFD)CFDFd_k registers for transmit/receive FIFO buffers. Messages in FIFO buffers can be read sequentially on a first-in, first-out basis.

When the message count display counter value matches the FIFO buffer depth (a value set by the RFDC[2:0] bits in the RSCFDn(CFD)RFCCx register or the CFDC[2:0] bits in the RSCFDn(CFD)CFCCk register), the RFFLL or CFFLL flag is set to 1 (the receive FIFO buffer is full).

When all messages have been read out of the FIFO buffer, the RFEMP flag in the RSCFDn(CFD)RFSTSx register or the CFEMP flag in the RSCFDn(CFD)CFSTSk register is set to 1 (the receive FIFO buffer contains no unread message (buffer empty)).

If the RFE bit or the CFE bit is cleared to 0 (no receive FIFO buffer is used) with the interrupt request flag (RFIF flag in the RSCFDn(CFD)RFSTSx register or CFRXIF flag in the RSCFDn(CFD)CFSTSk register) set to 1 (a receive FIFO interrupt request is present), the interrupt request flag is not automatically cleared to 0. The program must clear the interrupt request flag to 0.



Note: k = 0 to 8, d = 0 or 1 (classical CAN mode), 0 to 15 (CAN FD mode)

Figure 23.25 Transmit/Receive FIFO Buffer Reading Procedure

When reading a message in CAN FD mode, do not read the RSCFDnCFDRFDFd_x or RSCFDnCFDCFDFd_k register corresponding to the area exceeding the payload storage size specified by the RFPLS[2:0] bits in the RSCFDnCFDRFCCx register or the CFPLS[2:0] bits in the RSCFDnCFDCFCCk register.

Table 23.183 Payload Storage Area of Receive FIFO Buffer

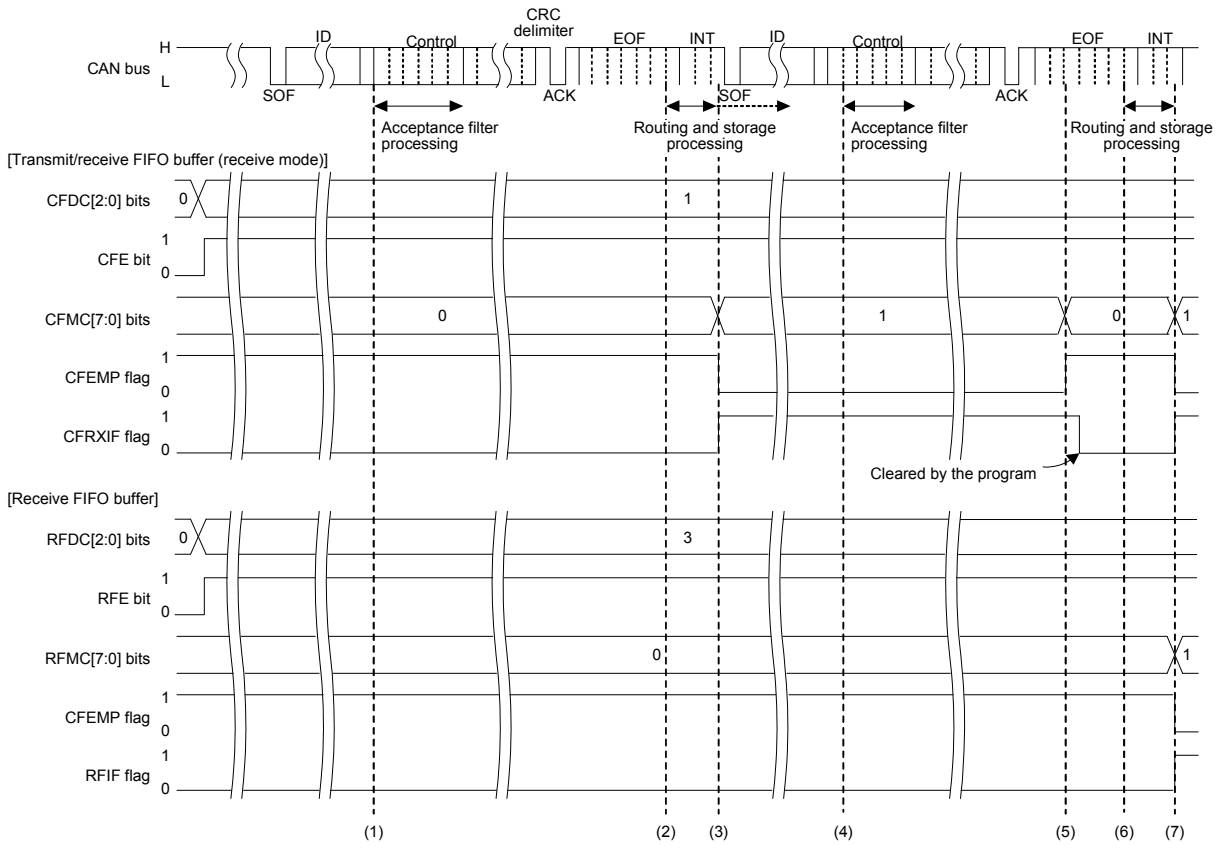
Set RFPLS[2:0] Value	Payload Storage Size	Corresponding Data Field Registers
000_B	8 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF1_x
001_B	12 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF2_x
010_B	16 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF3_x
011_B	20 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF4_x
100_B	24 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF5_x
101_B	32 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF7_x
110_B	48 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF11_x
111_B	64 bytes	RSCFDnCFDRFDF0_x to RSCFDnCFDRFDF15_x

Table 23.184 Payload Storage Area of Transmit/Receive FIFO Buffer (1/2)

Set CFPLS[2:0] Value	Payload Storage Size	Corresponding Data Field Registers
000_B	8 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF1_k
001_B	12 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF2_k
010_B	16 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF3_k
011_B	20 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF4_k
100_B	24 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF5_k

Table 23.184 Payload Storage Area of Transmit/Receive FIFO Buffer (2/2)

Set CFPLS[2:0] Value	Payload Storage Size	Corresponding Data Field Registers
101_B	32 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF7_k
110_B	48 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF11_k
111_B	64 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF15_k



CFDC[2:0], CFE: Bits in the RSCFDn(CFD)CFCCk register
CFMC[7:0], CFEMP, CFRXIF: Flags in the RSCFDn(CFD)CFSTSk register
RFDC[2:0], RFE: Bits in the RSCFDn(CFD)RFCCx register
RFMC[7:0], RFEMP, RFIF: Flags in the RSCFDn(CFD)RFSTSx register

Note: k = 0 to 8, x = 0 to 7

Figure 23.26 FIFO Buffer Reception Timing Chart

(1) When the ID field in a message has been received, the acceptance filter processing starts.

(2) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

(3) When the message has passed through the DLC filter processing and the CFE bit in the RSCFDn(CFD)CFCCk register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCFDn(CFD)CFCCk register is 001_B or more, the message is stored in the transmit/receive FIFO buffer that is set to receive mode. The CFMC[7:0] value in the

RSCFDn(CFD)CFSTSk register is incremented and becomes 01_H. When the CFIM bit in the RSCFDn(CFD)CFCCk register is set to 1 (a FIFO receive interrupt request is generated each time a message has been received), the CFRXIF flag in the RSCFDn(CFD)CFSTSk register is set to 1 (a transmit/receive FIFO receive interrupt request is present). The CFRXIF flag can be reset to 0 by the program.

(4) When the ID field of the next message has been received, the acceptance filter processing starts.

(5) Read received messages from the RSCFDn(CFD)CFIDk, RSCFDn(CFD)CFPTRk, and RSCFDn(CFD)CFDFd_k registers and write FF_H to the RSCFDn(CFD)CFPCTRk register. This causes the CFMC[7:0] bits in the RSCFDn(CFD)CFSTSk register to be decremented. When CFMC[7:0] becomes 00_H, the CFEMP flag in the RSCFDn(CFD)CFSTSk register becomes 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

(6) When the message matches the receive rule of the corresponding channel and the message has been successfully received, the routing processing to transfer the message to the specified buffer starts. When the DCE bit in the RSCFDn(CFD)GCFG register is set to 1 (DLC check is enabled), the DLC filter processing starts at this time.

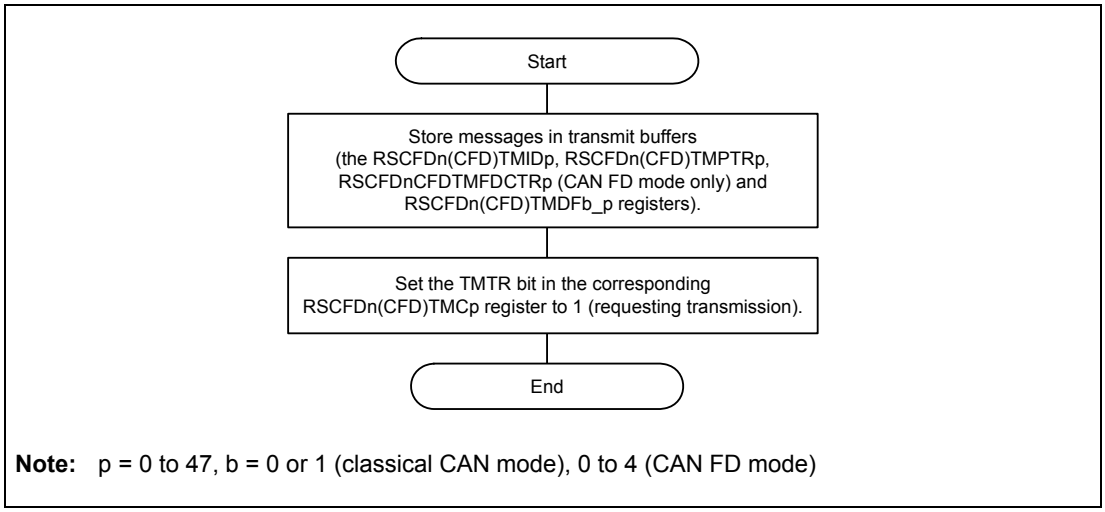
(7) The message is stored in the transmit/receive FIFO buffer set in receive mode when the message has passed through the DLC filter process if the CFE bit is set to 1 (transmit/receive FIFO buffers are used), and the CFDC[2:0] bits are set to 001_B or more. The CFMC[7:0] bit value is incremented by 1 to be 01_H. When the CFIM bit is set to 1 (an interrupt occurs each time a message has been received), the CFRXIF flag is set to 1 (a transmit/receive FIFO receive interrupt request is present).
The message is stored in the receive FIFO buffer if the RFE bit in the RSCFDn(CFD)RFCCx register is set to 1 (receive FIFO buffers are used), and the RFDC[2:0] bits in the RSCFDn(CFD)RFCCx register are set to 001_B or more. The RFMC[7:0] bits in the RSCFDn(CFD)RFSTSx register are set to 01_H by being incremented by 1. When the RFIM bit in the RSCFDn(CFD)RFCCx register is set to 1 (an interrupt occurs each time a message has been received), the RFIF flag in the RSCFDn(CFD)RFSTSx register is set to 1 (a receive FIFO interrupt request is present).

23.11.3 Transmission Procedure

23.11.3.1 Procedure for Transmission from Transmit Buffers

Figure 23.27 shows the procedure for transmission from transmit buffers.

Figure 23.28 shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmission has been successfully completed. **Figure 23.29** shows a timing chart where messages are transmitted from two transmit buffers in the same channel and transmit abort has been completed.



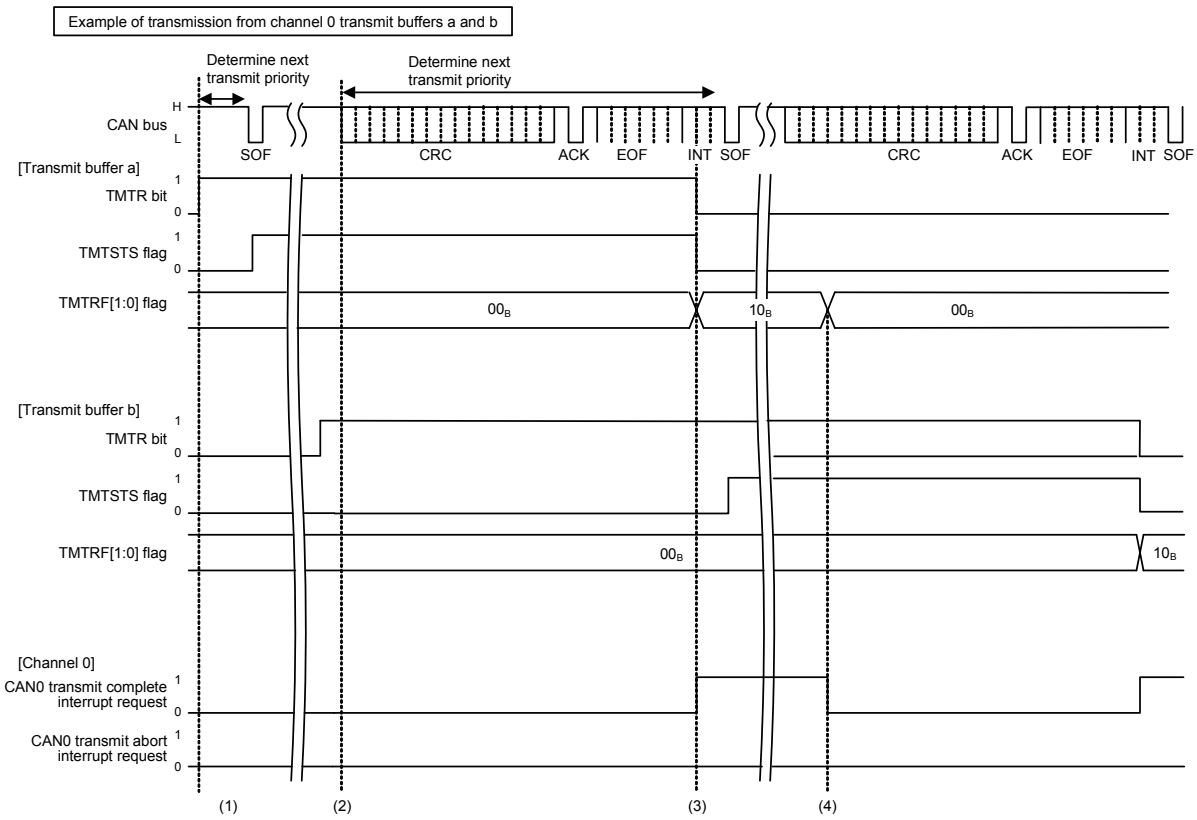
Note: p = 0 to 47, b = 0 or 1 (classical CAN mode), 0 to 4 (CAN FD mode)

Figure 23.27 Procedure for Transmission from Transmit Buffers

In CAN FD mode and transmit buffer merge mode, messages with a payload size of more than 20 bytes can be transmitted from transmit buffers (16 × m) + 0 and transmit buffers (16 × m) + 3. At this time, transmit buffers (16 × m) + 1 to (16 × m) + 2 and transmit buffers (16 × m) + 4 to (16 × m) + 5 are allocated as a payload storage area. Registers RSCFDnCFDTMIDp, RSCFDnCFDTMPTRp, and RSCFDnCFDTMFDCTRp corresponding to these buffers can be used as data field registers that can store 4-byte data bytes (payload) like the RSCFDnCFDTMDFb_p register. **Table 23.185** shows message storage registers when transmitting a message with a payload size of more than 20 bytes from transmit buffer 0.

Table 23.185 Message Storage Registers in Transmit Buffer Merge Mode (Example of Transmit Buffer 0)

Transmit Buffer	Offset from Base Address	Symbol	Register Function in Transmit Buffer Merge Mode
Transmit buffer 0	4000_H	RSCFDnCFDTMID0	Transmit buffer 0 ID data, transmit history data store enable bit, RTR bit,and IDE bit
	4004_H	RSCFDnCFDTMPTR0	Transmit buffer 0 label data and DLC data
	4008_H	RSCFDnCFDTMFDCTR0	Transmit buffer 0 ESI bit, BRS bit, and FDF bit
	$400C_H$ to $401C_H$	RSCFDnCFDTMDF0_0 to RSCFDnCFDTMDF4_0	Transmit buffer 0 data bytes 0, 1, 2, and 3 to transmit buffer 0 data bytes 16, 17, 18, and 19
Transmit buffer 1	4020_H	RSCFDnCFDTMID1	Transmit buffer 0 data bytes 20, 21, 22, and 23
	4024_H	RSCFDnCFDTMPTR1	Transmit buffer 0 data bytes 24, 25, 26, and 27
	4028_H	RSCFDnCFDTMFDCTR1	Transmit buffer 0 data bytes 28, 29, 30, and 31
	$402C_H$ to $403C_H$	RSCFDnCFDTMDF0_1 to RSCFDnCFDTMDF4_1	Transmit buffer 0 data bytes 32, 33, 34, and 35 to transmit buffer 0 data bytes 48, 49, 50, and 51
Transmit buffer 2	4040_H	RSCFDnCFDTMID2	Transmit buffer 0 data bytes 52, 53, 54, and 55
	4044_H	RSCFDnCFDTMPTR2	Transmit buffer 0 data bytes 56, 57, 58, and 59
	4048_H	RSCFDnCFDTMFDCTR2	Transmit buffer 0 data bytes 60, 61, 62, and 63
	$404C_H$ to $405C_H$	RSCFDnCFDTMDF0_2 to RSCFDnCFDTMDF4_2	Not used



Note: a = 0 to 15, b = 0 to 15
TMTR: Bit in the RSCFDn(CFD)TMCp register
TMTSTS, TMTRF[1:0]: Flags in the RSCFDn(CFD)TMSTSp registers

Figure 23.28 Transmit Buffer Transmission Timing Chart (Transmission Completed Successfully)

(1) When the TMTR bit in the RSCFDn(CFD)TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCFDn(CFD)TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.

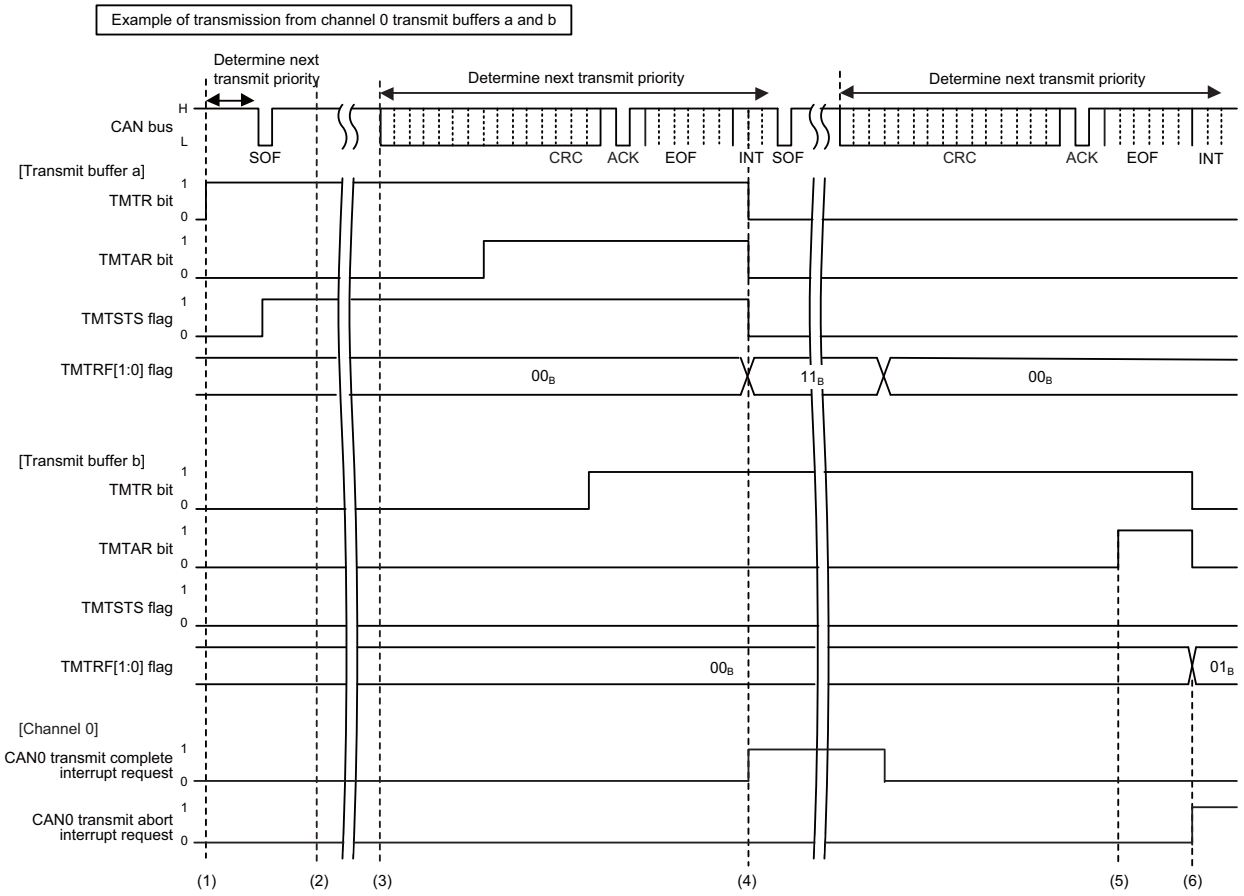
(2) When a transmit request from a buffer is present, the priority determination starts at the first bit of CRC field for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(3) When transmission completes successfully, the TMTRF[1:0] flag in the RSCFDn(CFD)TMSTSa register is set to 10_B (transmission has been completed (without transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCFDn(CFD)TMCa register are cleared to 0. When the TMIEa bit in the RSCFDn(CFD)TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).

(4) Before starting the next transmission, set the TMTRF[1:0] flag to 00_B. Write the next message to the transmit buffer, and then set the TMTR bit to 1 (transmission is requested). The TMTR bit can be set to 1 only when the TMTRF[1:0] flag value is 00_B.

If an arbitration-lost has occurred after transmission is started, the TMTSTS flag is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC field to search the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.

When a 2-bit ECC error is detected during the priority determination processing, no data is transmitted (In classical CAN mode, only when the EEFE bit in the RSCANnGCFG register is 1).



Note: a = 0 to 15, b = 0 to 15
TMTR, TMTAR: Bits in the RSCFDn(CFD)TMCp register
TMTSTS, TMTRF[1:0]: Flags in the RSCFDn(CFD)TMSTSp register

Figure 23.29 Transmit Buffer Transmission Timing Chart (Transmit Abort Completed)

(1) When the TMTR bit in the RSCFDn(CFD)TMCa register is set to 1 while the CAN bus is idle, the transmit priority determination processing starts to determine the highest-priority transmit buffer. If transmit buffer a is determined to be the highest-priority transmit buffer, the TMTSTS flag in the corresponding RSCFDn(CFD)TMSTSa register is set to 1 (transmission is in progress) and the CAN channel starts transmitting data.

(2) When it is determined that the transmit buffer is used for the next transmission or transmission is in progress, message transmission is not aborted unless an error or arbitration loss occurs even if the TMTAR bit is set to 1 (transmit abort is requested).

(3) The priority determination starts at the first bit of the CRC field for the next transmission. In this timing chart, buffer b is not selected as the next transmit buffer. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(4) When transmission completes successfully, the TMTRF[1:0] flag in the RSCFDn(CFD)TMSTSa register is set to 11_B (transmission has been completed (with transmit abort request)) and the TMTSTS flag and the TMTR bit in the RSCFDn(CFD)TMCa register are cleared to 0. When the TMIEa value in the RSCFDn(CFD)TMIEC0 register is 1 (transmit buffer interrupt is enabled), a CAN0 transmit complete interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B (transmission is in progress or no transmit request is present).

(5) While another CAN node is transmitting data on the CAN bus (TMTSTS flag = 0), if the TMTAR bit is set to 1 while the corresponding channel is determining transmit priority, the TMTR bit cannot be cleared to 0.

(6) After the internal processing time has passed, the transmission is terminated and the TMTRF[1:0] flag is set to 01_B. When the transmit buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not being made, an abort request is immediately accepted and the TMTRF[1:0] flag is set to 01_B. At this time, the TMTR and TMTAR bits are cleared to 0. When transmit abort is completed with the TAIE bit in the RSCFDn(CFD)CmCTR register set to 1 (transmit abort interrupt is enabled), an interrupt request is generated. To clear the interrupt request, set the TMTRF[1:0] flag to 00_B.

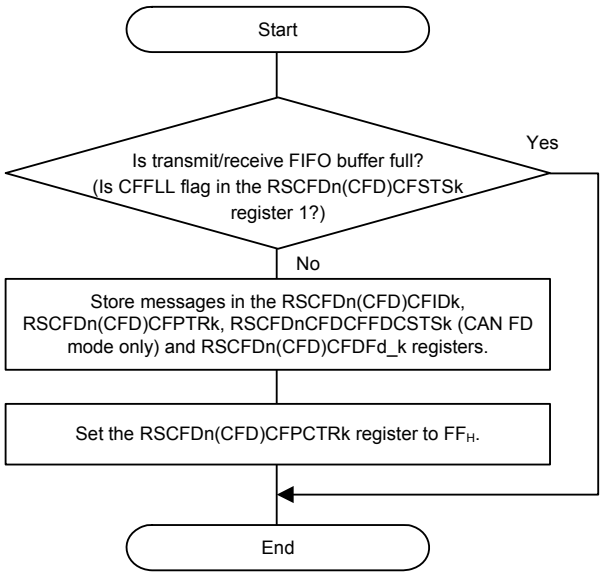
If an arbitration loss has occurred after the CAN channel started transmission, the TMTSTS bit is cleared to 0. The transmit priority determination is reexecuted at the beginning of the CRC field to find the highest-priority transmit buffer. If an error has occurred during transmission or after arbitration loss, the priority determination processing is reexecuted during transmission of an error frame.

When a 2-bit ECC error is detected during the priority determination processing, no data is transmitted (In classical CAN mode, only when the EEFE bit in the RSCANnGCFG register is 1).

23.11.3.2 Procedure for Transmission from Transmit/Receive FIFO Buffers

Figure 23.30 shows the procedure for transmission from transmit/receive FIFO buffers.

Figure 23.31 shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmission has been successfully completed. **Figure 23.32** shows a timing chart where messages are transmitted from two transmit/receive FIFO buffers in the same channel and transmit abort has been completed.



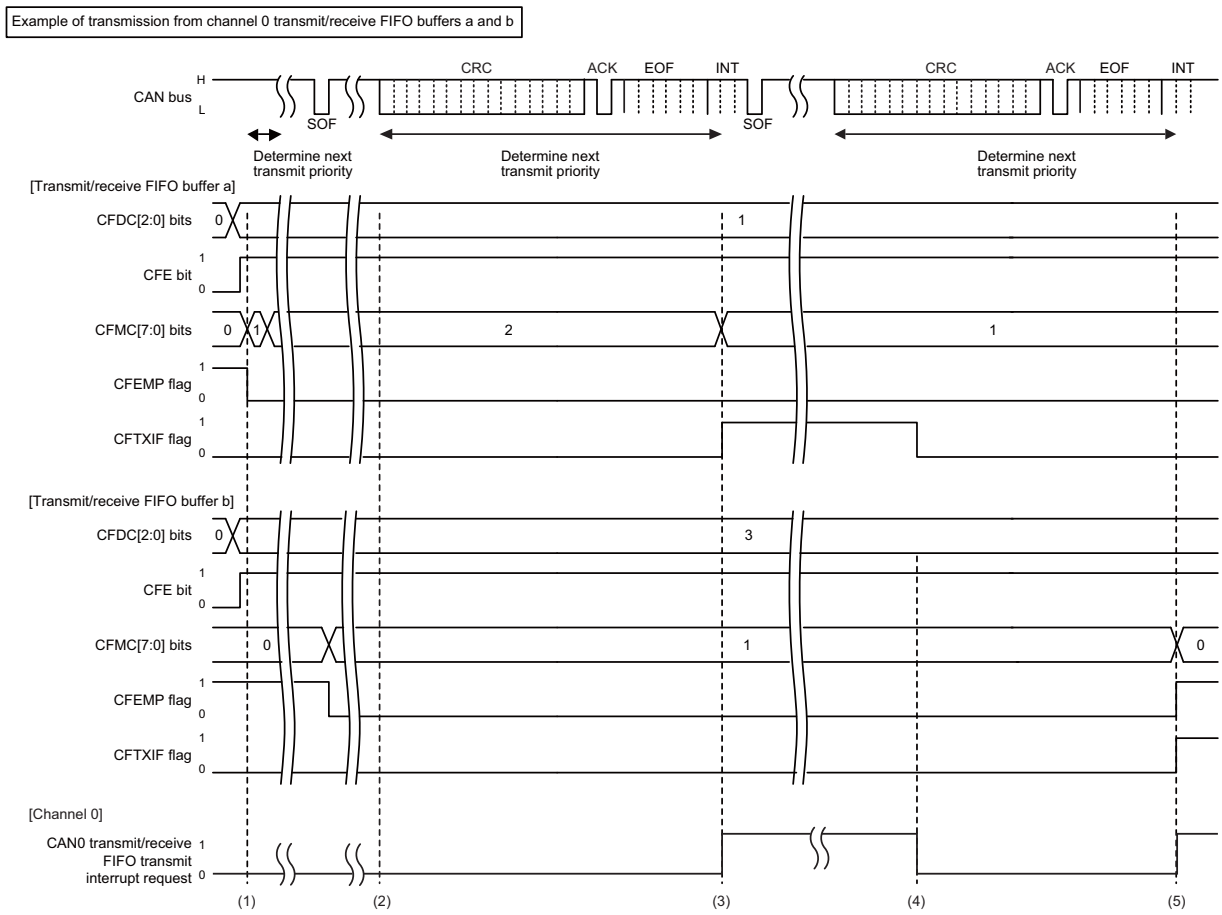
Note: k = 0 to 8, d = 0 or 1 (classical CAN mode), 0 to 15 (CAN FD mode)

Figure 23.30 Procedure for Transmission from Transmit/Receive FIFO Buffers

When storing a message, do not write a value to the RSCFDnCFDCFDFd_k register corresponding to the area exceeding the payload storage size specified by the CFPLS[2:0] bits in the RSCFDnCFDCFCCk register.

Table 23.186 Payload Storage Area of Transmit/Receive FIFO Buffer

Set CFPLS[2: 0] Value	Payload Storage Size	Corresponding Data Field Registers
000_B	8 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF1_k
001_B	12 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF2_k
010_B	16 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF3_k
011_B	20 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF4_k
100_B	24 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF5_k
101_B	32 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF7_k
110_B	48 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF11_k
111_B	64 bytes	RSCFDnCFDCFDF0_k to RSCFDnCFDCFDF15_k



Note: a = 0 to 2, b = 0 to 2
CFDC[2:0],CFE: Bits in the RSCFDn(CFD)CFCCk register
CFMC[7:0], CFEMP, CFTXIF: Flags in the RSCFDn(CFD)CFSTSk register

Figure 23.31 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmission Completed Successfully)

(1) While the CAN bus is idle, when the CFE bit in the RSCFDn(CFD)CFCCa register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCFDn(CFD)CFCCa register is 001_B (4 messages) or more and the CFMC[7:0] value in the RSCFDn(CFD)CFSTSa register is 01_H or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined, transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer a of channel 0.

(2) When a transmit request from a buffer is present, the priority determination starts at the first bit of the CRC field for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

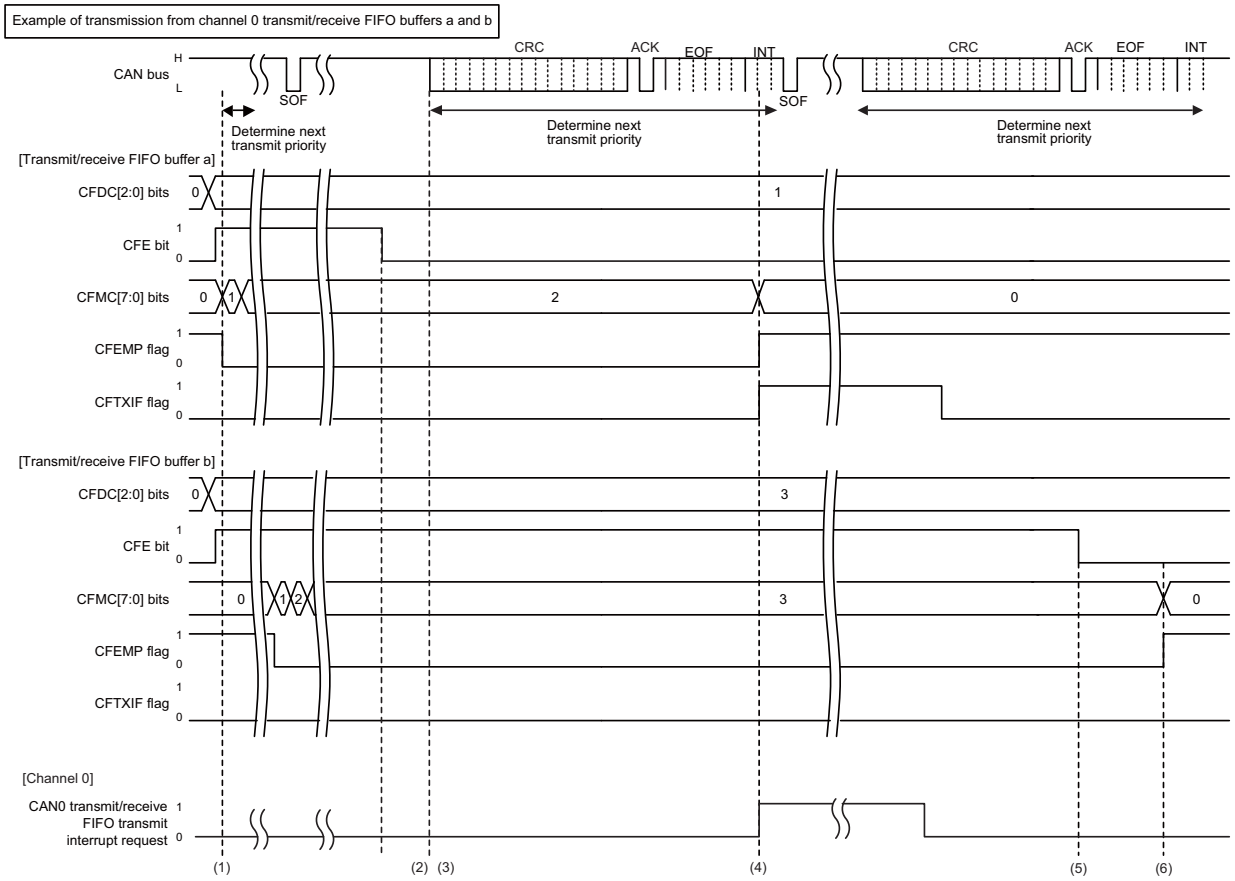
(3) When transmission completes successfully, the CFMC[7:0] value in the RSCFDn(CFD)CFSTSa register is decremented by 1. Setting the CFIM bit in the RSCFDn(CFD)CFCCa register to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the

CFTXIF flag in the RSCFDn(CFD)CFSTSk register to 1 (a transmit/receive FIFO transmit interrupt request is present).

(4) The program can clear the CFTXIF flag.

(5) Message transmission from transmit/receive FIFO buffer b of channel 0 has been completed and the CFMC[7:0] value in the RSCFDn(CFD)CFSTSb register is decremented by 1. The CFMC[7:0] bits are cleared to 00_H and therefore the CFEMP flag in the RSCFDn(CFD)CFSTSk register is set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)).

Transmission is continued until the CFEMP flag is set to 1. It is possible to continuously store transmit messages in FIFO buffers until the CFFLL flag in the RSCFDn(CFD)CFSTSa and RSCFDn(CFD)CFSTSb register is set to 1 (the transmit/receive FIFO buffer is full).



Note: a = 0 to 2, b = 0 to 2
CFDC[2:0], CFE: Bits in the RSCFDn(CFD)CFCCk register
CFMC[7:0], CFEMP, CFTXIF: Flags in the RSCFDn(CFD)CFSTSk register

Figure 23.32 Transmit/Receive FIFO Buffer Transmission Timing Chart (Transmit Abort Completed)

(1) While the CAN bus is idle, when the CFE bit in the RSCFDn(CFD)CFCCa register is 1 (transmit/receive FIFO buffers are used) and the CFDC[2:0] value in the RSCFDn(CFD)CFCCa register is 001_B (4 messages) or more and the CFMC[7:0] value in the RSCFDn(CFD)CFSTSa register is 01_H or more, the priority determination processing starts to determine the highest-priority transmit message. When the highest-priority transmit message has been determined,

transmission of the message starts. In this figure, the message is transmitted from transmit/receive FIFO buffer a of channel 0.

(2) When transmission is in progress or it is determined that the transmit/receive FIFO buffer is used for the next transmission, message transmission is not aborted unless an error or arbitration loss occurs even if the CFE bit is set to 0 (no transmit/receive FIFO buffer is used).

(3) When a transmit request from a buffer is present, the priority determination starts at the first bit of the CRC field for the next transmission. In this figure, transmit/receive FIFO buffer b is not selected as a buffer for the next transmission. The determination time may delay if the transmit priority determination processing is performed on another channel. However, the delay does not occur during transmission because the determination processing is completed by the third bit of the intermission.

(4) When transmit completes successfully, the CFMC[7:0] value is cleared to 00_H. Setting the CFIM bit to 1 (a FIFO transmit interrupt request is generated each time a message has been transmitted) sets the CFTXIF flag in the RSCFDn(CFD)CFSTSa register to 1 (a transmit/receive FIFO transmit interrupt request is present). The program can clear the CFTXIF flag.

(5) If another CAN node on the CAN bus is transmitting data (not from transmit/receive FIFO buffer b), transmit/receive FIFO buffers cannot be disabled immediately even if the CFE bit in the RSCFDn(CFD)CFCCb register is cleared to 0 (no transmit/receive FIFO buffer is used) during transmit priority determination. (The CFEMP flag in the RSCFDn(CFD)CFSTSb register is not set to 1 (the transmit/receive FIFO buffer contains no message (buffer empty)) immediately.)

(6) After the internal processing time has passed, transmit/receive FIFO buffers are disabled and the CFMC[7:0] bits in the RSCFDn(CFD)CFSTSb register are cleared to 00_H and the CFEMP flag is set to 1. When the transmit/receive FIFO buffer is not transmitting data and is not selected as the next transmit buffer and priority determination is not in progress, the transmit/receive FIFO buffer is immediately disabled. (The CFMC[7:0] bits are cleared to 00_H and the CFEMP flag is set to 1.)

23.11.3.3 Procedure for Transmission from the Transmit Queue

Figure 23.33 shows the procedure for transmission from the transmit queue.

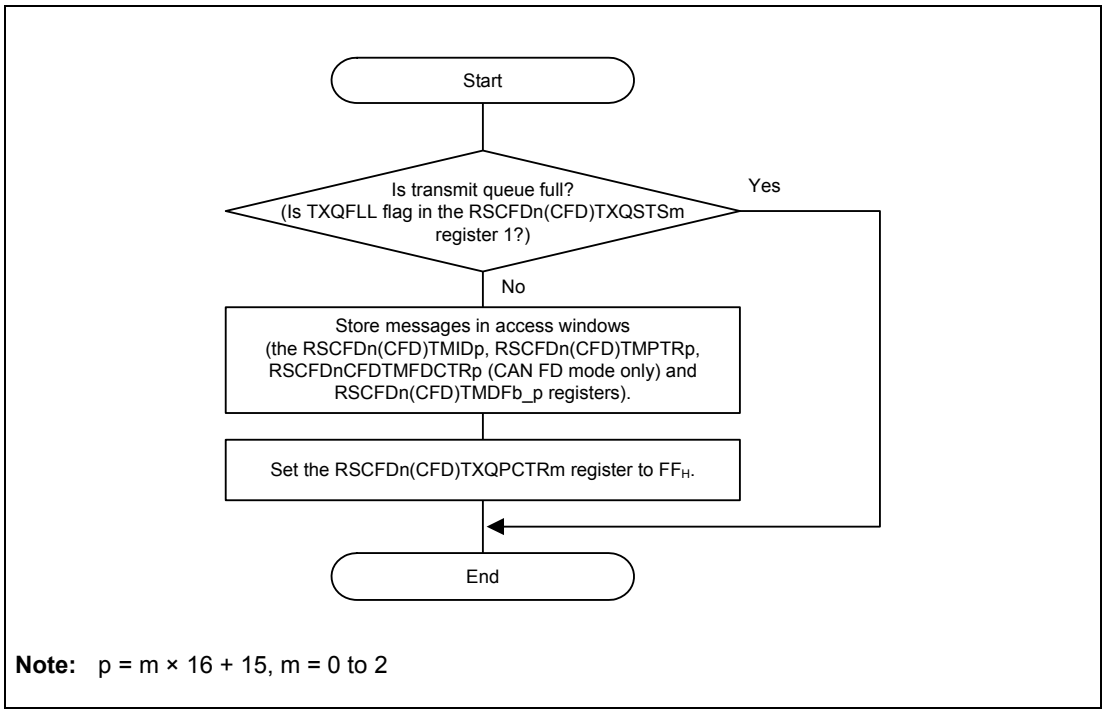


Figure 23.33 Procedure for Transmission from the Transmit Queue

23.11.3.4 Transmit History Buffer Reading Procedure

Transmit history data can be read from the RSCFDn(CFD)THLACCm register. The next data can be accessed by writing FF_H to the corresponding RSCFDn(CFD)THLPCTRm register (m = 0 to 2) after reading a set of data. **Figure 23.34** shows the transmit history buffer reading procedure.

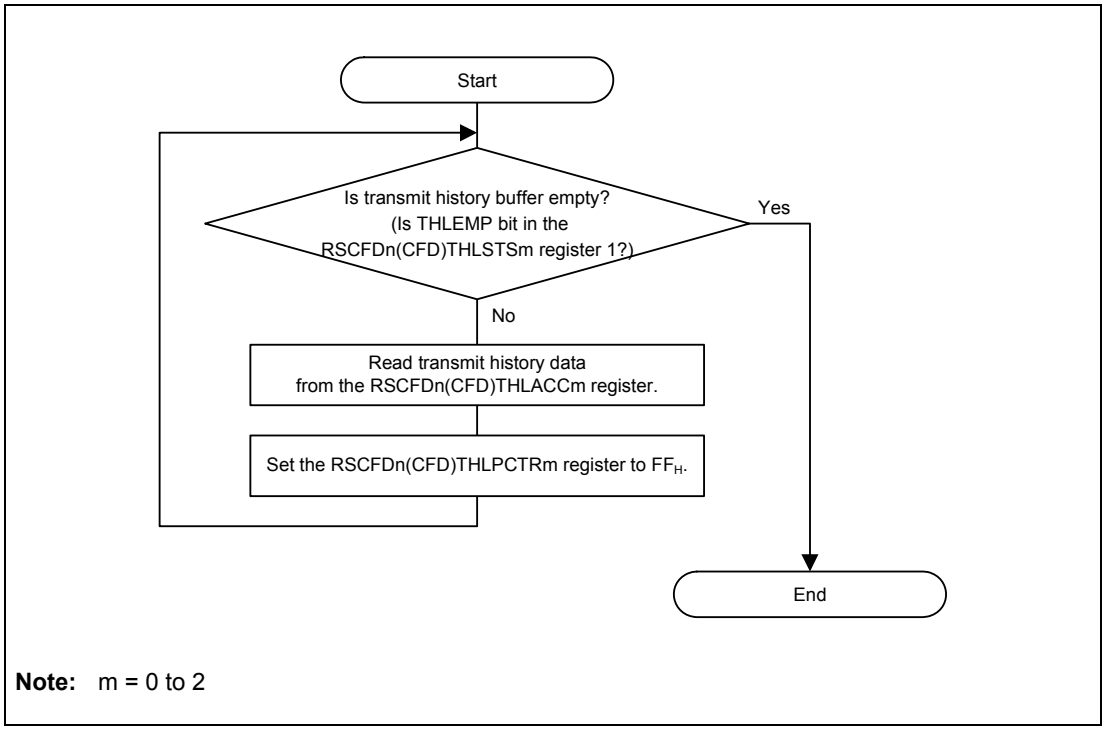


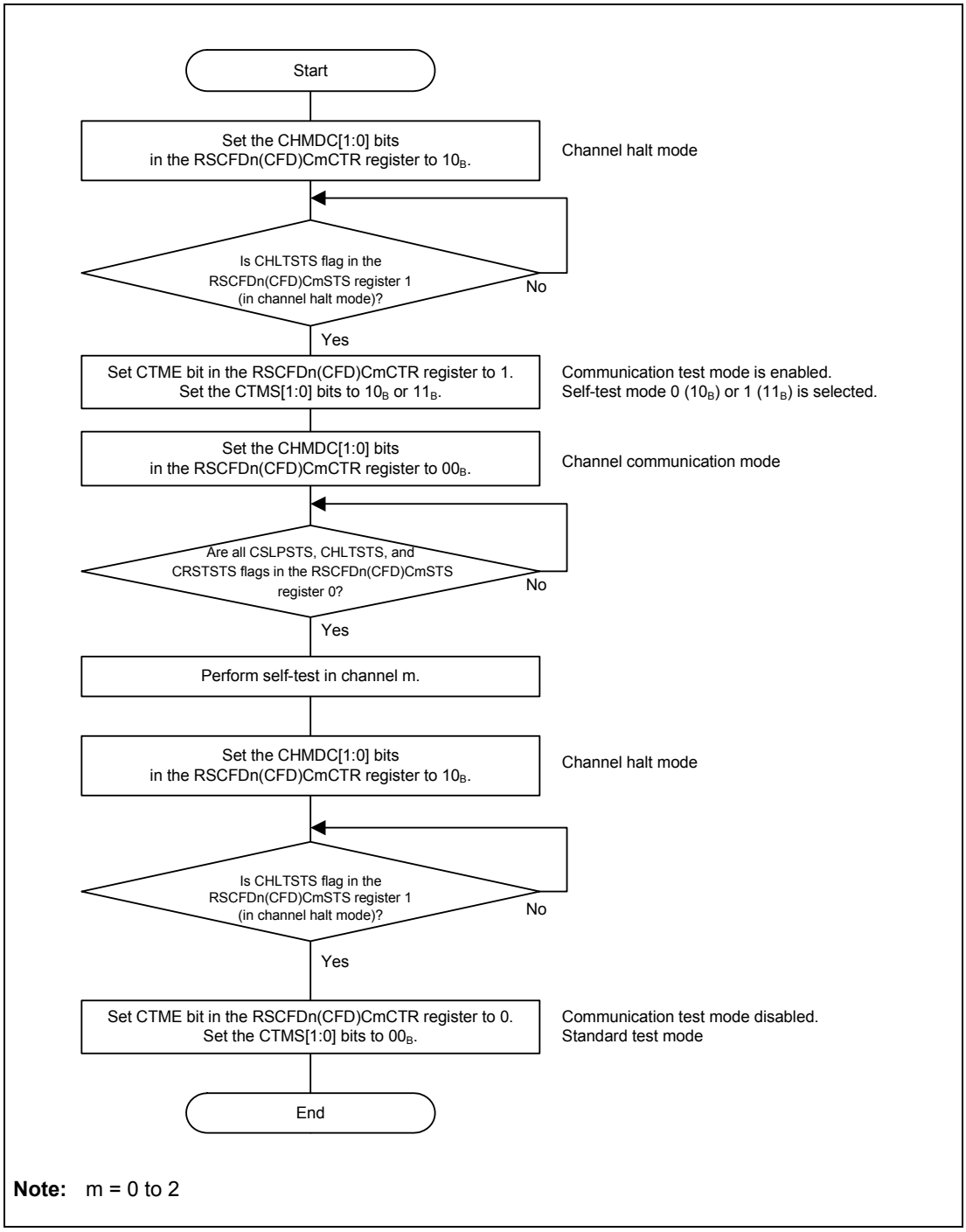
Figure 23.34 Transmit History Buffer Reading Procedure

23.11.4 Test Settings

23.11.4.1 Self-Test Mode Setting Procedure

Self-test mode allows communication test on a channel basis by enabling a CAN node to receive its own transmitted messages.

Figure 23.35 shows the self-test mode setting procedure.



Note: m = 0 to 2

Figure 23.35 Self-Test Mode Setting Procedure

23.11.4.2 Procedure for Releasing the Protection

Since the global test function in **Table 23.187** is protected, write the protection release data 1 and release data 2 in succession to the LOCK[15:0] bits in the RSCFDn(CFD)GLOCKK register, then set the target test bit to 1.

Table 23.187 Protection Release Data for Test Function

Test Function	Protection Release Data 1	Protection Release Data 2	Target Bit
RAM test	7575_H	$8A8A_H$	RTME bit in the RSCFDn(CFD)GTSTCTR register

If an incorrect value is written to the LOCK[15:0] bits, restart from writing the protection release data 1. **Figure 23.36** shows the procedure for releasing the protection.

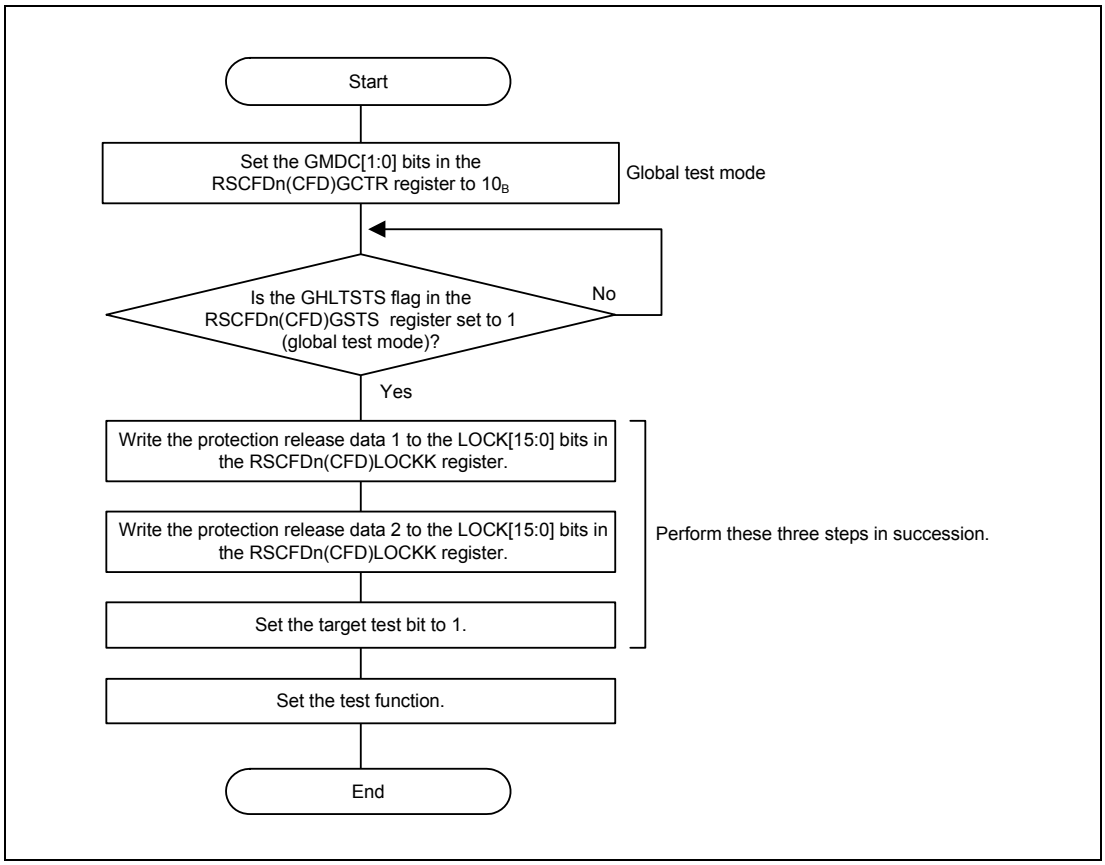
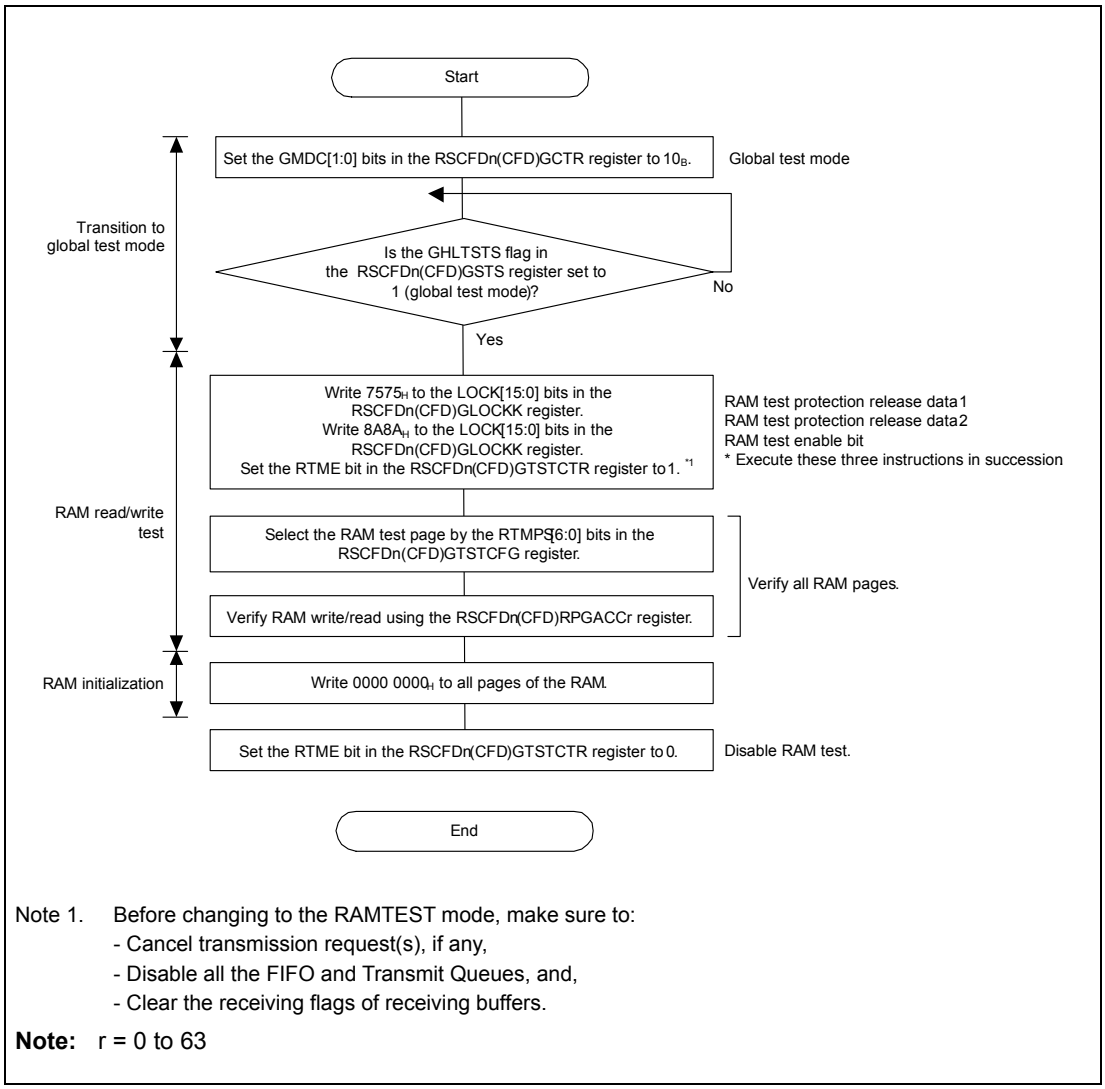


Figure 23.36 Protection Release Procedure

23.11.4.3 RAM Test Setting Procedure

RAM tests include CAN RAM read/write test. The read/write test verifies that data written to the RAM is read correctly. Before closing the RAM test, write 0000 0000_H to all pages of the CAN RAM.

Figure 23.37 shows the RAM test setting procedure.



Note 1. Before changing to the RAMTEST mode, make sure to:
- Cancel transmission request(s), if any,
- Disable all the FIFO and Transmit Queues, and,
- Clear the receiving flags of receiving buffers.

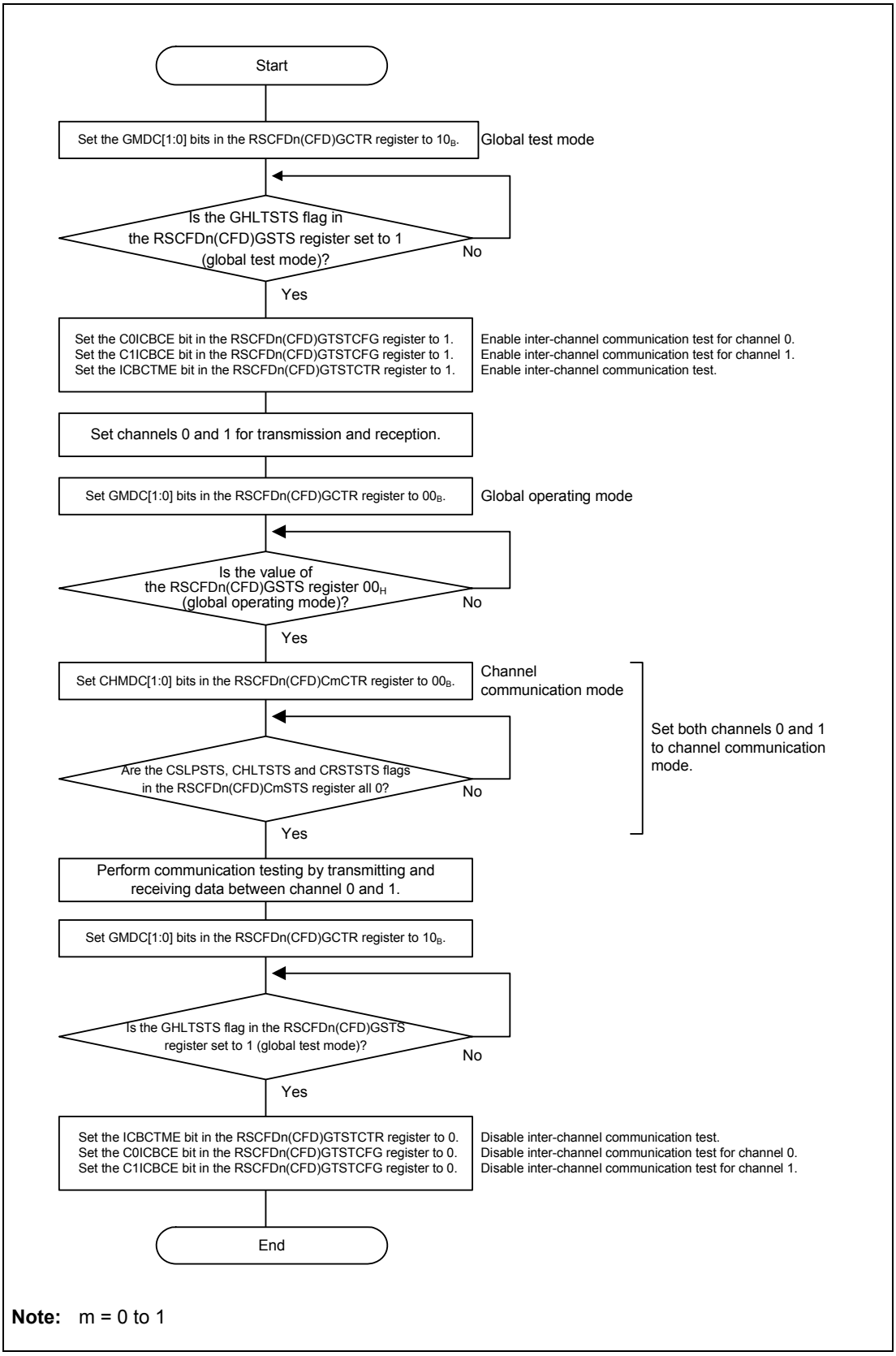
Note: r = 0 to 63

Figure 23.37 RAM Test Setting Procedure

23.11.4.4 Inter-Channel Communication Test Setting Procedure

Communication testing can be performed by transmitting and receiving data between different channels.

Figure 23.38 shows the inter-channel communication test setting procedure.



Note: m = 0 to 1

Figure 23.38 Inter-Channel Communication Test Setting Procedure (Example of Communication Test between Channel 0 and Channel 1)

23.12 Detection and Correction of Errors in RS-CANFD RAM

This section shows RS-CANFD RAM in CAN-FD mode. For RS-CANFD RAM in Classical CAN mode, see Section 22.11, Detection and Correction of Errors in RSCAN RAM.

23.12.1 ECC for the RSCFDn RAM

Table 23.188 gives an outline of the ECC functions for the RSCFDn RAM.

Table 23.188 List of the ECC Functions for the RSCFDn RAM

Item	Outline of Functions
ECC error detection/correction	The RAM is checked for ECC errors. The following options are selectable. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection The ECC error detection/correction can be disabled by using through mode. With the initial settings, error detection/correction is enabled.
Error notification	When an ECC 2-bit error is generated, the error is notified. • Error notification can be enabled or disabled when an ECC 2-bit error is detected. In the initial setting, 2-bit error notification is enabled. However, when the interrupt is masked by the FEINTFMSK register, interrupt processing is not performed.
Error status	Monitoring for the detection of 2-bit ECC errors and for the detection of 1-bit ECC errors is available. A register for clearing the error status is provided.
Address capture	• Only one address at which an ECC error has occurred can be captured. • A signal is generated upon detection of a 2-bit or 1-bit ECC error, and the signal is used as a trigger to capture the error-causing address (when the first (1-bit or 2-bit) error is detected after the flag is cleared).

CAUTION

When ECC error detection/correction is performed, confirm initialization of the RSCFDn RAM (RSCFDnGSTS.GRAMINIT = 0) before it is used.

23.12.2 Register Base Address

ECCRCFDn base addresses are listed in the following table.

ECCRCFD register addresses are given as offsets from the base addresses in general.

Table 23.189 Register Base Address

Base Address Name	Base Address
<ECCRCFD0_base>	FFC7 1800_H
<ECCRCFD1_base>	FFC7 $1C00_H$

23.12.3 Interrupt Request

Table 23.190 lists the ECC interrupt request of RSCFDn RAM.

Table 23.190 Interrupt Requests

Unit Interrupt Signal	Outline	Connected to
—	RSCFD ECC 2-bit error interrupt	Error Control Module INTECCDPERIRAM
—	RSCFD ECC 1-bit error interrupt	Error Control Module INTECCSPERIRAM

23.12.4 ECCRCFDnCTL — RSCFDn ECC Control Register

The ECCRCFDnCTL register controls the mode of the ECC and the status for RSCFDn.

Bits 7, 5 and 4 should be set (written) while the RSCFDn operation is stopped.

In addition, when writing to bit 7, EMCA1 and EMCA0 need to be 01_B.

Access: This register can be read/written in 16-bit units.

Address: <ECCRCFDn_base> + 0000_H

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA1	EMCA0	—	—	—	ECER2C	ECER1C	—	ECTHM	ECERVF	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	ECEMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	Undefined
R/W	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R	R/W	R/W	R/W	R/W	R/W	R	R	R

Note 1. These bits are always read as 0.

Table 23.191 ECCRCFDnCTL Register Contents (1/2)

Bit position	Bit Name	Function
15	EMCA1	Access control bits 1 and 0 to ECC mode selection bit These bits specify whether modifying the ECTHM bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 01_B, writing to bit 7 is enabled.
14	EMCA0	
13 to 11	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
10	ECER2C	2-bit ECC error detection flag clear bit This bit is used to clear the 2 bit error detection flag of ECER2F (bit 2). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER2F bit is set to clear the ECER2F bit. When a conflict between this bit writing and ECER2F bit setting occurs, writing to this bit has a priority.
9	ECER1C	1-bit ECC error correction accumulation flag clear bit This bit is used to clear the 1 bit error detection/correction flag of ECER1F (bit 1). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER1F bit is set to clear the ECER1F bit. When a conflict between this bit writing and ECER1F bit setting occurs, writing to this bit has a priority.
8	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
7	ECTHM	ECC function through mode selection bit This bit is used to set enabling and disabling of ECC. Setting this bit to 1 disables ECC function. When writing to this bit, (0, 1) must be written to (EMCA1, EMCA0) at the same time. 0: Through mode is disabled (normal operation mode). 1: Through mode is enabled. (ECC function disable)
6	ECERVF	ECC error judgment enable 0: error judgement disabled 1: error judgement enabled Error judgement is only effective in normal operation mode, i.e. if ECCRCFDnCTL.ECTHM = 0. If error judgement is disabled no interrupts are asserted in case of a single or double bit error detection. However detected single bit errors are corrected. Modification of this bit is only possible, if EMCA[1:0] = 01_B. Otherwise any write to this bit is ignored.

Table 23.191 ECCRCFDnCTL Register Contents (2/2)

Bit position	Bit Name	Function
5	EC1ECP	1-bit error correction enable bit This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected.
4	EC2EDIC	2-bit error detection interrupt control bit This bit controls whether to generate an interrupt when 2-bit error is detected. 0: When 2-bit error is detected, a INTECCDCNRAM interrupt will not be generated. 1: When 2-bit error is detected, a INTECCDCNRAM interrupt will be generated. (initial value)
3	EC1EDIC	1-bit error detection interrupt control bit This bit controls whether to generate an interrupt when 1-bit error is detected. 0: When 1-bit error is detected, a INTECCSPERIRAM interrupt will not be generated. 1: When 1-bit error is detected, a INTECCSPERIRAM interrupt will be generated.
2	ECER2F	2-bit error detection flag bit This flag indicates whether 2-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). When 2-bit error interrupt is enabled (EC2EDIC = 1) and this flag is set, an ECC 2-bit error interrupt (INTECCDCNRAM) is output. Write 1 to the ECER2C bit (bit 10) to clear the flag. This bit is cleared at the time of through mode enable selection (ECTHM = 1). If 2-bit error is detected again while this bit is set, an interrupt signal will not be generated. 0: 2-bit error has not occurred since this bit was cleared. 1: 2-bit error has occurred.
1	ECER1F	1-bit error detection/correction flag bit This flag indicates whether 1-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). Write 1 to the ECER1C bit (bit 9) to clear the flag. This bit is cleared at the time of through mode enable selection (ECTHM = 1). 0: 1-bit error has not occurred since this bit was cleared. 1: 1-bit error has occurred.
0	ECEMF	ECC error message flag This flag indicates whether an error exists in the current read data. This bit is updated whenever the RAM reads data. This bit might be set if it is read before initialization of the RAM. This bit is cleared at the time of through mode enable selection (ECTHM = 1) and when there is no 1 bit error in the decode circuit input data. 0: The currently-read RAM data does not have bit errors. 1: The currently-read RAM data have bit errors.

CAUTION

Bits 2 and 1 should be cleared when the ECC error message flag (ECEMF) is not set.

We recommend initializing the RAM before clearing bits 2 and 1.

23.12.5 ECCRCFDnTMC — RSCFDn ECC Test Mode Control Register

The ECCRCFDnTMC register switches to and controls the test mode.

This register can be used when RSCFD is not accessed to RAM.

Access: This register can be read or written in 16-bit units.

Address: <ECCRCFDn_base> + 0004_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA1	ETMA0	—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. These bits are always read as 0.

Table 23.192 ECCRCFDnTMC Register Contents (1/2)

Bit position	Bit Name	Function
15	ETMA1	Access control bits 1 and 0 to ECC test mode bit These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 10_B, writing to bit 7 is enabled.
14	ETMA0	
13 to 8	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
7	ECTMCE	ECC test mode enable bit This bit specifies whether to enable access to the test registers and test control bits. When writing to this bit, (1, 0) should be written to (ETMA1, ETMA0) at the same time. 0: Access to the test mode registers and bits is disabled. 1: Access to the test mode registers and bits is enabled. Test registers: ECCRCFDnTED, ECCRCFDnTRC, ECCRCFDnSYND, ECCRCFDnHORD, ECCRCFDnECRD, ECCRCFDnERDB Register test control bit: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
4	ECTRRS	ECC RAM read test mode selection bit Select reading destination when reading ECCRCFDnTED register and reading destination when reading ECCRCFDnERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0:The read value of the ECCRCFDnTED register is the write value of the ECCRCFDnTED register. The read value of the ECCRCFDnERDB register is the write value of the ECCRCFDnERDB register. 1: The read value of the ECCRCFDnTED register can read RAM data. The read value of the ECCRCFDnERDB register is the ECC data to be written to RAM.
3	ECREOS	ECC redundant bit output data selection bit This bit is used to identifyand select the ECC data to be stored in RAM as ECC data generated for write data or the value of the ECCRCFDnERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Store ECC data generated for write data to RAM. 1: Store the value of ECCRCFDnERDB register to RAM.

Table 23.192 ECCRCFDnTMC Register Contents (2/2)

Bit position	Bit Name	Function
2	ECENS	ECC encoder input selection bit This bit is used to identify the targeted data for ECC data generation as write data to RAM or the value of the ECCRCFDnTED register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate ECC data from the write data to RAM. 1: Generate ECC data from the value of the ECCRCFDnTED register.
1	ECDCS	ECC decoder input selection bit This bit is used to indetify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the ECCRCFDnTED register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate syndrome code from RAM data and detect errors. 1: Generate syndrome code from the value of the ECCRCFDnTED register and detect errors.
0	ECREIS	ECC redundant bit input data selection bit This bit is used to indetify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the ECCRCFDnERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate syndrome code from ECC data stored in RAM and detect errors. 1: Generate syndrome code from the value of the ECCRCFDnERDB register and detect errors.

23.12.6 ECCRCFDnTED — RSCFDn ECC Encode/Decode Input/Output Replacement Test Register

In ECC test mode, this register handles test data.

The value of the register can be used to generate ECC data or syndrome code.

When ECC test mode is enabled (ECCRCFDnTMC.ECTMCE = 1), it is accesible. When ECCRCFDnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when RSCFD is not accessed to RAM.

Access: This register can be read/written in 32-bit units.

Address: <ECCRCFDn_base> + $000C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.193 ECCRCFDnTED Register Contents

Bit position	Bit Name	Function
31 to 0	ECEDB[31:0]	When ECCRCFDnTMC.ECENS = 1, the value of this register is used to generate ECC data and the value this register to RAM is stored to RAM. When ECCRCFDnTMC.ECDCS = 1, the value of this register is used to generate syndrome code and the value of this register is stored in ECC decode syndrome data register (ECCRCFDnSYND). In addition, when ECCRCFDnTMC.ECTRRS = 1, RAM data [31:0], instead of written data, is read for the value of this register.

23.12.7 ECCRCFDnTRC — RSCFDn ECC Redundant Bit Data Control Test Register

In ECC test mode, this test register, for ECC data, consists of four 8-bit registers, ECCRCFDnSYND, ECCRCFDnHORD, ECCRCFDnECRD, and ECCRCFDnERDB.

When ECC test mode is enabled (ECCRCFDnTMC.ECTMCE = 1), this register can be accessed. When ECCRCFDnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when RSCFD is not accessed to RAM.

Access: This register can be read/written in 32-bit units.

Address: <ECCRCFDn_base> + 0008_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECCRCFDnSYND (See **Section 23.12.8**)								ECCRCFDnHORD (See **Section 23.12.9**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECCRCFDnECRD (See **Section 23.12.10**)								ECCRCFDnERDB (See **Section 23.12.11**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

23.12.8 ECCRCFDnSYND — RSCFDn ECC Decode Syndrome Data Register

In ECC test mode, this is a read-only register for storing generated syndrome code.

Writing to this register is ignored.

When ECC test mode is enabled (ECCRCFDnTMC.ECTMCE = 1), this register can be accessed.
When ECC test mode is disabled (ECCRCFDnTMC.ECTMCE = 0), 00_H is read.

Access: This is a read-only register that can be read in 8-bit units.

Address: <ECCRCFDn_base> + $000B_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SYND6	SYND5	SYND4	SYND3	SYND2	SYND1	SYND0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 23.194 ECCRCFDnSYND Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	SYND[6:0]	The generated syndrome code is stored as needed.

23.12.9 ECCRCFDnHORD — RSCFDn ECC 7-Bit Redundant Bit Data Hold Test Register

In ECC test mode, this register is used to store ECC data for read RAM data.

Writing to this register is ignored.

When ECC test mode is enabled (ECCRCFDnTMC.ECTMCE = 1), this register can be accessed.
When ECC test mode is disabled (ECCRCFDnTMC.ECTMCE = 0), 00_H is read.

Access: This is a read-only register that can be read in 8-bit units.

Address: <ECCRCFDn_base> + $000A_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	HORD6	HORD5	HORD4	HORD3	HORD2	HORD1	HORD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 23.195 ECCRCFDnHORD Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	HORD[6:0]	ECC code for read RAM data is stored as needed. When ECCRCFDnTMC.ECTRRS = 1 and ECCRCFDnTED register is read, ECC code is stored.

23.12.10 ECCRCFDnECRD — RSCFDn ECC Encode Test Register

In ECC test mode, this is a read-only register for storing ECC data generated for write RAM data.

Writing to this register is ignored.

When ECC test mode is enabled (ECCRCFDnTMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (ECCRCFDnTMC.ECTMCE = 0), 00_H is read.

Access: This is a read-only register that can be read in 8-bit units.

Address: <ECCRCFDn_base> + 0009_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ECRD6	ECRD5	ECRD4	ECRD3	ECRD2	ECRD1	ECRD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 23.196 ECCRCFDnECRD Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	ECRD[6:0]	These bits can read ECC data generated at the time of RAM data writing, and can read ECC data for the data written to the ECCRCFDnTED register when ECCRCFDnTMC.ECENS = 1.

23.12.11 ECCRCFDnERDB — RSCFDn ECC Redundant Bit Input/Output Replacement Buffer Register

In ECC test mode, this register handles ECC data.

The value of this register can be used as ECC data generated at the time of RAM writing or ECC data read at the time of RAM data reading.

When ECC test mode is enabled (ECCRCFDnTMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (ECCRCFDnTMC.ECTMCE = 0), 00_H is read.

Access: This register can be read/written in 8-bit units.

Address: <ECCRCFDn_base> + 0008_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ERDB6	ERDB5	ERDB4	ERDB3	ERDB2	ERDB1	ERDB0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.197 ECCRCFDnERDB Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6 to 0	ERDB[6:0]	When ECCRCFDnTMC.ECREOS = 1, these bits store the value of this register as ECC data to RAM. When ECCRCFDnTMC.ECREIS = 1, the value of this register is read as ECC data read from RAM. When ECCRCFDnTMC.ECTRRS = 1, ECC data to be stored in RAM is read instead of the write data for the read value of this register.

23.12.12 ECCRCFDnEAD0 — RSCFDn ECC Error Address Register

ECCRCFDnEAD0 is a read-only register to hold the address at which an ECC error has occurred.

Access: This register can be read/written in 32-bit units.

Address: <ECCRCFDn_base> + 0010_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEAD[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEAD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23.198 ECCRCFDnEAD0 Register Contents

Bit position	Bit Name	Function
31 to 0	ECEAD[31:0]	ECCRCFDnEAD0 is a read-only register to hold the address at which an ECC error has occurred. If an ECC error is detected while ECC error detection is enabled, the RAM address is latched using the detection signal as a trigger, and the address is stored in ECCRCFDnEAD0 as the address at which the ECC error has occurred. The address is stored upon detection of the first ECC error while no error status is set. However, if a 1-bit error is followed by a 2-bit error, the address of the latter is stored. Only one address can be held in ECCRCFDnEAD0.

23.13 Notes on the RS-CANFD Module

- When changing interface mode without resetting the RS-CANFD, write the value after reset to all registers and bits that are not allocated to the register map after change and then modify the RSCFDnCFDGRMCFG register.
- When changing a global mode, check the GSLPSTS, GHLTSTS, and GRSTSTS flags in the RSCFDn(CFD)GSTS register for transitions. When changing a channel mode, check the CSLPSTS, CHLTSTS, and CRSTSTS flags in the RSCFDn(CFD)CmSTS register (m = 0 to 2) for transitions.
- When only classical CAN frames are used in CAN FD mode, set the RSCFDnCFDCmDCFG register to the value equal to the set RSCFDnCFDCmNCFG register value.
- The acceptance filter processing checks receive rules sequentially in ascending order from the minimum rule number. If the same ID, IDE bit, or RTR bit value is set for multiple receive rules, the minimum number of receive rule is used for the acceptance filter processing. If the message does not pass through the subsequent DLC filter processing, the data processing is terminated without returning to the acceptance filter processing and the message is not stored in the buffer.
- When linking transmit buffers to transmit/receive FIFO buffers or allocating transmit buffers to transmit queues, set the control register (RSCFDn(CFD)TMCp) of the corresponding transmit buffer to 00_H. The status register (RSCFDn(CFD)TMSTSp) of the corresponding transmit buffer should not be used. Flags in other status registers (registers RSCFDn(CFD)TMTRSTS0 to RSCFDn(CFD)TMTRSTS1, RSCFDn(CFD)TMTARSTS0 to RSCFDn(CFD)TMTARSTS1, RSCFDn(CFD)TMTCSTS0 to RSCFDn(CFD)TMTCSTS1, and RSCFDn(CFD)TMTASTS0 to RSCFDn(CFD)TMTASTS1), which correspond to transmit buffers linked to transmit/receive FIFO buffers or allocated to transmit queues remain unchanged. Set the enable bit in the corresponding interrupt enable register (registers RSCFDn(CFD)TMIEC0 to RSCFDn(CFD)TMIEC1) to 0 (transmit buffer interrupt is disabled).
- When using transmit buffer merge mode (in CAN FD mode), write 00H to the control register (RSCFDn(CFD)TMCp) of the transmit buffer corresponding to the transmit buffer allocated as a payload storage area. Set the enable bit of corresponding interrupt enable registers (RSCFDn(CFD)TMIEC0 to RSCFDn(CFD)TMIEC1) to 0 (to disable interrupts).
- Transmit buffers that are linked to transmit/receive FIFO buffers must not be allocated to transmit queues. Do not allocate a transmit buffer allocated as a payload storage area in transmit buffer merge mode (in CAN FD mode) to the transmit queue either.
- Only a single transmit/receive FIFO buffer can be linked to a transmit buffer. Do not link two or more transmit/receive FIFO buffers to transmit buffers of the same number.
- When the CANm bit time clock is selected as a timestamp counter clock source, the timestamp counter stops when the corresponding channel has transitioned to channel reset mode or channel halt mode.
- In case of an attempt to store a new received message when the receive FIFO buffer and the transmit/receive FIFO buffer are full, the new message is discarded. If you wish to store a new transmit message in the transmit/receive FIFO buffer or the transmit queue, check that the transmit/receive FIFO buffer or the transmit queue is not full.
- In the case of registers that access the RAM, the value after reset shown in **Section 23.3, Registers (Classical CAN Mode)** and Section 23.4, Registers (CAN FD Mode) indicate the values cleared by initialization of the CAN RAM. Values before clear are undefined. The

following registers apply.

- Receive rule (RSCFDn(CFD)GAFLIDj, RSCFDn(CFD)GAFLMj, RSCFDn(CFD)GAFLP0_j, RSCFDn(CFD)GAFLP1_j registers)
- Receive buffers (RSCFDn(CFD)RMIDq, RSCFDn(CFD)RMPTRq, RSCFDnCFDRMFDSTSq, RSCFDn(CFD)RMDFb_q registers)
- Receive FIFO buffer access registers (RSCFDn(CFD)RFIDx, RSCFDn(CFD)RFPTRx, RSCFDnCFDRFFDSTSx, and RSCFDn(CFD)RFDFd_x registers)
- Transmit/receive FIFO buffer access registers (RSCFDn(CFD)CFIDk, RSCFDn(CFD)CFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDn(CFD)CFDFd_k registers)
- Transmit buffers (RSCFDn(CFD)TMIDp, RSCFDn(CFD)TMPTRp, RSCFDnCFDTMFDCTRp, and RSCFDn(CFD)TMDFb_p registers)
- Transmission history access register (RSCFDn(CFD)THLACCm registers)
- RAM test page access register (RSCFDn(CFD)RPGACCr registers)

- The values of unused receive buffers (RSCFDn(CFD)RMIDq, RSCFDn(CFD)RMPTRq, RSCFDnCFDRMFDSTSq, and RSCFDn(CFD)RMDFb_q registers), receive FIFO buffer access registers (RSCFDn(CFD)RFIDx, RSCFDn(CFD)RFPTRx, RSCFDnCFDRFFDSTSx, and RSCFDn(CFD)RFDFd_x registers) and transmit/receive FIFO buffer access registers (RSCFDn(CFD)CFIDk, RSCFDn(CFD)CFPTRk, RSCFDnCFDCFFDCSTSk, and RSCFDn(CFD)CFDFd_k registers) are undefined when the RS-CANFD module transitions to global operation mode or global test mode after exiting from global reset mode.

Section 24 Ethernet AVB MAC (ETNB)

This section contains a generic description of the Ethernet AVB MAC (ETNB).

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

24.1 Overview of RH850/D1L/D1M Ethernet AVB MAC (ETNB)

24.1.1 Units

This microcontroller has the following number of units of the Ethernet AVB MAC (ETNB).

Table 24.1 Units

Ethernet AVB MAC (ETNB)	D1L1	D1L2(H)	D1M1(H) D1M1-V2 D1M1A	D1M2(H)
Units	0	0	1	1
Names	–	–	ETNB0	ETNB0

Units index n

Throughout this section, the individual units of an Ethernet AVB MAC are identified by the index "n" (n = 0).

24.1.2 Register addresses

All Ethernet AVB MACs register addresses are given as address offsets from the individual base addresses <ETNBn_base>.
The <ETNBn_base> addresses of each ETNBn are listed in the following table:

Table 24.2 Register base addresses <ETNBn_base>

ETNBn unit	<ETNBn_base> address
ETNB0	FFDE 0000_H

24.1.3 Clock supply

All Ethernet AVB MACs provide three clock inputs.

Table 24.3 Clock supply

ETNBn unit	ETNBn clock	Connected to
ETNB0	PBUS clock	Clock Controller ETNBPCLK
	Cross-connect clock	Clock Controller ETNBXCCLK
	AVB clock	Clock Controller CLKFIX
	External clock	Clock Controller CLKFIX

24.1.4 Interrupts

The Ethernet AVB MACs can generate the following interrupt requests:

Table 24.4 ETNBn interrupt requests

ETNBn signals	Function	Connected to
ETNB0:		
pif_intr_line_0_n	Interrupt request line 0	Interrupt Controller INTETNB0LINE0
pif_intr_line_1_n	Interrupt request line 1	Interrupt Controller INTETNB0LINE1
pif_intr_line_2_n	Interrupt request line 2	Interrupt Controller INTETNB0LINE2
pif_intr_line_3_n	Interrupt request line 3	Interrupt Controller INTETNB0LINE3

24.1.5 Reset sources

The Ethernet AVB MACs and their registers are initialized by the following reset signal:

Table 24.5 Reset sources

ETNBn unit	Reset signal
ETNB0	• Reset Controller SYSRES • Reset Controller ETNB0RES • reset upon wake-up from DEEPSTOP mode

CAUTION

By default the ETNB0RES reset is active.
Thus before accessing this module this reset must be released via the MRSTC register of the Reset Controller. Refer to Section 9, Reset Controller for details.

24.1.6 I/O signals

The following table shows the I/O signals of the Ethernet AVB MAC.

Table 24.6 ETNBn I/O signals

ETNBn signals	Function	Connected to
ETNB0		
AVB_MDIO	Management information transmit/receive data	Port ETNB0MDIO
AVB_MDC	Management information transfer clock signal	Port ETNB0MDC
AVB_TXD[3:0]	Transmit data signal	Port ETNB0TXD[3:0]
AVB_TX_EN	Transmit data enable signal	Port ETNB0TXEN
AVB_TX_ER*1	Transmit error signal	Port ETNB0TXER
AVB_COL*1	Collision detection signal	Port ETNB0COL
AVB_CRS*1	Carrier detection signal	Port ETNB0CRS
AVB_TX_CLK	Transmit clock signal	Port ETNB0TXCLK
AVB_RX_CLK	Receive clock signal	Port ETNB0RXCLK
AVB_RXD[3:0]	Receive data signal	Port ETNB0RXD[3:0]
AVB_RX_DV	Receive data enable signal	Port ETNB0RXDV
AVB_RX_ER*1	Reception error signal	Port ETNB0RXER

Note 1. The Ethernet AVB MAC (ETNB) supports the MII-Lite interface.
When MII-Lite interface is used the signals AVB_TX_ER(out), AVB_RX_ER(in), AVB_COL(in) and AVB_CRS(in) are not used. The unused input signals are tied to low by internal circuit. Port pins with this unused ETNB functions can be used for other port mode or alternative mode functions.
Example: Port pin P42_15 (ETNB0RXER) can be used in port mode as GPIO input or output pin.

24.1.7 Bus master ID

The Ethernet AVB MAC Unit bus master interface is connected to the cross-connect system. The master interface has the following master ID:

ETNB0: MSTID6

24.2 Overview

The EthernetAVB is an on-chip Ethernet controller (E-MAC) that conforms to the definition of the MAC (Media Access Control) layer for Ethernet in the IEEE 802.3 standard.

When connected with a physical-layer LSI chip (PHY-LSI) that complies with the standard, the E-MAC is able to transmit and receive Ethernet (IEEE 802.3) frames. The E-MAC has a single MAC layer interface.

The EthernetAVB has a dedicated direct memory access controller (AVB-DMAC) for transferring transmitted Ethernet frames to and received Ethernet frames from respective storage areas in the URAM at high speed.

The AVB-DMAC is compliant with the following three standards formulated for IEEE 802.1BA: the IEEE 802.1AS timing and synchronization protocol, the IEEE 802.1Qav real-time transfer, and the IEEE 802.1Qat stream reservation protocol.

In this section, URAM refers to the CPU local RAM (LRAM), video RAM (VRAM) or external SDRAM.
For details about the memory access options refer to Section 14.2.3, Cross-connect details.

24.2.1 Specifications (Functions)

Table 24.7 lists the specifications of the EthernetAVB module.

Table 24.7 Specifications (Functions)

Item	Description
Protocol	Flow control conforming with the IEEE 802.3x standard
Data transmission and reception	Transmission and reception of Ethernet (IEEE 802.3) frames
Transfer speed	Supports transfer at 10 and 100 Mbps
Mode	• Full-duplex mode • Half-duplex mode
Interface	Supports the IEEE 802.3 standard MII (Media Independent Interface)
Summary of the EthernetAVB function	An intelligent frame separation DMAC (AVB-DMAC) conforming with the following standards stipulated for IEEE 802.1BA: • IEEE 802.1AS (time synchronization protocol) • IEEE 802.1Qav (real-time transfer) • IEEE 1722 (AVTP presentation timestamp) IEEE 802.1Qat is supported by software. • Descriptor management system • Identification and sorting of frame data, and extraction and gathering of valid data • Controllable interrupt frequency (reducing the load on the CPU)
Magic Packet™	Detection of Magic Packets™*1

Note 1. Magic Packet™ is a trademark of Advanced Micro Devices, Inc.

24.2.2 Block Diagram

Figure 24.1 is a block diagram of the EthernetAVB.

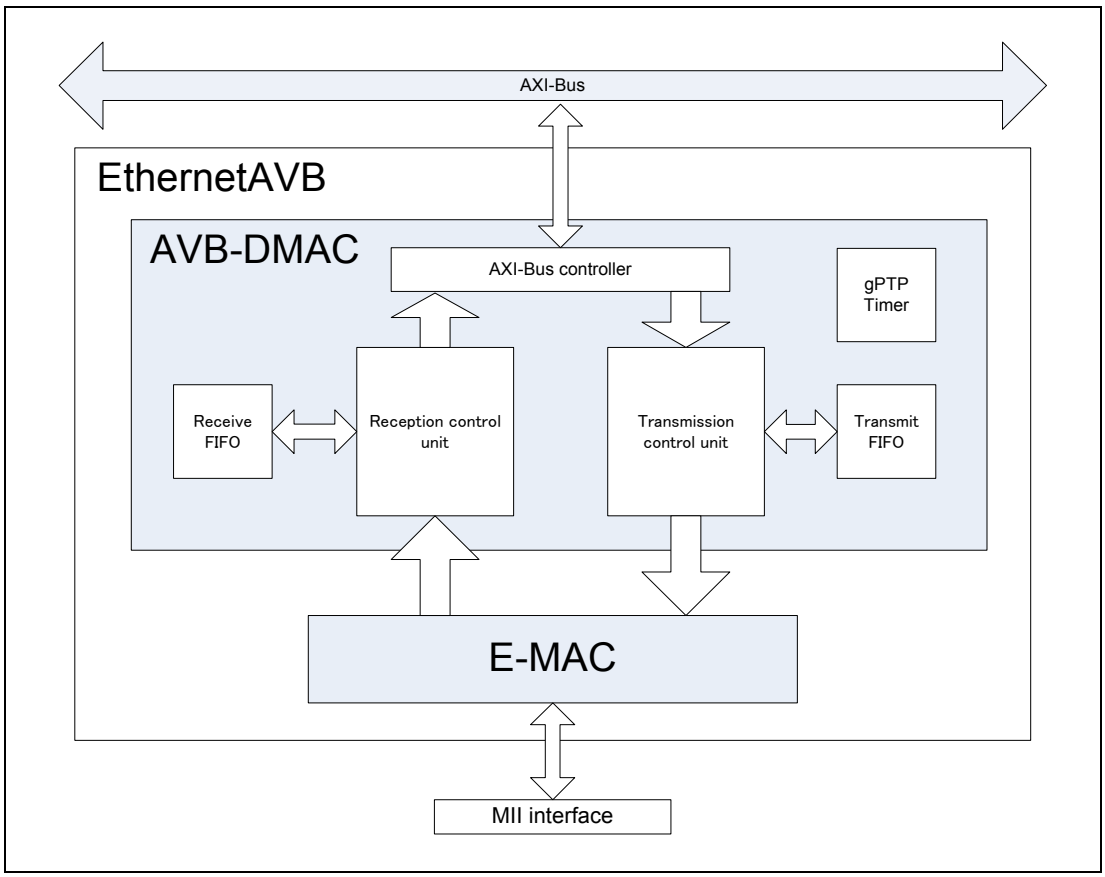


Figure 24.1 Block Diagram of EthernetAVB

24.3 Register Descriptions

Table 24.8 and **Table 24.9** lists the EthernetAVB related registers and their configurations.

Table 24.8 Configuration of AVB-DMAC-related Registers (1/2)

Module Name	Register Name	Symbol	Value after Reset	Address	Access size
ETNBn	AVB-DMAC mode register	CCC	H'0000 0000	<ETNBn_base> + 000_H	32
ETNBn	Descriptor base address table register	DBAT	H'0000 0000	<ETNBn_base> + 004_H	32
ETNBn	Descriptor base address load request register	DLR	H'003F FFFF	<ETNBn_base> + 008_H	32
ETNBn	AVB-DMAC status register	CSR	H'0000 0001	<ETNBn_base> + $00C_H$	32
ETNBn	Current descriptor address register q (q = 0 to 21)	CDARq	H'0000 0000	<ETNBn_base> + 010_H+q*4	32
ETNBn	Error Status Register	ESR	H'0000 0000	<ETNBn_base> + 088_H	32
ETNBn	Receive configuration register	RCR	H'1800 0000	<ETNBn_base> + 090_H	32
ETNBn	Receive queue configuration register i (i = 0 to 4)	RQCi	H'0000 0000	<ETNBn_base> + 094_H+i*4	32
ETNBn	Receive padding configuration register	RPC	H'0000 0100	<ETNBn_base> + $0B0_H$	32
ETNBn	Unread frame counter warning level register	UFCW	H'0000 0000	<ETNBn_base> + $0BC_H$	32
ETNBn	Unread frame counter stop level register	UFCS	H'0000 0000	<ETNBn_base> + $0C0_H$	32
ETNBn	Unread frame counter register i (i = 0 to 4)	UFCVi	H'0000 0000	<ETNBn_base> + $0C4_H$+i*4	32
ETNBn	Unread frame counter decrement register i (i = 0 to 4)	UFCDi	H'0000 0000	<ETNBn_base> + $0E0_H$+i*4	32
ETNBn	Separation filter offset register	SFO	H'0000 0000	<ETNBn_base> + $0FC_H$	32
ETNBn	Separation filter pattern register i (i = 0 to 31)	SFPi	H'0000 0000	<ETNBn_base> + 100_H+i*4	32
ETNBn	Separation filter mask register i (i = 0, 1)	SFMi	H'0000 0000	<ETNBn_base> + $1C0_H$+i*4	32
ETNBn	Transmit configuration register	TGC	H'0022 2200	<ETNBn_base> + 300_H	32
ETNBn	Transmit configuration control register	TCCR	H'0000 0000	<ETNBn_base> + 304_H	32
ETNBn	Transmit status register	TSR	H'0000 0000	<ETNBn_base> + 308_H	32
ETNBn	Time stamp FIFO access register 0	TFA0	H'0000 0000	<ETNBn_base> + 310_H	32
ETNBn	Time stamp FIFO access register 1	TFA1	H'0000 0000	<ETNBn_base> + 314_H	32
ETNBn	Time stamp FIFO access register 2	TFA2	H'0000 0000	<ETNBn_base> + 318_H	32
ETNBn	CBS increment value register c (c= 0, 1)	CIVRc	H'0000 0001	<ETNBn_base> + 320_H+c*4	32
ETNBn	CBS decrement value register c (c = 0, 1)	CDVRc	H'FFFF FFFF	<ETNBn_base> + 328_H+c*4	32
ETNBn	CBS upper limit register c (c = 0, 1)	CULc	H'7FFF FFFF	<ETNBn_base> + 330_H+c*4	32
ETNBn	CBS lower limit register c (c = 0, 1)	CLLc	H'8000 0001	<ETNBn_base> + 338_H+c*4	32
ETNBn	Descriptor interrupt control register	DIC	H'0000 0000	<ETNBn_base> + 350_H	32
ETNBn	Descriptor interrupt status register	DIS	H'0000 0000	<ETNBn_base> + 354_H	32
ETNBn	Error interrupt control register	EIC	H'0000 0000	<ETNBn_base> + 358_H	32
ETNBn	Error interrupt status register	EIS	H'0000 0000	<ETNBn_base> + $35C_H$	32
ETNBn	Receive interrupt control register 0	RIC0	H'0000 0000	<ETNBn_base> + 360_H	32
ETNBn	Receive interrupt status register 0	RIS0	H'0000 0000	<ETNBn_base> + 364_H	32
ETNBn	Receive interrupt control register 1	RIC1	H'0000 0000	<ETNBn_base> + 368_H	32
ETNBn	Receive interrupt status register 1	RIS1	H'0000 0000	<ETNBn_base> + $36C_H$	32
ETNBn	Receive interrupt control register 2	RIC2	H'0000 0000	<ETNBn_base> + 370_H	32
ETNBn	Receive interrupt status register 2	RIS2	H'0000 0000	<ETNBn_base> + 374_H	32
ETNBn	Transmit interrupt control register	TIC	H'0000 0000	<ETNBn_base> + 378_H	32
ETNBn	Transmit interrupt status register	TIS	H'0000 0000	<ETNBn_base> + $37C_H$	32

Table 24.8 Configuration of AVB-DMAC-related Registers (2/2)

Module Name	Register Name	Symbol	Value after Reset	Address	Access size
ETNBn	Interrupt summary status register	ISS	H'0000 0000	<ETNBn_base> + 380_H	32
ETNBn	gPTP configuration control register	GCCR	H'0000 003C	<ETNBn_base> + 390_H	32
ETNBn	gPTP maximum transit time register	GMTT	H'0000 0000	<ETNBn_base> + 394_H	32
ETNBn	gPTP presentation time comparison register	GPTC	H'0000 0000	<ETNBn_base> + 398_H	32
ETNBn	gPTP timer increment register	GTI	H'0000 0001	<ETNBn_base> + $39C_H$	32
ETNBn	gPTP timer offset register i (i = 0 to 2)	GTOi	H'0000 0000	<ETNBn_base> + $3A0_H$+i*4	32
ETNBn	gPTP interrupt control register	GIC	H'0000 0000	<ETNBn_base> + $3AC_H$	32
ETNBn	gPTP interrupt status register	GIS	H'0000 0000	<ETNBn_base> + $3B0_H$	32
ETNBn	gPTP timer capture register i (i = 0 to 2)	GCTi	H'0000 0000	<ETNBn_base> + $3B8_H$+i*4	32

Table 24.9 Configuration of E-MAC-related Registers

Module Name	Register Name	Symbol	Value after Reset	Address	Access size
ETNBn	E-MAC mode register	ECMR	H'0000 0000	<ETNBn_base> + 500_H	32
ETNBn	Receive frame length register	RFLR	H'0000 0000	<ETNBn_base> + 508_H	32
ETNBn	E-MAC status register	ECSR	H'0000 0000	<ETNBn_base> + 510_H	32
ETNBn	E-MAC interrupt permission register	ECSIPR	H'0000 0000	<ETNBn_base> + 518_H	32
ETNBn	PHY interface register	PIR	H'0000 0000	<ETNBn_base> + 520_H	32
ETNBn	Auto PAUSE frame time parameter register	APFTP	H'0000 0000	<ETNBn_base> + 554_H	32
ETNBn	Manual PAUSE frame register	MPR	H'0000 0000	<ETNBn_base> + 558_H	32
ETNBn	PAUSE frame transmit counter	PFTCR	H'0000 0000	<ETNBn_base> + $55C_H$	32
ETNBn	PAUSE frame receive counter	PFRCR	H'0000 0000	<ETNBn_base> + 560_H	32
ETNBn	EthernetAVB Mode Register	GECMR	H'0000 0000	<ETNBn_base> + $5B0_H$	32
ETNBn	E-MAC address high register	MAHR	H'0000 0000	<ETNBn_base> + $5C0_H$	32
ETNBn	E-MAC address low register	MALR	H'0000 0000	<ETNBn_base> + $5C8_H$	32
ETNBn	Transmit retry over counter register	TROCR	H'0000 0000	<ETNBn_base> + 700_H	32
ETNBn	Lost carrier counter register	LCCR	H'0000 0000	<ETNBn_base> + 710_H	32
ETNBn	CRC error frame receive counter register	CEFCR	H'0000 0000	<ETNBn_base> + 740_H	32
ETNBn	Frame receive error counter register	FRECR	H'0000 0000	<ETNBn_base> + 748_H	32
ETNBn	Too-short frame receive counter register	TSFRCR	H'0000 0000	<ETNBn_base> + 750_H	32
ETNBn	Too-long frame receive counter register	TLFRCR	H'0000 0000	<ETNBn_base> + 758_H	32
ETNBn	Residual-bit frame receive counter register	RFCR	H'0000 0000	<ETNBn_base> + 760_H	32
ETNBn	Multicast address frame receive counter register	MAFCR	H'0000 0000	<ETNBn_base> + 778_H	32

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above tables.

24.3.1 AVB-DMAC Mode Register (CCC)

The CCC register specifies the operating mode of the AVB-DMAC.

Access: This register can be read/written in 8/16/32-bit units.

Address: <ETNBn_base> + 000_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	FCE	LBME	—	—	—	BOC	—	—	CSEL[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTSR	—	—	—	—	—	—	OPC[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W

Table 24.10 CCC register contents

Bit Position	Bit Name	Function
b31 to b26	—	Reserved These bits are read as 0. The write value should be 0.
b25	FCE	Flow Control Enable 0: Flow control disabled 1: Flow control enabled
b24	LBME	Loopback Mode Enable 0: Normal operation 1: Loopback mode is enabled.
b23 to b21	—	Reserved These bits are read as 0. The write value should be 0.
b20	BOC	First Byte Specification 0: The first byte is the 8 lower-order bits (URAM[7:0]) 1: The first byte is the 8 higher-order bits (URAM[31:24])
b19, b18	—	Reserved These bits are read as 0. The write value should be 0.
b17, b16	CSEL[1:0]	gPTP Clock Select B'00: gPTP is not in use. B'01: Peripheral bus clock B'10: Ethernet transmission clock B'11: External clock
b15 to b9	—	Reserved These bits are read as 0. The write value should be 0.
b8	DTSR	Data Transmission Suspend Request 0: Normal operation 1: Requests suspension
b7 to b2	—	Reserved These bits are read as 0. The write value should be 0.
b1, b0	OPC[1:0]	Operating Mode Configuration B'00: Reset mode B'01: Configuration mode B'10: Operation mode B'11: Standby mode

FCE (Flow Control Enable) Bit

This bit enables the flow control support of MAC.

When flow control is enabled, the MAC gets informed about the Rx-FIFO level (Rx-FIFO fill level reached RCR.RFCL[12:0]).

LBME (Loopback Mode Enable) Bit

This bit enables loopback mode.

In loopback mode, the transmission lines are internally connected to the reception lines. When loopback mode is to be used, the Ethernet transmission clock must be supplied to the MII interface. A received clock signal is not required. Writing to this bit is only possible when the current operating mode is configuration mode.

CAUTION

Data for transmission are still output normally. To eliminate effects on external modules, pin control should be applied to block the output of data. For pin control, see section 2, Pins.

BOC (First Byte Specification) Bit

Specifies the allocation of the first byte from a received Ethernet frame to the URAM.

This configuration setting does not affect the format and filter parameters of the descriptor in the URAM.

Writing to this bit is only possible when the current operating mode is configuration mode.

Figure 24.2 to **Figure 24.4** show how data from frames received via the Ethernet connection are stored in the URAM.

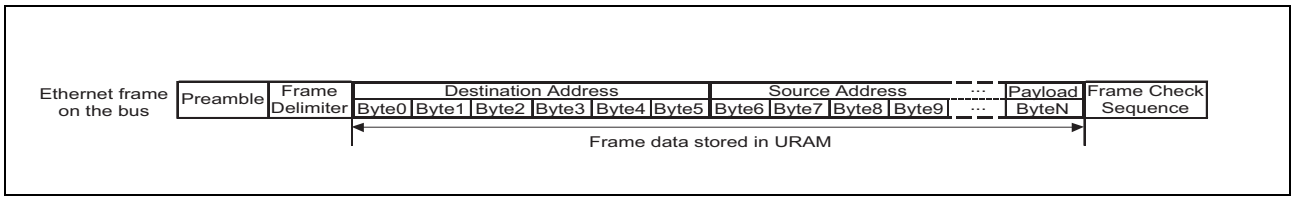


Figure 24.2 Data for Reception in an Ethernet Frame

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPTR+0	Byte3								Byte2								Byte1								Byte0							
DPTR+4	Byte7								Byte6								Byte5								Byte4							
DPTR+8	Byte11								Byte10								Byte9								Byte8							

Figure 24.3 When CCC.BOC = 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPTR+0	Byte0								Byte1								Byte2								Byte3							
DPTR+4	Byte4								Byte5								Byte6								Byte7							
DPTR+8	Byte8								Byte9								Byte10								Byte11							

Figure 24.4 When CCC.BOC = 1

CSEL[1:0] (gPTP Clock Select) Bits

These bits select the clock source for the gPTP timer.

Writing to these bits is only possible when the current operating mode is configuration mode.

DTSR (Data Transmission Suspend Request) Bit

This bit can suspend access to the URAM.

The access is suspended on completion of the transfer of the frame currently being transferred.

This function disables access to the URAM without affecting normal operation of the AVB-DMAC. Use this bit when exclusive control over the contents of the URAM is necessary, for example, in checking its integrity.

Note that the transmission and reception queues are not processed while access is suspended.

Change neither the AVB-DMAC settings nor the mode while access is suspended.

OPC[1:0] (Operating Mode Configuration) Bits

These bits specify the operating mode.

For the operating modes, see Section 24.4.1.1, Operating Modes.

Writing to this bit is possible in any of the operating modes, but should not be done after the application system has issued a Power Off request.

24.3.2 Descriptor Base Address Table Register (DBAT)

The DBAT register is used to set the base address of the descriptor table.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 004_H

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.11 DBAT register contents

Bit Position	Bit Name	Function
b31 to b0	TA[31:0]	Descriptor Base Table Address Base address of the descriptor table in the URAM

CAUTION

The setting of this bit must be a multiple of four (i.e. b0 and b1 must be set to 0).

TA[31:0] (Descriptor Base Table Address) Bits

These bits specify the base address of the descriptor table in the URAM.

For the structure of this table, see Section 24.4.3, Descriptors.

Writing to this bit is only possible when the current operating mode is configuration mode.

24.3.3 Descriptor Base Address Load Request Register (DLR)

The DLR register is used to issue a request to load the values from the current descriptor address register q (CDARq) for each queue to the descriptor base address table register (DBAT).

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 008_H

Initial value: $003F\ FFFF_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	LBA21	LBA20	LBA19	LBA18	LBA17	LBA16
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LBA15	LBA14	LBA13	LBA12	LBA11	LBA10	LBA9	LBA8	LBA7	LBA6	LBA5	LBA4	LBA3	LBA2	LBA1	LBA0
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.12 DLR register contents (1/3)

Bit Position	Bit Name	Function
b31 to 22	—	Reserved These bits are read as 0. The write value should be 0.
b21	LBA21	Base Address Load Request (Rx17: Stream 15) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b20	LBA20	Base Address Load Request (Rx16: Stream 14) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b19	LBA19	Base Address Load Request (Rx15: Stream 13) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b18	LBA18	Base Address Load Request (Rx14: Stream 12) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b17	LBA17	Base Address Load Request (Rx13: Stream 11) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b16	LBA16	Base Address Load Request (Rx12: Stream 10) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.

Table 24.12 DLR register contents (2/3)

Bit Position	Bit Name	Function
b15	LBA15	Base Address Load Request (Rx11: Stream 9) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b14	LBA14	Base Address Load Request (Rx10: Stream 8) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b13	LBA13	Base Address Load Request (Rx9: Stream 7) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b12	LBA12	Base Address Load Request (Rx8: Stream 6) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b11	LBA11	Base Address Load Request (Rx7: Stream 5) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b10	LBA10	Base Address Load Request (Rx6: Stream 4) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b9	LBA9	Base Address Load Request (Rx5: Stream 3) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b8	LBA8	Base Address Load Request (Rx4: Stream 2) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b7	LBA7	Base Address Load Request (Rx3: Stream 1) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b6	LBA6	Base Address Load Request (Rx2: Stream 0) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b5	LBA5	Base Address Load Request (Rx1: Network Control) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b4	LBA4	Base Address Load Request (Rx0: Best Effort) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.

Table 24.12 DLR register contents (3/3)

Bit Position	Bit Name	Function
b3	LBA3	Base Address Load Request (Tx3: Stream Class A) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b2	LBA2	Base Address Load Request (Tx2: Stream Class B) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b1	LBA1	Base Address Load Request (Tx1: Network Control) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.
b0	LBA0	Base Address Load Request (Tx0: Best Effort) 0: No load request is issued. 1: When written: A request for loading the corresponding base address is issued. When read: The given base address is being loaded.

LBAq (Base Address Load Request) Bits

Each bit is used to issue requests to load base addresses and to indicate that a base address is currently being loaded.

Setting a bit to 1 issues a request for loading the descriptor base address for the queue q.

If transfer is currently in progress, loading is executed on completion of transfer for the current frame.

Completion of loading leads to automatic setting of the corresponding bit to 0.

For the transmission queues, base address load requests are executed even while fetching is in progress (the transmit start request bit in the transmit configuration control register (TCCR.TSRQt) is 1). Therefore, be sure to check that fetching is not in progress before issuing a request.

Writing to a bit of this register is only possible when the current operating mode is configuration mode.

Only 1 can be written to this bit.

24.3.4 AVB-DMAC Status Register (CSR)

The CSR register is used to indicate the operating mode in which the AVB-DMAC is running and the individual communications states.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $00C_H$

Initial value: $0000\ 0001_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	RPO	TPO3	TPO2	TPO1	TPO0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTS	—	—	—	—	OPS[3:0]			
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.13 CSR register contents

Bit Position	Bit Name	Function
b31 to b21	—	Reserved These bits are read as 0. The write value should be 0.
b20	RPO	Receive Process Status 0: Normal operation 1: Reception is in progress.
b19	TPO3	Transmit Process Status 3 (Stream Class A) 0: Normal operation 1: Transmission is in progress.
b18	TPO2	Transmit Process Status 2 (Stream Class B) 0: Normal operation 1: Transmission is in progress.
b17	TPO1	Transmit Process Status 1 (Network Control) 0: Normal operation 1: Transmission is in progress.
b16	TPO0	Transmit Process Status 0 (Best Effort) 0: Normal operation 1: Transmission is in progress.
b15 to b9	—	Reserved These bits are read as 0. The write value should be 0.
b8	DTS	Data Transmission Suspended Status 0: Normal operation 1: Transmission is suspended.
b7 to b4	—	Reserved These bits are read as 0. The write value should be 0.
b3 to b0	OPS[3:0]	Operating Mode Status B'0001: Reset mode B'0010: Configuration mode B'0100: Operation mode B'1000: Standby mode Other settings are reserved.

RPO (Receive Process Status) Bit

This bit indicates whether a reception queue contains an unread received frame.

This bit being set to 1 indicates that a received frame is yet to be stored in the URAM.

- [Clearing conditions]
  The current operating mode is not operation mode.
  Received frames in the reception FIFO all being stored in the URAM.
- [Setting condition]
  A received frame being stored in the reception FIFO (but not yet in the URAM)

TPO3 (Transmit Process Status 3) Bit

This bit indicates whether a class A stream is being transmitted.

This bit being set to 1 indicates that the AVB-DMAC is fetching data for transmission from the URAM, or the E-MAC is transmitting data.

- [Clearing conditions]
  The current operating mode is not operation mode.
  Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (TCCR.TSRQ3) is 0)
- [Setting condition]
  Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (TCCR.TSRQ3))

TPO2 (Transmit Process Status 2) Bit

This bit indicates whether a class B stream is being transmitted.

This bit being set to 1 indicates that the AVB-DMAC is fetching data for transmission from the URAM, or the E-MAC is transmitting data.

- [Clearing conditions]
  The current operating mode is not operation mode.
  Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (TCCR.TSRQ2) is 0)
- [Setting condition]
  Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (TCCR.TSRQ2))

TPO1 (Transmit Process Status 1) Bit

This bit indicates whether a network control is being transmitted.

This bit being set to 1 indicates that the AVB-DMAC is fetching data for transmission from the URAM, or the E-MAC is transmitting data.

- [Clearing conditions]
  The current operating mode is not operation mode.
  Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (TCCR.TSRQ1) is 0)
- [Setting condition]
  Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (TCCR.TSRQ1))

TPO0 (Transmit Process Status 0) Bit

This bit indicates whether a best effort is being transmitted.

This bit being set to 1 indicates that the AVB-DMAC is fetching data for transmission from the URAM, or the E-MAC is transmitting data.

- [Clearing conditions]
  The current operating mode is not operation mode.
  Completion of transfer of all frames for transmission from the transmission FIFO (the transmit start request bit in the transmit configuration control register (TCCR.TSRQ0) is 0)
- [Setting condition]
  Transmission being started (by writing 1 to the transmit start request bit in the transmit configuration control register (TCCR.TSRQ0))

DTS (Data Transmission Suspend Status) Bit

This bit indicates whether access to the URAM is enabled.

- [Clearing condition]
  The data transmission suspend request bit in the AVB-DMAC mode register (CCC.DTSR) being 0.
- [Setting condition]
  Access to the URAM not proceeding while the data transmission suspend request bit (CCC.DTSR) in the AVB-DMAC mode register (CCC) is 1 (if the URAM is being accessed, this bit is set to 1 on completion of access).

OPS[3:0] (Operating Mode Status) Bits

These bits indicate the current operating mode.

For the operating modes, see Section 24.4.1.1, Operating Modes.

24.3.5 Current Descriptor Address Register q (CDARq) (q = 0 to 21)

The CDARq register indicates the current descriptor address.

Access: This register is read-only in 32-bit units.

Address: CDAR0: <ETNBn_base> + 010_H; CDAR1: <ETNBn_base> + 014_H; CDAR2: <ETNBn_base> + 018_H
CDAR3: <ETNBn_base> + $01C_H$; CDAR4: <ETNBn_base> + 020_H; CDAR5: <ETNBn_base> + 024_H
CDAR6: <ETNBn_base> + 028_H; CDAR7: <ETNBn_base> + $02C_H$; CDAR8: <ETNBn_base> + 030_H
CDAR9: <ETNBn_base> + 034_H; CDAR10: <ETNBn_base> + 038_H; CDAR11: <ETNBn_base> + $03C_H$
CDAR12: <ETNBn_base> + 040_H; CDAR13: <ETNBn_base> + 044_H; CDAR14: <ETNBn_base> + 048_H
CDAR15: <ETNBn_base> + $04C_H$; CDAR16: <ETNBn_base> + 050_H; CDAR17: <ETNBn_base> + 054_H
CDAR18: <ETNBn_base> + 058_H; CDAR19: <ETNBn_base> + $05C_H$; CDAR20: <ETNBn_base> + 060_H
CDAR21: <ETNBn_base> + 064_H

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CDA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CDA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.14 CDARq (q = 0 to 21) register contents

Bit Position	Bit Name	Function
b31 to b0	CDA[31:0]	Current Descriptor Address The address of the current descriptors for the transmission queues

CDA[31:0] (Current Descriptor Address) Bits

CDAR0 to CDAR3 indicate the addresses of the current descriptors for the corresponding transmission queues while CDAR4 to CDAR21 indicate the addresses of the current descriptors for the corresponding reception queues.

If the operating mode is changed to operation mode, the contents of the register for the queue to be used are set in the descriptor base address table register (DBAT).

Also, when the descriptor base address load request register (DLR) issues a load request, the contents of the corresponding register are set in the descriptor base address table register (DBAT).

Conditions for updating:

These bits are set to 0 when the operating mode is not operation mode.

This register is updated in response to processing of the descriptor for a queue.

24.3.6 Error Status Register (ESR)

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 088_H

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	EIL	ET[3:0]				—	—	—	EQN[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.15 ESR register contents

Bit Position	Bit Name	Function
b31 to b13	—	Reserved These bits are read as 0. The write value should be 0.
b12	EIL	Error Information Lost 0: No loss of error information 1: Lost of error information detected
b11 to b8	ET[3:0]	Error Type B'0000: Read descriptor from URAM B'0001: Write descriptor to URAM B'0010: Interpret read descriptor B'0011: Tx-Buffer is corrupted B'0100: Read data from URAM B'0101: Write data or timestamp to URAM B'0110: Reading from Rx-FIFO B'0111: Rx-FIFO is corrupted B'1000: Frame size error during reception detected B'1001: Frame size error during transmission detected B'1010: Tx-Buffer overflow
b7 to b5	—	Reserved These bits are read as 0. The write value should be 0.
b4 to b0	EQN[4:0]	Error Queue Number

EIL Error Information Lost Bit

This bit indicates that error information detected by ETNB is lost because the previous reported error has not been processed by CPU.

[Changing condition]

This bit is set to 0 when leaving OPERATION mode.

This bit is set to 0 when CPU writes 0b to EIS.QEF.

This bit is set to 1 when the set condition of EIS.QEF is fulfilled while EIS.QEF is 1.

ET[3:0] Error Type Bits

These bits indicate details about the transfer stage which was handled when ETNB has detected an error.

When the fault is related to the read descriptor (ESR.ET[3:0] = B'0000 or B'0010), CPU needs to correct the faulty descriptor before the related queue can continue processing. Because the queue halts at the faulty descriptor CDARq.CDA[31:0] (with q = ESR.EQN[4:0]) is identifying the faulty descriptor directly.

When the fault is related to descriptor writing (ESR.ET[3:0] = B'0001), CPU needs recognise the not-updated or incorrectly updated descriptor in queue ESR.EQN[4:0]. The write problem is not influencing how ETNB processes the descriptor chain.

When the fault is related to the Tx-Buffer (ESR.ET[3:0] = B'0011) CPU needs to clean-up the Tx-Buffer to correct the buffer control structures.

All other errors are transient in nature and may be corrected by continuation of HW or SW operation; so there is no strong demand on CPU interaction.

The CPU should only evaluate these bits when EIS.QEF is 1.

[Changing condition]

These bits are updated when the set condition of EIS.QEF is fulfilled and EIS.QEF is 0.

EQN[4:0] Error Queue Number Bits

These bits indicate the queue number which was handled when ETNB has detected an error.

A fault reported for ESR.EQN[4:0] = 0 to 3 is related to transmit queue t = 0 to 3.
From ESR.EQN[4:0] = 4 the fault is related to receive queue r = ESR.EQN[4:0] - 4.

The CPU should only evaluate these bits when EIS.QEF is 1.

The CPU should not evaluate these bits when ESR.ET[3:0] is B'0011 or B'0111.

[Changing condition]

These bits are updated when the set condition of EIS.QEF is fulfilled and EIS.QEF is 0.

24.3.7 Receive Configuration Register (RCR)

The RCR register is used to make settings related to reception for the AVB-DMAC.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 090_H

Initial value: $1800\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	RFCL[12:0]												
Value after reset	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	ETS2	ETS0	ESF[1:0]		ENCF	EFFS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.16 RCR register contents

Bit Position	Bit Name	Function
b31 to b29	—	Reserved These bits are read as 0. The write value should be 0.
b28 to b16	RFCL[12:0]	Receive FIFO Caution Level Recommended value: H'1800
b15 to b6	—	Reserved These bits are read as 0. The write value should be 0.
b5	ETS2	Time Stamp Enable (Stream) 0: Time stamping is disabled. 1: Time stamping is enabled. Recommended value: 0
b4	ETS0	Time Stamp Enable (Best Effort) 0: Time stamping is disabled. 1: Time stamping is enabled. Recommended value: 0
b3, b2	ESF[1:0]	Stream Filtering Select Settings for reception queues 2 to 17 B'00: Filtering is disabled. Frames are processed in queue 0 (best effort). B'01: The filter for separating AVB stream frames from non-AVB stream frames is enabled; non-matching frames from the stream are processed in queue 0 (best effort). B'10: The filter for separating AVB stream frames from non-AVB stream frames is enabled; non-matching frames are discarded. B'11: The filter for separating AVB stream frames from non-AVB stream frames is enabled; non-matching frames from a stream are processed in queue 0 (best effort). Recommended value: 10 or 11
b1	ENCF	Network Control Filtering Enable Setting for reception queue 1 (network control) 0: Network control is disabled. 1: Network control is enabled.
b0	EFFS	Error Frame Enable 0: Error frames are disabled. 1: Error frames are enabled. Recommended value: 0

RFCL[12:0] (Receive FIFO Caution Level) Bits

These bits set the caution level for the reception FIFO and are used to maintain the priority order of the storage of received data and the fetching of data for transmission.

If the reception FIFO contains less data than this level, processing of both transmission and reception queues becomes pending.

If the reception FIFO contains more data than this level, only data in the reception queue are transferred, and processing of the transmission queue becomes pending.

Writing to this bit is only possible when the current operating mode is configuration mode.

CAUTION

- **The setting of this bit must be a multiple of four (i.e. set RFCL[1:0] = B'00).**
- **In the case of this LSI chip, set these bits to H'1800.**

ETS2 (Time Stamp Enable (Stream)) Bit

Enables the inclusion of time-stamp information in reception queues 2 to 17.

Writing to this bit is only possible when the current operating mode is configuration mode.

ETS0 (Time Stamp Enable (Best Effort)) Bit

Enables the inclusion of time-stamp information in reception queue 0.

Writing to this bit is only possible when the current operating mode is configuration mode.

ESF[1:0] (Stream Filtering Select) Bits

These bits select separation filtering for reception queues 2 to 17.

The queue-dependent separation filter can be used in combination with the identification of AVB stream frames.

When the value is B'00, filtering is disabled and frames from streams are processed in reception queue 0 (best effort).

When the value is B'01, the separation filter is enabled for both AVB stream frames and non-AVB stream frames; frames from non-matching streams are processed in reception queue 0 (best effort).

When the value is B'10, the separation filter is enabled for AVB stream frames; frames from non-matching streams are discarded.

When the value is B'11, the separation filter is enabled for AVB stream frames; frames from non-matching streams are processed in reception queue 0 (best effort).

For separation filtering, see **24.4.4.1** (1) Separation Filtering.

Writing to this bit is only possible when the current operating mode is configuration mode.

ENCF (Enable Network Control Filtering) Bit

Enables the AVB network control frame for reception queue 1.

When reception queue 1 is disabled, a received frame is stored in reception queue 0 (best effort).

Writing to this bit is only possible when the current operating mode is configuration mode.

EFFS (Enable Error Frame) Bit

Enables or disables the reception of frames that have been classified as error frames by the E-MAC.

Received error frames are stored in reception queue 0 (best effort).

An indicator of error detection by the E-MAC during reception is stored in the descriptor (DESCR.MS).

Writing to this bit is only possible when the current operating mode is configuration mode.

24.3.8 Receive Queue Configuration Register i (RQCi) (i = 0 to 4)

The RQC0 register is used to set up reception queues 0 to 3.

The RQC1 register is used to set up reception queues 4 to 7.

The RQC2 register is used to set up reception queues 8 to 11.

The RQC3 register is used to set up reception queues 12 to 15.

The RQC4 register is used to set up reception queues 16 to 17.

Access: This register can be read/written in 8/16/32-bit units.

Address: RQC0: <ETNBn_base> + 094_H RQC1: <ETNBn_base> + 098_H RQC2: <ETNBn_base> + $09C_H$
RQC3: <ETNBn_base> + $0A0_H$ RQC4: <ETNBn_base> + $0A4_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	UFCC3[1:0]		—	—	RSM3[1:0]		—	—	UFCC2[1:0]		—	—	RSM2[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	UFCC1[1:0]		—	—	RSM1[1:0]		—	—	UFCC0[1:0]		—	—	RSM0[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 24.17 RQCi register contents (1/2)

Bit Position	Bit Name	Function
b31, b30	—	Reserved These bits are read as 0. The write value should be 0.
b29, b28	UFCC3[1:0]	Unread Frame Counter Configuration (Receive Queue 3+i*4) These bits set the unread frame counter used in reception queue 3+4*i.
b27, b26	—	Reserved These bits are read as 0. The write value should be 0.
b25, b24	RSM3[1:0]	Receive Synchronous Mode (Receive Queue 3+i*4) B'00: Mode with write-back Other than B'00: Setting prohibited
b23, b22	—	Reserved These bits are read as 0. The write value should be 0.
b21, b20	UFCC2[1:0]	Unread Frame Counter Configuration (Receive Queue 2+i*4) These bits set the unread frame counter used in reception queue 2+4*i.
b19, b18	—	Reserved These bits are read as 0. The write value should be 0.
b17, b16	RSM2[1:0]	Receive Synchronous Mode (Receive Queue 2+i*4) B'00: Mode with write-back Other than B'00: Setting prohibited
b15, b14	—	Reserved These bits are read as 0. The write value should be 0.
b13, b12	UFCC1[1:0]	Unread Frame Counter Configuration (Receive Queue 1+i*4) These bits set the unread frame counter used in reception queue 1+4*i.
b11, b10	—	Reserved These bits are read as 0. The write value should be 0.

Table 24.17 RQCi register contents (2/2)

Bit Position	Bit Name	Function
b9, b8	RSM1[1:0]	Receive Synchronous Mode (Receive Queue 1+i*4) B'00: Mode with write-back Other than B'00: Setting prohibited
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5, b4	UFCC0[1:0]	Unread Frame Counter Configuration (Receive Queue 0+i*4) These bits set the unread frame counter used in reception queue 0+4*i.
b3, b2	—	Reserved These bits are read as 0. The write value should be 0.
b1, b0	RSM0[1:0]	Receive Synchronous Mode (Receive Queue 0+i*4) B'00: Mode with write-back Other than B'00: Setting prohibited

UFCCr[1:0] (r = 0 to 17) Unread Frame Counter Configuration Bits

These bits set the unread frame counter for reception queue r.

With the AVB-DMAC, four patterns of settings are available for the unread frame counter. Use the unread frame counter warning level configuration register (UFCW) and the unread frame counter stop level configuration register (UFCS) to set the warning level and stop level of the unread frame counter.

Set the pattern number (0 to 3) set in the unread frame counter warning level configuration register (UFCW) and unread frame counter stop level configuration register (UFCS) in this bit.

When the value is B'00, the stop function is disabled.

Writing to the bits is only possible when the current operating mode is configuration mode.

RSMr[1:0] (r = 0 to 17) Receive Synchronous Mode Bits

These bits set receive synchronous mode.

Set B'00 in this bit.

For receive synchronous mode, see **Section 24.4.4.3** (3) Mode with Write-Back.

Writing to the bits is only possible when the current operating mode is configuration mode.

24.3.9 Receive Padding Configuration Register (RPC)

The RPC register is used to set padding for received frames.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $0B0_H$

Initial value: 0000 0100_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DCNT[7:0]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	PCNT[2:0]			—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 24.18 RPC register contents

Bit Position	Bit Name	Function
b31 to b24	—	Reserved These bits are read as 0. The write value should be 0.
b23 to b16	DCNT[7:0]	Stored Data Counter These bits specify the amount of data to be stored with the descriptor. The setting is in words. I.e. 1 in the counter indicates 1 word (4 bytes).
b15 to b11	—	Reserved These bits are read as 0. The write value should be 0.
b10 to b8	PCNT[2:0]	Stored Padding Counter These bits indicate the amount of padding to be stored in data areas for descriptors. The setting is in words. I.e. 1 in the counter indicates 1 word (4 bytes).
b7 to b0	—	Reserved These bits are read as 0. The write value should be 0.

CAUTION

Padding can be used to extend frame lengths, but frame lengths should not exceed 4 Kbytes.

DCNT[7:0] Stored Data Counter Bits

These bits specify the amount of the frame data (1 to 255) to be stored following the padding. Counting by one indicates one word (4 bytes). For example, when these bits are set to 47, the amount of data is 47 words (= 188 bytes).

When these bits are 0, all received data have been stored following the initial padding.

Writing to the bits is only possible when the current operating mode is configuration mode.

For details on padding, see **Section 24.4.4.3** (c) Padding.

PCNT[2:0] Stored Padding Counter Bits

These bits specify the amount of padding to be appended to the URAM. Counting by one indicates one word (4 bytes). For example, when these bits are set to 1, the amount of padding is one word (= 4 bytes).

Writing to the bits is only possible when the current operating mode is configuration mode.

For details on padding, see **Section 24.4.4.3** (c) Padding.

24.3.10 Unread Frame Counter Warning Level Configuration Register (UFCW)

The UFCW register sets the warning levels for the number of unread frames. Four levels are available and the RQC0 to RQC4 receive queue configuration registers (RQC0 to RQC4.UFFC0 to UFFC3) refer to the selected levels.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $0BC_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	WL3[5:0]						—	—	WL2[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	WL1[5:0]						—	—	WL0[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.19 UFCW register contents

Bit Position	Bit Name	Function
b31, b30	—	Reserved These bits are read as 0. The write value should be 0.
b29 to b24	WL3[5:0]	Warning Level 3 Unread frame count warning level 3
b23, b22	—	Reserved These bits are read as 0. The write value should be 0.
b21 to b16	WL2[5:0]	Warning Level 2 Unread frame count warning level 2
b15, b14	—	Reserved These bits are read as 0. The write value should be 0.
b13 to b8	WL1[5:0]	Warning Level 1 Unread frame count warning level 1
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5 to b0	WL0[5:0]	Warning Level 0 Unread frame count warning level 0

WL0 to WL3[5:0] Warning Level 0 to 3 Bits

These bits set the warning levels for unread frames.

One of the four warning levels from 0 to 3 can be set for each reception queue. When these bits are set to 0, the stop function is disabled. The level to be used is specified by the receive queue configuration register i (RQCi) (i = 0 to 4).

Writing to the bits is only possible when the current operating mode is configuration mode.

24.3.11 Unread Frame Counter Stop Level Configuration Register (UFCS)

The UFCS register sets the stop levels for unread frames.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $0C0_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	SL3[5:0]						—	—	SL2[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	SL1[5:0]						—	—	SL0[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 24.20 UFCS register contents

Bit Position	Bit Name	Function
b31, b30	—	Reserved These bits are read as 0. The write value should be 0.
b29 to b24	SL3[5:0]	Stop Level 3 Unread frame count stop level 3
b23, b22	—	Reserved These bits are read as 0. The write value should be 0.
b21 to b16	SL2[5:0]	Stop Level 2 Unread frame count stop level 2
b15, b14	—	Reserved These bits are read as 0. The write value should be 0.
b13 to b8	SL1[5:0]	Stop Level 1 Unread frame count stop level 1
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5 to b0	SL0[5:0]	Stop Level 0 Unread frame count stop level 0 The write value should be 0.

SL0 to SL3[5:0] Stop Level 0 to 3 Bits

These bits set the stop levels for unread frames.

One of the four stop levels from 0 to 3 can be set for each reception queue. When these bits are set to 0, the stop function is disabled. The level to be used is specified by the receive queue configuration register i (RQCi) (i = 0 to 4).

Writing to the bits is only possible when the current operating mode is configuration mode.

24.3.12 Unread Frame Counter Register i (UFCVi) (i = 0 to 4)

The UFCV0 register indicates the number of unread frames in reception queues 0 to 3.

The UFCV1 register indicates the number of unread frames in reception queues 4 to 7.

The UFCV2 register indicates the number of unread frames in reception queues 8 to 11.

The UFCV3 register indicates the number of unread frames in reception queues 12 to 15.

The UFCV4 register indicates the number of unread frames in reception queues 16 and 17.

Access: This register can be read in 32-bit units.

Address: UFCV0: <ETNBn_base> + $0C4_H$ UFCV1: <ETNBn_base> + $0C8_H$ UFCV2: <ETNBn_base> + $0CC_H$ UFCV3: <ETNBn_base> + $0D0_H$
UFCV4: <ETNBn_base> + $0D4_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	CV3[5:0]						—	—	CV2[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CV1[5:0]						—	—	CV0[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.21 UFCVi register contents

Bit Position	Bit Name	Function
b31, b30	—	Reserved These bits are read as 0. The write value should be 0.
b29 to b24	CV3[5:0]	Unread Frame Count 3+4*i Number of unread frames in reception queue 3+4*i
b23, b22	—	Reserved These bits are read as 0. The write value should be 0.
b21 to b16	CV2[5:0]	Unread Frame Count 2+4*i Number of unread frames in reception queue 2+4*i
b15, b14	—	Reserved These bits are read as 0. The write value should be 0.
b13 to b8	CV1[5:0]	Unread Frame Count 1+4*i Number of unread frames in reception queue 1+4*i
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5 to b0	CV0[5:0]	Unread Frame Count 0+4*i Number of unread frames in reception queue 0+4*i

CVr[5:0] Unread Frame Count r (r = 0 to 17) Bits

These bits indicate the number of unread frames in reception queue r.

The number of unread frames is decremented by the value that is written to the unread frame counter decrement register i (UFCDi).

For a description of how to use unread frames, refer to Section 24.4.4.4, Unread Frame Counters.

Conditions for updating:

The bits are set to 0 when the operating mode is not operation mode and when the descriptor base address load request register (DLR) issues a base address load request.

The number is incremented when data received in reception queue r are stored normally. The maximum increment is H’3F. If the value exceeds H’3F, incrementation will not proceed.)

The number is decremented by the value written to the unread frame counter decrement register i (UFCDi).

24.3.13 Unread Frame Counter Decrement Register i (UFCDi) (i = 0 to 4)

The UFCD0 register is used to decrement unread counters in reception queues 0 to 3.

The UFCD1 register is used to decrement unread counters in reception queues 4 to 7.

The UFCD2 register is used to decrement unread counters in reception queues 8 to 11.

The UFCD3 register is used to decrement unread counters in reception queues 12 to 15.

The UFCD4 register is used to decrement unread counters in reception queues 16 and 17.

Access: This register can be read/written in 8/16/32-bit units.

Address: UFCD0: <ETNBn_base> + $0E0_H$ UFCD1: <ETNBn_base> + $0E4_H$ UFCD2: <ETNBn_base> + $0E8_H$
UFCD3: <ETNBn_base> + $0EC_H$ UFCD4: <ETNBn_base> + $0F0_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	DV3[5:0]						—	—	DV2[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	DV1[5:0]						—	—	DV0[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.22 UFCDi register contents

Bit Position	Bit Name	Function
b31, b30	—	Reserved These bits are read as 0. The write value should be 0.
b29 to b24	DV3[5:0]	Unread Frame Decrement Value 3+4*i Unread frame decrement value for reception queue 3+4*i
b23, b22	—	Reserved These bits are read as 0. The write value should be 0.
b21 to b16	DV2[5:0]	Unread Frame Decrement Value 2+4*i Unread frame decrement value for reception queue 2+4*i
b15, b14	—	Reserved These bits are read as 0. The write value should be 0.
b13 to b8	DV1[5:0]	Unread Frame Decrement Value 1+4*i Unread frame decrement value for reception queue 1+4*i
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5 to b0	DV0[5:0]	Unread Frame Decrement Value 0+4*i Unread frame decrement value for reception queue 0+4*i

DVr[5:0] Unread Frame Decrement Value r (r = 0 to 17) Bits

These bits set the decrement value for unread frames in reception queue r. The value of an unread frame counter register i (UFCVi) (i = 0 to 4) is decremented by the value set in the corresponding bits of this register.

Write H'3F to these bits to reset the unread counters in reception queue r.

These bits are always read as 0.

24.3.14 Separation Filter Offset Register (SFO)

The SFO register sets an offset into frames for use by the separation filter.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $0FC_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	FBP[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.23 SFO register contents

Bit Position	Bit Name	Function
b31 to b6	—	Reserved These bits are read as 0. The write value should be 0.
b5 to b0	FBP[5:0]	First Byte Position Position in Ethernet frames of the first byte of the bytes to be used by the separation filter

FBP[5:0] First Byte Position Bits

These bits set the position in Ethernet frames of the first byte of the bytes to be used by the separation filter.

When these bits are 0, the separation filter starts from the start of each Ethernet frame (first byte of the destination address). For bytes in Ethernet frames, see Figure 24.2, Data for Reception in an Ethernet Frame, in Section 24.3.1, AVB-DMAC Mode Register (CCC).

Writing to the bits is only possible when the current operating mode is configuration mode.

For separation filtering, see **Section 24.4.4.1** (1) Separation Filtering.

CAUTION

Received frames having fewer bytes than the setting of these bits + 8 bytes are judged to be non-matching by the separation filter. In this case, the data will either be sorted into a reception queue or discarded in accord with the setting of the separation filtering select bits in the receive configuration register (RCR.ESF[1:0]).

24.3.15 Separation Filter Pattern Register i (SFPi)

Pairs of SFPi registers set the pattern for the separation filters to be used by the corresponding reception queues 2 to 17.

Access: This register can be read/written in 32-bit units.

Address: SFP0: <ETNBn_base> + 100_H; SFP1: <ETNBn_base> + 104_H; SFP2: <ETNBn_base> + 108_H
SFP3: <ETNBn_base> + $10C_H$; SFP4: <ETNBn_base> + 110_H; SFP5: <ETNBn_base> + 114_H
SFP6: <ETNBn_base> + 118_H; SFP7: <ETNBn_base> + $11C_H$; SFP8: <ETNBn_base> + 120_H
SFP9: <ETNBn_base> + 124_H; SFP10: <ETNBn_base> + 128_H; SFP11: <ETNBn_base> + $12C_H$
SFP12: <ETNBn_base> + 130_H; SFP13: <ETNBn_base> + 134_H; SFP14: <ETNBn_base> + 138_H
SFP15: <ETNBn_base> + $13C_H$; SFP16: <ETNBn_base> + 140_H; SFP17: <ETNBn_base> + 144_H
SFP18: <ETNBn_base> + 148_H; SFP19: <ETNBn_base> + $14C_H$; SFP20: <ETNBn_base> + 150_H
SFP21: <ETNBn_base> + 154_H; SFP22: <ETNBn_base> + 158_H; SFP23: <ETNBn_base> + $15C_H$
SFP24: <ETNBn_base> + 160_H; SFP25: <ETNBn_base> + 164_H; SFP26: <ETNBn_base> + 168_H
SFP27: <ETNBn_base> + $16C_H$; SFP28: <ETNBn_base> + 170_H; SFP29: <ETNBn_base> + 174_H;
SFP30: <ETNBn_base> + 178_H; SFP31: <ETNBn_base> + $17C_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FP[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.24 SFPi register contents

Bit Position	Bit Name	Function
b31 to b0	FP[31:0]	Separation Filter Pattern These bits set the pattern of the separation filter. The 64-bit filter pattern is set for each queue.

FP[63:0] Separation Filter Pattern Bits

These bits set the pattern for a separation filter to be used with reception queues 2 to 17 (for streams 0 to 15).

Each queue shares a 64-bit setting; reception queue 2 (for stream 0) uses SFP0 and SFP1, reception queue 17 (for stream 15) uses SFP30 and SFP31, and so on.

The separation filter passes a frame when, after masking by the mask value set in the separation filter mask register (SFMi), data from received frames match the value defined in these bits.

SFPi.FP[7:0] (where i is an even number) are used for the byte of Ethernet frame data specified by the separation filter offset register, while SFPi.FP[63:56] (where i is the corresponding odd number) are used for the byte at the address specified by the separation filter offset register (SFO) + 7.

Writing to the bits is only possible when the current operating mode is configuration mode.

For separation filtering, see **Section 24.4.4.1** (1) Separation Filtering.

24.3.16 Separation Filter Mask Register i (SFMi) (i = 0 or 1)

A pair of SFMi registers sets the mask value for the separation filter used by the corresponding reception queue 2 to 17.

Access: This register can be read/written in 32-bit units.

Address: SFM0: <ETNBn_base> + $1C0_H$ SFM1: <ETNBn_base> + $1C4_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CFM[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CFM[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.25 SFMi register contents

Bit Position	Bit Name	Function
b31 to b0	CFM[31:0]	Separation Filter Mask These bits set the mask value for the separation filter.

CFM[63:0] Separation Filter Mask Bits

These bits set the mask value for the separation filter for use with the corresponding reception queue 2 to 17 (stream 0 to 15).

SFM0.CFM[7:0] are used for bytes of Ethernet frame data specified by the separation filter offset register, while SFM1.CFM[63:56] are used for the separation filter offset register (SFO) + 7.

Frame data at the positions of mask bits that are set to 0 are masked; that is, they do not affect pattern-matching by the separation filter.

Writing to the bits is only possible when the current operating mode is configuration mode.

For separation filtering, see **Section 24.4.4.1** (1) Separation Filtering.

24.3.17 Transmit Configuration Register (TGC)

The TGC register is used to make settings related to transmission for the AVB-DMAC.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 300_H

Initial value: 0022 2200_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	TBD3[1:0]		—	—	TBD2[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	TBD1[1:0]		—	—	TBD0[1:0]		—	—	TQP[1:0]		TSM3	TSM2	TSM1	TSM0
Value after reset	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.26 TGC register contents (1/2)

Bit Position	Bit Name	Function
b31 to b22	—	Reserved These bits are read as 0. The write value should be 0.
b21, b20	TBD3[1:0]	Transmit FIFO Size (Stream Class A) Number of frames to be fetched from transmission queue 3 (for stream class A) **CAUTION** Write 2 to these bits.
b19, b18	—	Reserved These bits are read as 0. The write value should be 0.
b17, b16	TBD2[1:0]	Transmit FIFO Size (Stream Class B) Number of frames to be fetched from transmission queue 2 (for stream class B) **CAUTION** Write 2 to these bits.
b15, b14	—	Reserved These bits are read as 0. The write value should be 0.
b13, b12	TBD1[1:0]	Transmit FIFO Size (Network Control) Number of frames to be fetched from transmission queue 1 (for network control) **CAUTION** Write 2 to these bits.
b11, b10	—	Reserved These bits are read as 0. The write value should be 0.
b9, b8	TBD0[1:0]	Transmit FIFO Size (Best Effort) Number of frames to be fetched from transmission queue 0 (for best effort) **CAUTION** Write 2 to these bits.

Table 24.26 TGC register contents (2/2)

Bit Position	Bit Name	Function
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5, b4	TQP[1:0]	Transmit Queue Priority 00: Non-AVB mode 01: AVB mode 1 10: Setting prohibited 11: AVB mode 2
b3	TSM3	Transmit Synchronous Mode (Stream Class A) 0: With write-back 1: Setting prohibited
b2	TSM2	Transmit Synchronous Mode (Stream Class B) 0: With write-back 1: Setting prohibited
b1	TSM1	Transmit Synchronous Mode (Network Control) 0: With write-back 1: Setting prohibited
b0	TSM0	Transmit Synchronous Mode (Best Effort) 0: With write-back 1: Setting prohibited

TBD0 to TBD3[1:0] Transmit FIFO Size (Stream Class A/ Stream Class B/Network Control/Best Effort) Bits

These bits set the sizes of the transmission FIFO buffers for use with each of the transmission queues.

Writing to these bits is only possible when the current operating mode is configuration mode.

Set these bits to 2.

TQP[1:0] Transmit Queue Priority Bits

These bits set the priority of the transmission queues.

B’00: Non-AVB mode: Q3→Q2→Q1→Q0

B’01: AVB mode 1: Q3 (CBS)→Q2 (CBS)→Q1→Q0

B’10: Setting prohibited

B’11: AVB mode 2: Q1→Q3 (CBS)→Q2 (CBS)→Q0

For the credit-based shaping (CBS) algorithm, see Section 24.4.6, CBS (Credit-Based Shaping).

The CBS algorithm is not applied in non-AVB mode (i.e. when the value is B’00).

Writing to the bits is only possible when the current operating mode is configuration mode.

TSM0 to TSM3 Transmit Synchronous Bits

Set these bits to 0.

24.3.18 Transmit Configuration Control Register (TCCR)

The TCCR register controls transmission by the AVB-DMAC and is used to make related settings.

Access: This register can be read/written in 8/16/32-bit units.

Address: <ETNBn_base> + 304_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TFR	TFEN	—	—	—	—	TSRQ3	TSRQ2	TSRQ1	TSRQ0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 24.27 TCCR register contents

Bit Position	Bit Name	Function
b31 to b10	—	Reserved These bits are read as 0. The write value should be 0.
b9	TFR	Time Stamp FIFO Release 0: (Not operating) 1: Releases the oldest entry in the time-stamp FIFO.
b8	TFEN	Time Stamp FIFO Enable 0: Recording of transmission time stamps in the time-stamp FIFO is disabled. 1: Recording of transmission time stamps in the time-stamp FIFO is enabled.
b7 to b4	—	Reserved These bits are read as 0. The write value should be 0.
b3	TSRQ3	Transmit Start Request (Queue 3 (Stream Class A)) 0: Transmission queue is empty or stopped. 1: When written: A transmission start request is issued. When read: Fetching of data for transmission is pending.
b2	TSRQ2	Transmit Start Request (Queue 2 (Stream Class B)) 0: Transmission queue is empty or stopped. 1: When written: A transmission start request is issued. When read: Fetching of data for transmission is pending.
b1	TSRQ1	Transmit Start Request (Queue 1 (Network Control)) 0: Transmission queue is empty or stopped. 1: When written: A transmission start request is issued. When read: Fetching of data for transmission is pending.
b0	TSRQ0	Transmit Start Request (Queue 0 (Best Effort)) 0: Transmission queue is empty or stopped. 1: When written: A transmission start request is issued. When read: Fetching of data for transmission is pending.

TFR Time Stamp FIFO Release Bit

This bit releases the oldest entry in the time-stamp FIFO.

For a description of how to use the time-stamp FIFO, see Section 24.4.5.4, Time Stamping in Transmission.

TFEN Time Stamp FIFO Enable Bit

This bit enables storage in the time-stamp FIFO.

When it is set, time-stamp information is stored for descriptors with DESCR.TSR set to 1 (for DESCR.TSR, see **Section 24.4.5.2** (2) Configuration of Transmission Frame Data Descriptors.

When 0 is set in this bit, no entries are made in the time-stamp FIFO.

For a description of how to use the time-stamp FIFO, see section 16.3.5.4, Time Stamping in Transmission.

TSRQt Transmit Start Request (Queue t) (t = 0 to 3) Bit

This bit issues a request to start transmission for transmission queue t.

When read, this bit being set to 1 indicates that transmission queue t has a frame that has not yet been fetched to the transmission FIFO.

Frame transmission by the E-MAC is processed independently from fetching to the transmission FIFO. The timing of transmission from a queue depends on the priority order of transmission.

For the scheduling of transmission queues, see Section 24.4.5.1, Transmission Modes.

Writing to this bit is only possible when the current operating mode is configuration mode.

Only 1 can be written to the bit. Writing 0 to the bit has no effect.

Conditions for updating:

The bit is set to 0 when the operating mode is not operation mode, when a descriptor of type FEMPTY or LEMPTY (no usable data) is processed, when an EOS descriptor is processed, and when a descriptor with defective data is processed.

24.3.19 Transmit Status Register (TSR)

The TSR register indicates the state of transmission by the AVB-DMAC.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 308_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TFFL[2:0]			—	—	—	—	CCS1[1:0]		CCS0[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.28 TSR register contents

Bit Position	Bit Name	Function
b31 to b11	—	Reserved These bits are read as 0. The write value should be 0.
b10 to b8	TFFL[2:0]	Time Stamp FIFO Count Number of time-stamp FIFOs
b7 to b4	—	Reserved These bits are read as 0. The write value should be 0.
b3, b2	CCS1[1:0]	CBS Counter Status 1 (Class A) B'00: The current credit value is within the limit. B'01: The current credit value is less than or equal to the lower limit. B'10: The current credit value is greater than or equal to the upper limit. B'11: (Reserved)
b1, b0	CCS0[1:0]	CBS Counter Status 0 (Class B) B'00: The current credit value is within the limit. B'01: The current credit value is less than or equal to the lower limit. B'10: The current credit value is greater than or equal to the upper limit. B'11: (Reserved)

TFFL[3:0] Time Stamp FIFO Count Bits

These bits indicate the number of time stamps in the time-stamp FIFO.

The values 0 and 2 indicate that the time-stamp FIFO is empty and full, respectively (values 3 to 7 are reserved).

Conditions for updating:

The bits are set to 0 when the operating mode is not operation mode and when the time stamp FIFO enable bit in the transmit configuration control register (TCCR.TFEN) = 0.

When the time stamp FIFO enable bit (TCCR.TFEN) is 1 and these bits are not 2, the value of these bits is incremented after a frame with DESCR.TSR set has been transmitted by the E-MAC (for DESCR.TSR, see **Section 24.4.5.2** (2) Configuration of Transmission Frame Data Descriptors.

The value of these bits is decremented if it is not 0 when 1 is written to the time stamp FIFO release bit in the transmit configuration control register (TCCR.TFR).

CCS0 and CCS1[1:0] CBS Counter Status 0 and 1 Bits

These bits indicate the CBS (credit-based shaping) state of stream data transmission queues 0 and 1. If the calculated credit value is outside the range specified by CBS upper limit register c (CULc) and CBS lower limit register c (CLLc), it falls outside the range for CBS.

Conditions for updating:

The bits are set to B’00 when the operating mode is not operation mode.

The bits are set to B’01 if the credit value calculated by the CBS is lower than the value in CBS lower limit register c (CLLc).

The bits are set to B’10 if the credit value calculated by the CBS is higher than the value in CBS upper limit register c (CULc).

24.3.20 Time Stamp FIFO Access Register 0 (TFA0)

TFA0 indicates the seconds portion of the time stamp value.

Access: This register is read-only in 32-bit units.

Address: <ETNBn_base> + 310_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.29 TFA0 register contents

Bit Position	Bit Name	Function
b31 to b0	TSV[31:0]	Time Stamp Value Time stamp value

TSV[79:0] Time Stamp Value Bits

These 80 bits consist of TFA0.TSV[31:0], TFA1.TSV[63:32], and TFA2.TSV[79:64], which together indicate the oldest time stamp value stored in the time-stamp FIFO.

Once the time-stamp FIFO is full, no further time-stamp values are stored.

Conditions for updating:

The bits are set to H'0000 0000 when the operating mode is not operation mode.

The register is updated whenever a value is stored in the time-stamp FIFO (when the time-stamp FIFO count bit in the transmit status register (TSR.TFFL[2:0]) changes from 0 to 1).

The register is updated when the oldest entry is released (when the time stamp FIFO release bit in the transmit configuration control register (TCCR.TFR) is set to 1).

24.3.21 Time Stamp FIFO Access Register 1 (TFA1)

The TFA1 register indicates the nanoseconds portion of the time-stamp value.

Access: This register is read-only in 32-bit units.

Address: <ETNBn_base> + 314_H

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TSV[63:48]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSV[47:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.30 TFA1 register contents

Bit Position	Bit Name	Function
b31 to b0	TSV[63:32]	Time Stamp Value Time-stamp value

TSV[63:32] Time Stamp Value Bits

For details, see Section 24.3.20, Time Stamp FIFO Access Register 0 (TFA0).

24.3.22 Time Stamp FIFO Access Register 2 (TFA2)

The TFA2 register indicates the time-stamp tag and part of the time-stamp value.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 318_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	TST[9:0]									
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSV[79:64]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.31 TFA2 register contents

Bit Position	Bit Name	Function
b31 to b26	—	Reserved These bits are read as 0.
b25 to b16	TST[9:0]	Time Stamp Tag Time-stamp tag
b15 to b0	TSV[79:64]	Time Stamp Value Time-stamp value

TST[9:0] Time Stamp Tag Bits

These bits indicate the contents of the DESCR.TAG bit within the descriptor for frame transmission. These values are used to check the correlation between frames within the transmission queue and the time-stamp values (accessible through time stamp FIFO access register i (TFAi)) which can be placed in the FIFO.

For the tagging of frames in transmission, Section 24.4.5.4, Time Stamping in Transmission.

Conditions for updating:

The bits are set to H’000 when the operating mode is not operation mode.

Updated when a value is stored in the time-stamp FIFO (when the value of the time stamp FIFO count bit in the transmit status register (TSR.TFFL[2:0]) changes from 0 to 1).

Updated when the oldest entry has been released (1 is set in the time stamp FIFO release bit in the transmit configuration control register (TCCR.TFR)).

TSV[79:64] Time Stamp Value Bits

For details, see Section 24.3.20, Time Stamp FIFO Access Register 0 (TFA0).

24.3.23 CBS Increment Value Register c (CIVRc) (c = 0 or 1)

The CIVR0 register sets the increment in the CBS algorithm for transmission queue 2 (for stream class B).

The CIVR1 register sets the increment in the CBS algorithm for transmission queue 3 (for stream class A).

Access: This register can be read/written in 32-bit units.

Address: CIVR0: <ETNBn_base> + 320_H CIVR1: <ETNBn_base> + 324_H

Initial value: $0000\ 0001_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CIV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CIV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.32 CIVRc register contents

Bit Position	Bit Name	Function
b31 to b0	CIV[31:0]	CBS Increment Value CBS increment value (1 to H'FFFF)

CIV[31:0] CBS Increment Value Bits

These bits set the increment for the CBS algorithm.

Set a value in the range from H'0000 0001 to H'0000 FFFF.

The value to be written to these bits depends on the Ethernet bit rate and fCHI (peripheral bus clock). For details, see Section 24.4.6, CBS (Credit-Based Shaping).

24.3.24 CBS Decrement Value Register c (CDVRc) (c = 0 or 1)

The CDVR0 register sets the decrement in the CBS algorithm for transmission queue 2 (for stream class B).

The CDVR1 register sets the decrement in the CBS algorithm for transmission queue 3 (for stream class A).

Access: This register can be read/written in 32-bit units.

Address: CDVR0: <ETNBn_base> + 328_H CDVR1: <ETNBn_base> + $32C_H$

Initial value: FFFF $FFFF_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CDV[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CDV[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.33 CDVR0 and CDVR1 register contents

Bit Position	Bit Name	Function
b31 to b0	CDV[31:0]	CBS Decrement Value Setting value: -1 to -65536 (H'FFFF FFFF to H'FFFF 0000)

CDV[31:0] CBS Decrement Value Bits

These bits set the decrement for the CBS algorithm.

Set a negative value from -1 to -65536 (H'FFFF FFFF to H'FFFF 0000).

The value to be written to these bits depends on the Ethernet bit rate and fCHI (peripheral bus clock). For details, see Section 24.4.6, CBS (Credit-Based Shaping).

24.3.25 CBS Upper Limit Register c (CULc) (c = 0 or 1)

The CUL0 register sets the upper limit for credit values calculated by using the CSB algorithm for transmission queue 2 (for stream class B).

The CUL1 register sets the upper limit for credit values calculated by using the CSB algorithm for transmission queue 3 (for stream class A).

Access: This register can be read/written in 32-bit units.

Address: CUL0: <ETNBn_base> + 330_H CUL1: <ETNBn_base> + 334_H

Initial value: 7FFF FFFF$_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	ULV[31:16]															
Value after reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ULV[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.34 CUL0 and CUL1 register contents

Bit Position	Bit Name	Function
b31 to b0	ULV[31:0]	CBS Upper Limit Upper limit on CBS values

ULV[31:0] CBS Upper Limit Bits

These bits set the upper limit for credit values calculated by using the CBS algorithm.

The setting is a limiting value for error detection and does not normally affect operation of the algorithm.

Write a positive value to these bits.

For details, see Section 24.4.6, CBS (Credit-Based Shaping).

24.3.26 CBS Lower Limit Register c (CLLc) (c = 0 or 1)

The CUL0 register sets the lower limit for credit values calculated by using the CSB algorithm for transmission queue 2 (for stream class B).

The CUL1 register sets the lower limit for credit values calculated by using the CSB algorithm for transmission queue 3 (for stream class A).

Access: This register can be read/written in 32-bit units.

Address: CLL0: <ETNBn_base> + 338_H CLL1: <ETNBn_base> + $33C_H$

Initial value: $8000\ 0001_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	LLV[31:16]															
Value after reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LLV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.35 CLLc register contents

Bit Position	Bit Name	Function
b31 to b0	LLV[31:0]	CBS Lower Limit Lower limit on CBS values

LLV[31:0] CBS Lower Limit Bits

These bits set the lower limit for credit values calculated by using the CBS algorithm.

The setting is a limiting value for error detection and does not normally affect operation of the algorithm.

Write a negative value to these bits.

For details, see Section 24.4.6, CBS (Credit-Based Shaping).

24.3.27 Descriptor Interrupt Control Register (DIC)

The DIC register is used to control descriptor interrupts 1 to 15.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 350_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DPE15	DPE14	DPE13	DPE12	DPE11	DPE10	DPE9	DPE8	DPE7	DPE6	DPE5	DPE4	DPE3	DPE2	DPE1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 24.36 DIC register contents (1/2)

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15	DPE15	Descriptor Interrupt Enable 15 0: Disabled 1: Enabled
b14	DPE14	Descriptor Interrupt Enable 14 0: Disabled 1: Enabled
b13	DPE13	Descriptor Interrupt Enable 13 0: Disabled 1: Enabled
b12	DPE12	Descriptor Interrupt Enable 12 0: Disabled 1: Enabled
b11	DPE11	Descriptor Interrupt Enable 11 0: Disabled 1: Enabled
b10	DPE10	Descriptor Interrupt Enable 10 0: Disabled 1: Enabled
b9	DPE9	Descriptor Interrupt Enable 9 0: Disabled 1: Enabled
b8	DPE8	Descriptor Interrupt Enable 8 0: Disabled 1: Enabled
b7	DPE7	Descriptor Interrupt Enable 7 0: Disabled 1: Enabled
b6	DPE6	Descriptor Interrupt Enable 6 0: Disabled 1: Enabled

Table 24.36 DIC register contents (2/2)

Bit Position	Bit Name	Function
b5	DPE5	Descriptor Interrupt Enable 5 0: Disabled 1: Enabled
b4	DPE4	Descriptor Interrupt Enable 4 0: Disabled 1: Enabled
b3	DPE3	Descriptor Interrupt Enable 3 0: Disabled 1: Enabled
b2	DPE2	Descriptor Interrupt Enable 2 0: Disabled 1: Enabled
b1	DPE1	Descriptor Interrupt Enable 1 0: Disabled 1: Enabled
b0	—	Reserved This bit is read as 0. The write value should be 0.

RPE1 to RPE15 Descriptor Interrupt Enable Bits 1 to 15

When an interrupt source flag is set (a bit from among the DPF1 to DPF15 bits in the descriptor interrupt status register (DIS) = 1) while the interrupt is enabled, the interrupt is issued.

24.3.28 Descriptor Interrupt Status Register (DIS)

The DIS register indicates the state of descriptor interrupts.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 354_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DPF15	DPF14	DPF13	DPF12	DPF11	DPF10	DPF9	DPF8	DPF7	DPF6	DPF5	DPF4	DPF3	DPF2	DPF1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 24.37 DIS register contents (1/2)

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15	DPF15	Descriptor Interrupt Status15 0: The interrupt is not pending. 1: The interrupt is pending.
b14	DPF14	Descriptor Interrupt Status14 0: The interrupt is not pending. 1: The interrupt is pending.
b13	DPF13	Descriptor Interrupt Status13 0: The interrupt is not pending. 1: The interrupt is pending.
b12	DPF12	Descriptor Interrupt Status12 0: The interrupt is not pending. 1: The interrupt is pending.
b11	DPF11	Descriptor Interrupt Status11 0: The interrupt is not pending. 1: The interrupt is pending.
b10	DPF10	Descriptor Interrupt Status10 0: The interrupt is not pending. 1: The interrupt is pending.
b9	DPF9	Descriptor Interrupt Status9 0: The interrupt is not pending. 1: The interrupt is pending.
b8	DPF8	Descriptor Interrupt Status8 0: The interrupt is not pending. 1: The interrupt is pending.
b7	DPF7	Descriptor Interrupt Status7 0: The interrupt is not pending. 1: The interrupt is pending.
b6	DPF6	Descriptor Interrupt Status6 0: The interrupt is not pending. 1: The interrupt is pending.

Table 24.37 DIS register contents (2/2)

Bit Position	Bit Name	Function
b5	DPF5	Descriptor Interrupt Status5 0: The interrupt is not pending. 1: The interrupt is pending.
b4	DPF4	Descriptor Interrupt Status4 0: The interrupt is not pending. 1: The interrupt is pending.
b3	DPF3	Descriptor Interrupt Status3 0: The interrupt is not pending. 1: The interrupt is pending.
b2	DPF2	Descriptor Interrupt Status2 0: The interrupt is not pending. 1: The interrupt is pending.
b1	DPF1	Descriptor Interrupt Status1 0: The interrupt is not pending. 1: The interrupt is pending.
b0	—	Reserved This bit is read as 0. The write value should be 0.

DPF1 to DPF15 Descriptor Interrupt Status Bits

When DESCR.DIE is 1 to 15, the corresponding bit indicates completion of the processing of a descriptor within the reception or transmission queue.

When DESCR.DIE is 0, the descriptor interrupt is not generated.

Only 0 can be written to these bits.

[Conditions for Changing]

A bit is set to 0 when the operating mode is not operation mode.

A bit is set to 1 when a descriptor with DESCR.DIE set to the corresponding number from 1 to 15 is processed.

24.3.29 Error Interrupt Control Register (EIC)

The EIC register controls the AVB-DMAC-related error interrupts.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 358_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TFFE	CULE1	CULE0	CLLE1	CLLE0	SEE	QEE	MTEE	MREE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.38 EIC register contents

Bit Position	Bit Name	Function
b31 to b9	—	Reserved These bits are read as 0. The write value should be 0.
b8	TFFE	Time Stamp FIFO Full-Error Interrupt Enable 0: Disabled 1: Enabled
b7	CULE1	CBS Upper Limit Error Interrupt Enable (Class A) 0: Disabled 1: Enabled
b6	CULE0	CBS Upper Limit Error Interrupt Enable (Class B) 0: Disabled 1: Enabled
b5	CLLE1	CBS Lower Limit Error Interrupt Enable (Class A) 0: Disabled 1: Enabled
b4	CLLE0	CBS Lower Limit Error Interrupt Enable (Class B) 0: Disabled 1: Enabled
b3	SEE	Separation Error interrupt Enable 0: Disabled 1: Enabled
b2	QEE	Queue Error interrupt Enable 0: Disabled 1: Enabled
b1	MTEE	MAC Transmission Error interrupt Enable 0: Disabled 1: Enabled
b0	MREE	MAC Reception Error interrupt Enable 0: Disabled 1: Enabled

TFFE Time Stamp FIFO Full-Error Interrupt Enable Bits

When the time stamp FIFO is full (TFFF in the error interrupt status register (EIS) = 1) and the interrupt is enabled, the interrupt is issued.

CULE1 CBS Upper Limit Error Interrupt Enable Bit (Class A)

When the Class A CBS reaches its upper limit (CULF1 in the error interrupt status register (EIS) = 1), the interrupt is issued.

CULE0 CBS Upper Limit Error Interrupt Enable Bit (Class B)

When the Class B CBS reaches its upper limit (CULF0 in the error interrupt status register (EIS) = 1), the interrupt is issued.

CLLE1 CBS Lower Limit Error Interrupt Enable Bit (Class A)

When the Class A CBS reaches its lower limit (CLLF1 in the error interrupt status register (EIS) = 1), the interrupt is issued.

CLLE0 CBS Lower Limit Error Interrupt Enable Bit (Class B)

When the Class B CBS reaches its lower limit (CLLF0 in the error interrupt status register (EIS) = 1), the interrupt is issued.

SEE Separation Error interrupt Enable Bit

While this bit is 1 an interrupt will be generated when EIS.SEF is 1.

QEE Queue Error interrupt Enable Bit

While this bit is 1 an interrupt will be generated when EIS.QEF is 1.

MTEE MAC Transmission Error interrupt Enable Bit

While this bit is 1 an interrupt will be generated when EIS.MTEF is 1.

MREE MAC Reception Error interrupt Enable Bit

While this bit is 1 an interrupt will be generated when EIS.MREF is 1.

24.3.30 Error Interrupt Status Register (EIS)

The EIS register indicates the states of AVB-DMAC-related error interrupts.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $35C_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	QFS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TFFF	CULF1	CULF0	CLLF1	CLLF0	SEF	QEF	MTEF	MREF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.39 EIS register contents

Bit Position	Bit Name	Function
b31 to b17	—	Reserved These bits are read as 0. The write value should be 0.
b16	QFS	Queue Full Error Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b15 to b9	—	Reserved These bits are read as 0. The write value should be 0.
b8	TFFF	Time Stamp FIFO Full Error Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b7	CULF1	CBS Upper Limit Error Interrupt Status (Class A) 0: The interrupt is not pending. 1: The interrupt is pending.
b6	CULF0	CBS Upper Limit Error Interrupt Status (Class B) 0: The interrupt is not pending. 1: The interrupt is pending.
b5	CLLF1	CBS Lower Limit Error Interrupt Status (Class A) 0: The interrupt is not pending. 1: The interrupt is pending.
b4	CLLF0	CBS Lower Limit Error Interrupt Status (Class B) 0: The interrupt is not pending. 1: The interrupt is pending.
b3	SEF	Separation Error Flag 0: The interrupt is not pending. 1: The interrupt is pending.
b2	QEF	Queue Error Flag 0: The interrupt is not pending. 1: The interrupt is pending.
b1	MTEF	MAC Transmission Error Flag 0: The interrupt is not pending. 1: The interrupt is pending.
b0	MREF	MAC Reception Error Flag 0: The interrupt is not pending. 1: The interrupt is pending.

QFS Queue Full Error Status Bit

With the interrupts enabled, this bit indicates that a queue is full (the receive queue r full interrupt status bit (QFFr) or the receive FIFO full interrupt status bit (RFFF) in receive interrupt status register 2 (RIS2) = 1).

[Conditions for Changing]

If the receive queue r full interrupt status bit (RIS2.QFFr) and the receive queue r full interrupt enable bit in the receive interrupt control register 2 (RIC2.QFEr) are updated, this bit is also updated.

If the receive FIFO full interrupt status bit (RIS2.RFFF) and the receive FIFO full interrupt enable bit (RIC2.RFFE) are updated, this bit is also updated.

TFFF Time Stamp FIFO Full-Error Interrupt Status Bit

This bit indicates that a new transmission time stamp has been discarded due to the time-stamp FIFO being full (i.e. has reached the overflow state).

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.
The bit is set to 1 when a frame with DESCR.TSR set is transmitted while the time stamp FIFO enable bit in the transmit configuration control register (TCCR.TFEN) is set to 1 and the time stamp FIFO count bit in the transmit status register (TSR.TFFL[2:0]) is set to 2.

CULF1 CBS Upper Limit Error Interrupt Status Bit (Class A)

This bit indicates that CBS counter 1 has exceeded the set upper limit (CUL1.ULV[31:0] in the CBS upper limit register c (CULc)).

Only 0 can be written to the bit.

[Conditions for Changing]

This bit is set to 0 when the operating mode is not operation mode.

This bit is set to 1 when the value of the CBS counter status 1 (Class A) bits in the transmit status register (TSR.CCS1[1:0]) change from B’00 (indicating a value within the range between the limits) to B’10 (indicating a value over the upper limit).

CULF0 CBS Upper Limit Error Interrupt Status Bit (Class B)

This bit indicates that CBS counter 0 has exceeded the set upper limit (CUL0.ULV[31:0] in the CBS upper limit register c (CULc)).

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when the value of the CBS counter status 0 (Class B) bit in the transmit status register changes from B’00 (indicating a value within the range between the limits) to B’10 (indicating a value over the upper limit).

CLLF1 CBS Lower Limit Error Interrupt Status Bit (Class A)

This bit indicates that CBS counter 1 has fallen below the set lower limit (CLL1.LLV[31:0] in CBS lower limit register c (CLLc)).

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when the value of the CBS counter status 1 (Class A) bit in the transmit status register (TSR.CCS1[1:0]) changes from B'00 (indicating a value within the range between the limits) to B'01 (indicating a value less than the lower limit).

CLLF0 CBS Lower Limit Error Interrupt Status Bit (Class B)

This bit indicates that CBS counter 0 has fallen below the set lower limit (CLL0.LLV[31:0] in the CBS lower limit register c (CLLc)).

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when the value of the CBS counter status 0 (Class B) bit in the transmit status register (TSR.CCS0[1:0]) changes from B'00 (indicating a value within the range between the limits) to B'01 (indicating a value less than the lower limit).

SEF Separation Error Flag

This bit indicates that a received frame was discarded because it has not matched any configured separation filter for AVB stream data frames.

The CPU can only write 0 to this bit.

[Changing condition]

This bit is set to 0b when leaving OPERATION mode.

This bit is set to 1 when a valid AVB stream data frame was received by MAC but discarded because RCR.ESF[1:0] is B'10 and no separation filter has matched.

QEF Queue Error Flag

This bit indicates that an error has been detected while processing reception or transmit queue.

Detail about the detected error is indicated by ESR.

The CPU can only write 0 to this bit.

[Changing condition]

This bit is set to 0 when leaving OPERATION mode.

This bit is set to 1 when an error condition is detected.

MTEF MAC Transmission Error Flag

This bit indicates that the MAC has detected a fault during transmission.

For detail the MAC registers have to be checked.

The CPU can only write 0 to this bit.

[Changing condition]

This bit is set to 0 when leaving OPERATION mode.

This bit is set to 1 when MAC detects an error during frame transmission.

MREF MAC Reception Error Flag

This bit indicates that the MAC has detected a fault during reception.

For detail the MAC registers have to be checked.

NOTE

When the storage of faulty received frames (RCR.EFFS) is enabled, the MAC error code (DESCR.MSC) is stored in the descriptor. By evaluating this information CPU can identify corrupted frames in URAM.

The CPU can only write 0 to this bit.

[Changing condition]

This bit is set to 0 when leaving OPERATION mode.

This bit is set to 1 when MAC detects an error during frame reception.

24.3.31 Receive Interrupt Control Register 0 (RIC0)

The RIC0 register controls the AVB-DMAC receive interrupts.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 360_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRE17	FRE16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FRE15	FRE14	FRE13	FRE12	FRE11	FRE10	FRE9	FRE8	FRE7	FRE6	FRE5	FRE4	FRE3	FRE2	FRE1	FRE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.40 RIC0 register contents (1/2)

Bit Position	Bit Name	Function
b31 to b18	—	Reserved These bits are read as 0. The write value should be 0.
b17	FRE17	Receive Frame Enable 17 (Stream) 0: Disabled 1: Enabled
b16	FRE16	Receive Frame Enable 16 (Stream) 0: Disabled 1: Enabled
b15	FRE15	Receive Frame Enable 15 (Stream) 0: Disabled 1: Enabled
b14	FRE14	Receive Frame Enable 14 (Stream) 0: Disabled 1: Enabled
b13	FRE13	Receive Frame Enable 13 (Stream) 0: Disabled 1: Enabled
b12	FRE12	Receive Frame Enable 12 (Stream) 0: Disabled 1: Enabled
b11	FRE11	Receive Frame Enable 11 (Stream) 0: Disabled 1: Enabled
b10	FRE10	Receive Frame Enable 10 (Stream) 0: Disabled 1: Enabled
b9	FRE9	Receive Frame Enable 9 (Stream) 0: Disabled 1: Enabled
b8	FRE8	Receive Frame Enable 8 (Stream) 0: Disabled 1: Enabled

Table 24.40 RIC0 register contents (2/2)

Bit Position	Bit Name	Function
b7	FRE7	Receive Frame Enable 7 (Stream) 0: Disabled 1: Enabled
b6	FRE6	Receive Frame Enable 6 (Stream) 0: Disabled 1: Enabled
b5	FRE5	Receive Frame Enable 5 (Stream) 0: Disabled 1: Enabled
b4	FRE4	Receive Frame Enable 4 (Stream) 0: Disabled 1: Enabled
b3	FRE3	Receive Frame Enable 3 (Stream) 0: Disabled 1: Enabled
b2	FRE2	Receive Frame Enable 2 (Stream) 0: Disabled 1: Enabled
b1	FRE1	Receive Frame Enable 1 (Network Control) 0: Disabled 1: Enabled
b0	FRE0	Receive Frame Enable 0 (Best Effort) 0: Disabled 1: Enabled

FRE0 to FRE17 Receive Frame Interrupt Enable Bits 0 to 17

When an interrupt source flag is set (a bit from among the receive interrupt status bits in the receive interrupt status register (RIS0.FRF0 to 17) = 1) while the interrupt is enabled, the interrupt is issued.

24.3.32 Receive Interrupt Status Register 0 (RIS0)

The RIS0 register indicates the states of the AVB-DMAC receive interrupts.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 364_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRF17	FRF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FRF15	FRF14	FRF13	FRF12	FRF11	FRF10	FRF9	FRF8	FRF7	FRF6	FRF5	FRF4	FRF3	FRF2	FRF1	FRF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.41 RIS0 register contents (1/2)

Bit Position	Bit Name	Function
b31 to b18	—	Reserved These bits are read as 0. The write value should be 0.
b17	FRF17	Receive Frame Interrupt Status 17 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b16	FRF16	Receive Frame Interrupt Status 16 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b15	FRF15	Receive Frame Interrupt Status 15 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b14	FRF14	Receive Frame Interrupt Status 14 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b13	FRF13	Receive Frame Interrupt Status 13 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b12	FRF12	Receive Frame Interrupt Status 12 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b11	FRF11	Receive Frame Interrupt Status 11 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b10	FRF10	Receive Frame Interrupt Status 11 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b9	FRF9	Receive Frame Interrupt Status 9 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b8	FRF8	Receive Frame Interrupt Status 8 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.

Table 24.41 RIS0 register contents (2/2)

Bit Position	Bit Name	Function
b7	FRF7	Receive Frame Interrupt Status 7 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b6	FRF6	Receive Frame Interrupt Status 6 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b5	FRF5	Receive Frame Interrupt Status 5 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b4	FRF4	Receive Frame Interrupt Status 4 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b3	FRF3	Receive Frame Interrupt Status 3 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b2	FRF2	Receive Frame Interrupt Status 2 (Stream) 0: The interrupt is not pending. 1: The interrupt is pending.
b1	FRF1	Receive Frame Interrupt Status 1 (Network Control) 0: The interrupt is not pending. 1: The interrupt is pending.
b0	FRF0	Receive Frame Interrupt Status 0 (Best Effort) 0: The interrupt is not pending. 1: The interrupt is pending.

FRF0 to FRF17 Receive Frame Interrupt Status Bits 0 to 17

Each bit indicates that a corresponding frame has been stored normally in reception queues 0 to 17 and that data are ready for CPU processing.

Only 0 can be written to the bit.

[Conditions for Changing]

A bit is set to 0 when the operating mode is not operation mode.

A bit is set to 0 when a value is written to the unread frame counter decrement register i (UFCDi) (i = 0 to 4), and this decrements the value of unread frame counter register i (UFCVi) (i = 0 to 4) to 0.

When a frame is stored normally in a reception queue, the corresponding bit is set to 1.

24.3.33 Receive Interrupt Control Register 1 (RIC1)

The RIC1 register controls the AVB-DMAC receive interrupts.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 368_H

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFWE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.42 RIC1 register contents

Bit Position	Bit Name	Function
b31	RFWE	Receive FIFO Warning Interrupt Enable 0: Disabled 1: Enabled
b30 to b0	—	Reserved These bits are read as 0. The write value should be 0.

RFWE Receive FIFO Warning Interrupt Enable Bit

If the reception FIFO reaches the caution level (the value set in the receive FIFO caution level bits in the receive configuration register (RCR.RFCL[12:0]) with the corresponding interrupt enabled, the interrupt is issued.

24.3.34 Receive Interrupt Status Register 1 (RIS1)

The RIS1 register indicates the states of the AVB-DMAC receive interrupts.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $36C_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFWF	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.43 RIS1 register contents

Bit Position	Bit Name	Function
b31	RFWF	Receive FIFO Warning Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b30 to b0	—	Reserved These bits are read as 0. The write value should be 0.

RFWF Receive FIFO Warning Interrupt Status Bit

This bit indicates that the reception FIFO has reached the set caution level (the value set in the receive FIFO caution level bits in the receive configuration register (RCR.RFCL[12:0])).

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when the reception FIFO reaches the set caution level (the value set in the receive FIFO caution level bits in the receive configuration register (RCR.RFCL[12:0])).

24.3.35 Receive Interrupt Control Register 2 (RIC2)

The RIC2 register controls the AVB-DMAC receive interrupts.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 370_H

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFFE	—	—	—	—	—	—	—	—	—	—	—	—	—	QFE17	QFE16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	QFE15	QFE14	QFE13	QFE12	QFE11	QFE10	QFE9	QFE8	QFE7	QFE6	QFE5	QFE4	QFE3	QFE2	QFE1	QFE0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.44 RIC2 register contents (1/2)

Bit Position	Bit Name	Function
b31	RFFE	Receive FIFO Full Interrupt Enable 0: Disabled 1: Enabled
b30 to b18	—	Reserved These bits are read as 0. The write value should be 0.
b17	QFE17	Receive Queue 17 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b16	QFE16	Receive Queue 16 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b15	QFE15	Receive Queue 15 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b14	QFE14	Receive Queue 14 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b13	QFE13	Receive Queue 13 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b12	QFE12	Receive Queue 12 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b11	QFE11	Receive Queue 11 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b10	QFE10	Receive Queue 10 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b9	QFE9	Receive Queue 9 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled

Table 24.44 RIC2 register contents (2/2)

Bit Position	Bit Name	Function
b8	QFE8	Receive Queue 8 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b7	QFE7	Receive Queue 7 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b6	QFE6	Receive Queue 6 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b5	QFE5	Receive Queue 5 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b4	QFE4	Receive Queue 4 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b3	QFE3	Receive Queue 3 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b2	QFE2	Receive Queue 2 (Stream) Full Interrupt Enable 0: Disabled 1: Enabled
b1	QFE1	Receive Queue 1 (Network Control) Full Interrupt Enable 0: Disabled 1: Enabled
b0	QFE0	Receive Queue 0 (Best Effort) Full Interrupt Enable 0: Disabled 1: Enabled

RFFE Receive FIFO Full Interrupt Enable Bit

When the reception FIFO is full and the interrupt is enabled, the interrupt is issued.

QFE0 to 17 Receive Queue 0 to 17 Full Interrupt Enable Bits

When a reception queue is full and the interrupt is enabled, the interrupt is issued.

24.3.36 Receive Interrupt Status Register 2 (RIS2)

The RIS2 register indicates the states of the AVB-DMAC receive interrupts.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 374_H

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	RFFF	—	—	—	—	—	—	—	—	—	—	—	—	—	QFF17	QFF16
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	QFF15	QFF14	QFF13	QFF12	QFF11	QFF10	QFF9	QFF8	QFF7	QFF6	QFF5	QFF4	QFF3	QFF2	QFF1	QFF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.45 RIS2 register contents (1/2)

Bit Position	Bit Name	Function
b31	RFFF	Receive FIFO Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending
b30 to b18	—	Reserved These bits are read as 0. The write value should be 0.
b17	QFF17	Receive Queue 17 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b16	QFF16	Receive Queue 16 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b15	QFF15	Receive Queue 15 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b14	QFF14	Receive Queue 14 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b13	QFF13	Receive Queue 13 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b12	QFF12	Receive Queue 12 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b11	QFF11	Receive Queue 11 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b10	QFF10	Receive Queue 10 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b9	QFF9	Receive Queue 9 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.

Table 24.45 RIS2 register contents (2/2)

Bit Position	Bit Name	Function
b8	QFF8	Receive Queue 8 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b7	QFF7	Receive Queue 7 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b6	QFF6	Receive Queue 6 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b5	QFF5	Receive Queue 5 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b4	QFF4	Receive Queue 4 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b3	QFF3	Receive Queue 3 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b2	QFF2	Receive Queue 2 (Stream) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b1	QFF1	Receive Queue 7 (Network Control) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.
b0	QFF0	Receive Queue 0 (Best Effort) Full Interrupt Status 0: The interrupt is not pending. 1: The interrupt is pending.

RFFF Receive FIFO Full Interrupt Status Bit

This bit indicates that a frame was received but storing it was not possible due to the reception FIFO being full.

When receiving a frame is not possible, the frame will be discarded.

Other information regarding discarded frames is not retained. Even if the frame is not discarded, this bit may also be set to 1 if the E-MAC determines that the frame is an error frame

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when the reception FIFO cannot hold received data.

QFF0 to 17 Receive 0 to 17 Full Interrupt Status Bits

When a received frame is dropped due to non-availability of empty descriptor, RIS2.QFFr is correctly set to 1.

Additionally, when UFCVi.CVr reached the configured stop level (UFCS.SLj), the queue full flag (RIS2.QFFr) is set to 1 even before any further received frame is dropped.

CAUTION

If no FEMPTY descriptors or no empty space for descriptors remains in the queue during storing of a divided frame (see Section 24.4.4.3 (b) Storing Frame Data as

Divided Frames, Storing a frame as a divided frame), an error frame is stored in the queue. Such error frames are treated as descriptor sequence errors.

[Conditions for Changing]

A bit is set to 0 when the operating mode is not operation mode.

A bit is set to 1 when reception queue r has no space available for storage.

A bit is set to 1 when the unread frame counter (unread frame counter register i (UFCVi) (i = 0 to 4)) reaches the set level for stopping.

24.3.37 Transmit Interrupt Control Register (TIC)

The TIC register controls the AVB-DMAC transmit interrupts.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 378_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TFWE	TFUE	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 24.46 TIC register contents

Bit Position	Bit Name	Function
b31 to b10	—	Reserved These bits are read as 0. The write value should be 0.
b9	TFWE	Time Stamp FIFO Warning Interrupt Enable 0: Disabled 1: Enabled
b8	TFUE	Time Stamp FIFO Update Interrupt Enable 0: Disabled 1: Enabled
b7 to b0	—	Reserved These bits are read as 0. The write value should be 0.

TFWE Time Stamp FIFO Warning Interrupt Enable Bit

When the time-stamp FIFO reaches the warning level while the interrupt is enabled, the interrupt is issued.

TFUE Time Stamp FIFO Update Interrupt Enable Bit

When the time-stamp FIFO is updated while the interrupt is enabled, the interrupt is issued.

24.3.38 Transmit Interrupt Status Register (TIS)

The TIS register indicates the states of the AVB-DMAC transmit interrupts.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $37C_H$

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TFWF	TFUF	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R

Table 24.47 TIS register contents

Bit Position	Bit Name	Function
b31 to b10	—	Reserved These bits are read as 0. The write value should be 0.
b9	TFWF	Time Stamp FIFO Warning Interrupt Status 0: The interrupt is not pending. 1: The time-stamp FIFO has reached the warning level.
b8	TFUF	Time Stamp FIFO Update Interrupt Status 0: The interrupt is not pending. 1: The time-stamp FIFO has been updated.
b7 to b0	—	Reserved These bits are read as 0. The write value should be 0.

TFWF Time Stamp FIFO Warning Interrupt Status Bit

This bit indicates that the transmission time-stamp FIFO has reached the warning level.

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode and when the time stamp FIFO enable bit in the transmit configuration control register (TCCR.TFEN) is 0.

The bit is set to 1 after a frame including DESCR.TSR set has been transmitted and one entry has already been stored in the time-stamp FIFO.

TFUF Time Stamp FIFO Update Interrupt Status Bit

This bit indicates that the transmission time-stamp FIFO has been updated.

Only 0 can be written to the bit.

[Conditions for Changing]

This bit is set to 0 when the operating mode is not operation mode.

This bit is set to 0 when the time stamp FIFO enable bit in the transmit configuration control register (TCCR.TFEN) is 0.

This bit is set to 0 when 1 is written to the time stamp FIFO release bit in the transmit configuration control register (TCCR.TFR).

This bit is set to 1 when the time stamp FIFO enable bit (TCCR.TFEN) is 1 after a frame including DESCR.TSR set has been transmitted.

24.3.39 Interrupt Summary Status Register (ISS)

The ISS register gives a summary of the states of AVB-DMAC-related interrupts.

Access: This register can be read in 32-bit units.

Address: <ETNBn_base> + 380_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DPS15	DPS14	DPS13	DPS12	DPS11	DPS10	DPS9	DPS8	DPS7	DPS6	DPS5	DPS4	DPS3	DPS2	DPS1	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	CGIS	RFWS	—	—	TFWS	TFUS	MS	ES	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.48 ISS register contents (1/2)

Bit Position	Bit Name	Function
b31	DPS15	Descriptor Interrupt 15 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b30	DPS14	Descriptor Interrupt 14 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b29	DPS13	Descriptor Interrupt 13 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b28	DPS12	Descriptor Interrupt 12 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b27	DPS11	Descriptor Interrupt 11 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b26	DPS10	Descriptor Interrupt 10 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b25	DPS9	Descriptor Interrupt 9 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b24	DPS8	Descriptor Interrupt 8 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b23	DPS7	Descriptor Interrupt 7 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b22	DPS6	Descriptor Interrupt 6 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b21	DPS5	Descriptor Interrupt 5 Summary 0: The interrupt is not pending. 1: The interrupt is pending.

Table 24.48 ISS register contents (2/2)

Bit Position	Bit Name	Function
b20	DPS4	Descriptor Interrupt 4 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b19	DPS3	Descriptor Interrupt 3 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b18	DPS2	Descriptor Interrupt 2 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b17	DPS1	Descriptor Interrupt 1 Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b16 to b14	—	Reserved These bits are read as 0. The write value should be 0.
b13	CGIS	gPTP Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b12	RFWS	Receive FIFO Warning Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b11 to b10	—	Reserved These bits are read as 0. The write value should be 0.
b9	TFWS	Time Stamp FIFO Warning Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b8	TFUS	Time Stamp FIFO Update Interrupt 0: The interrupt is not pending. 1: The interrupt is pending.
b7	MS	E-MAC Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b6	ES	Error Interrupt Summary 0: The interrupt is not pending. 1: The interrupt is pending.
b5 to b0	—	Reserved These bits are read as 0. The write value should be 0.

DPS1 to DPS15 Descriptor Interrupt 1 to 15 Summary Bits

These bits are set to 1 when the given descriptor interrupt enable bit (DIC.DPE1 to DPE15) and descriptor interrupt status flag (DIS.DPF1 to DPF15) are both 1.

CGIS gPTP Interrupt Summary Bit

This bit is set to 1 when either interrupt-related bit in the two gPTP-related interrupt registers (GIC and GIS) is 1.

RFWS Receive FIFO Warning Interrupt Summary Bit

This bit is set to 1 when the receive FIFO warning interrupt enable bit (RIC1.RFWE) and receive FIFO warning interrupt status flag (RIS1.RFWF) are both 1.

TFWS Time Stamp FIFO Warning Interrupt Summary Bit

This bit is set to 1 when the time stamp FIFO warning interrupt enable bit (TIC.TFWE) and time stamp FIFO warning interrupt status flag (TIS.TFWF) are both 1.

TFUS Time Stamp FIFO Update Interrupt Summary Bit

This bit is set to 1 when the time stamp FIFO update interrupt enable bit (TIC.TFUE) and time stamp FIFO update interrupt status flag (TIS.TFUF) are both 1.

MS E-MAC Interrupt Summary Bit

This bit is set to 1 when an E-MAC interrupt is issued.

ES Error Interrupt Summary Bit

This bit is set to 1 when any of the valid flags in the error interrupt status register (EIS) is 1 or the queue full error interrupt status bit (EIS.QFS) in the error interrupt status register (EIS) is 1.

24.3.40 gPTP Configuration Control Register (GCCR)

The GCCR register is used to set and control the gPTP (generalized precision time protocol).

Access: This register can be read/written in 8/16/32-bit units.

Address: <ETNBn_base> + 390_H

Initial value: 0000 $003C_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TCSS[1:0]		—	—	LMTT	LPTC	LTI	LTO	TCR[1:0]	
Value after reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.49 GCCR register contents

Bit Position	Bit Name	Function
b31 to b10	—	Reserved These bits are read as 0. The write value should be 0.
b9, b8	TCSS[1:0]	Timer Capture Source Select 00: gPTP timer value 01: Adjusted gPTP timer value 10: AVTP presentation time 11: Setting prohibited
b7, b6	—	Reserved These bits are read as 0. The write value should be 0.
b5	LMTT	Maximum Transit Time Configuration Request 0: Setting completed 1: When written: Issue a configuration request. When read: Completion of settings is pending.
b4	LPTC	Presentation Time Compare Value Configuration Request 0: Setting completed 1: When written: Issue a configuration request. When read: Completion of settings is pending.
b3	LTI	Timer Increment Value Configuration Request 0: Setting completed 1: When written: Issue a configuration request. When read: Completion of settings is pending.
b2	LTO	Timer Offset Value Configuration Request 0: Setting completed 1: When written: Issue a configuration request. When read: Completion of settings is pending.
b1, b0	TCR[1:0]	Timer Control Request 00: Timer control is not requested. 01: gPTP/AVPT presentation time reset 10: Setting prohibited 11: Captures the value set in the TCSS[1:0] bit.

TCSS[1:0] Timer Capture Source Select Bits

These bits select the source used for updating the captured timer register (gPTP timer capture register (GCTi.CTV[31:0])).

These bits should still be controlled when timer control is not being requested (GCCR.TCR[1:0] = 0).

LMTT Maximum Transit Time Configuration Request Bit

This bit issues requests for configuring the gPTP maximum transit time configuration register (GMTT).

Only 1 can be written to the bit.

[Conditions for Changing]

This bit is set to 1 when the operating mode is not operation mode.

This bit is set to 0 when the value of the gPTP maximum transit time configuration register (GMTT) is loaded.

LPTC Presentation Time Compare Value Configuration Request Bit

This bit issues requests for configuring the gPTP presentation time comparison register (GPTC).

Only 1 can be written to the bit.

[Conditions for Changing]

The bit is set to 1 when the operating mode is not operation mode.

The bit is set to 0 when the value of the gPTP presentation time comparison register (GPTC) is loaded.

LTI Timer Increment Value Configuration Request Bit

This bit issues requests for configuring the gPTP timer increment configuration register (GTI).

Only 1 can be written to the bit.

[Conditions for Changing]

The bit is set to 1 when the operating mode is not operation mode.

The bit is set to 0 when the value of the gPTP timer increment configuration register (GTI) is loaded.

LTO Timer Offset Value Configuration Request Bit

This bit issues requests for configuring gPTP timer offset configuration register i (GTOi).

Only 1 can be written to the bit.

[Conditions for Changing]

The bit is set to 1 when the operating mode is not operation mode.

The bit is set to 0 when the value of gPTP timer offset configuration register i (GTOi) is loaded.

TCR[1:0] Timer Control Request Bits

These bits issue requests for controlling the gPTP timer.

Writing to the bits is only possible when the current operating mode is operation mode.

Do not write to the bit when the gPTP timer clock select bit in the AVB-DMAC mode register is B’00.

[Conditions for Changing]

These bits are set to B’00 when the operating mode is not operation mode.

These bits are set to B'00 when the completion of the requested processing.

24.3.41 gPTP Maximum Transit Time Configuration Register (GMTT)

The GMTT register sets the maximum time for transitions of the gPTP timer.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 394_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MTTV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MTTV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.50 GMTT register contents

Bit Position	Bit Name	Function
b31 to b0	MTTV[31:0]	Maximum Transit Time The maximum transition time for addition to the presentation time

MTTV[31:0] Maximum Transit Time Bits

These bits set the maximum transition time for use in calculating AVTP presentation times.

Write the desired setting to the bits, then issue the configuration request by setting the maximum transit time configuration request bit in the gPTP configuration control register (GCCR.LMTT) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the maximum transit time configuration request bit (GCCR.LMTT) is 1.

24.3.42 gPTP Presentation Time Comparison Register (GPTC)

The GPTC register sets a value for comparison with presentation times in the gPTP timer.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 398_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	PTCV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PTCV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.51 GPTC register contents

Bit Position	Bit Name	Function
b31 to b0	PTCV[31:0]	Presentation Time Comparison Value Value for comparison with the gPTP presentation times

PTCV[31:0] Presentation Time Comparison Value Bits

These bits set a value for comparison with AVTP timer values to which a maximum transit time is not appended.

Write the desired setting to the bits, then issue the configuration request by setting the presentation time comparison value configuration request bit in the gPTP configuration control register (GCCR.LPTC) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the presentation time comparison value configuration request bit (GCCR.LPTC) is 1.

24.3.43 gPTP Timer Increment Configuration Register (GTI)

The GTI register sets the increment for the gPTP timer.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $39C_H$

Initial value: 0000 0001_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	TIV[27:16]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TIV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.52 GTI register contents

Bit Position	Bit Name	Function
b31 to b28	—	Reserved These bits are read as 0. The write value should be 0.
b27 to b0	TIV[27:0]	gPTP Timer Increment Value Increment for the gPTP timer

TIV[27:0] Bits (gPTP Timer Increment Value)

When the gPTP clock select bits in the AVB-DMAC mode register (CCC.CSEL[1:0]) are selecting a clock signal, these bits set the value by which the timer is incremented each time a cycle of that clock signal elapses.

Write the desired setting to the bits, then issue the configuration request by setting the timer increment value configuration request bit in the gPTP configuration control register (GCCR.LTI) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the timer increment value configuration request bit (GCCR.LTI) is 1.

Do not write 0 to the bits.

24.3.44 gPTP Timer Offset Configuration Register i (GTOi) (i = 0 to 2)

The GTOi register sets an offset value for the gPTP timer.

The offset value is added to the combination of bits 0 to 31 in GTO0, 32 to 63 in GTO1, and 64 to 79 in GTO2, which together make up the gPTP timer.

Access: This register can be read/written in 32-bit units.

Address: GTO0: <ETNBn_base> + $3A0_H$ GTO1: <ETNBn_base> + $3A4_H$ GTO2: <ETNBn_base> + $3A8_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	TOV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TOV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.53 GTOi register contents

Bit Position	Bit Name	Function
b31 to b0	TOV[31:0]	Timer Offset Value Offset value for the gPTP timer

TOV[79:0] Timer Offset Value Bits

This is an 80-bit value consisting of the settings in GTO0.TOV[31:0], GTO1.TOV[63:32], and GTO2.TOV[79:64], and is used to set an offset for adding to the value of the gPTP timer.

Write the desired setting to the bits, then issue the configuration request by setting the timer offset value configuration request bit in the gPTP configuration control register (GCCR.LTO) to 1.

CAUTION

Do not write a value to these bits when the operating mode is operation mode and the timer offset value configuration request bit (GCCR.LTO) is 1.

Write H'0000 to GTO2.TOV[95:80].

Set a value in the range from 0 to 10^9-1 (H'0000 0000 to H'3B9A C9FF) in GTO0.TOV[31:0].

24.3.45 gPTP Interrupt Control Register (GIC)

The GCI register is used to control gPTP-related interrupts.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $3AC_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PTME	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Table 24.54 GIC register contents

Bit Position	Bit Name	Function
b31 to b3	—	Reserved These bits are read as 0. The write value should be 0.
b2	PTME	Presentation Time Match Interrupt Enable 0: Disabled 1: Enabled
b1, b0	—	Reserved These bits are read as 0. The write value should be 0.

PTME Presentation Time Match Interrupt Enable Bit

When this bit is 1, setting of the presentation time match interrupt flag in the gPTP interrupt status register (GIS.PTMF) to 1 leads to generation of that interrupt.

24.3.46 gPTP Interrupt Status Register (GIS)

The GIC register indicates the state of the gPTP-related interrupt.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $3B0_H$

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PTMF	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R

Table 24.55 GIS register contents

Bit Position	Bit Name	Function
b31 to b3	—	Reserved These bits are read as 0. The write value should be 0.
b2	PTMF	Presentation Time Match Interrupt Flag 0: The interrupt is not pending. 1: The interrupt is pending.
b1, b0	—	Reserved These bits are read as 0. The write value should be 0.

PTMF Presentation Time Match Interrupt Flag Bit

This bit indicates that the value of the AVTP timer exceeds the value of the gPTP presentation time comparison register (GPTC).

Only 0 can be written to the bit.

[Conditions for Changing]

The bit is set to 0 when the operating mode is not operation mode.

The bit is set to 1 when the AVTP timer value is greater than or equal to the value of the gPTP presentation time comparison register (GPTC).

24.3.47 gPTP Timer Capture Register i (GCTi) (i = 0 to 2)

The GCTi registers form an 80-bit register that captures the gPTP timer value.

Access: This register can be read in 32-bit units.

Address: GCT0: <ETNBn_base> + $3B8_H$ GCT1: <ETNBn_base> + $3BC_H$ GCT2: <ETNBn_base> + $3C0_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	CTV[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CTV[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.56 GCTi register contents

Bit Position	Bit Name	Function
b31 to b0	CTV[31:0]	gPTP Timer Capture Value Captured timer value

CTV[79:0] gPTP Timer Capture Value Bits

These 80 bits consist of GCT0.CTV[31:0], GCT1.CTV[63:32 and GCT2.CTV[79:64], which together indicate captured timer values.

When B’00 (value of the gPTP timer) or B'01 (adjusted gPTP timer value) is selected by the timer capture source select bits in the gPTP configuration control register, the corresponding 80-bit values are stored in these bits.

When B’10 (AVTP presentation time) is selected by the timer capture source select bit, the corresponding 32-bit values are stored in these bits.

Actual writing of the timer value specified by the timer capture source select bits (GCCR.TCSS[1:0]) proceeds when B’11 (timer capture request) is written to the timer control request bits in the gPTP configuration control register (GCCR.TCR[1:0]).

Do not read the value while the value of the timer control request bits (GCCR.TCR[1:0]) is B’11, because this indicates that storage is still in progress.

24.3.48 E-MAC Mode Register (ECMR)

ECMR is used to specify the operating mode of the E-MAC. The settings in this register are normally made in the initialization process following a reset.

The operating mode settings must not be changed while transmission or reception is enabled (i.e. while the RE or TE bit in this register is 1).

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 500_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	TRCCM	—	—	RCSC	—	DPAD	RZPF	ZPF	PFR	RXF	TXF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	MPDE	—	—	RE	TE	—	—	—	DM	PRM
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R	R	R	R/W	R/W

Table 24.57 ECMR register contents (1/2)

Bit Position	Bit Name	Function
b31 to b27	—	Reserved These bits are read as 0. The write value should be 0.
b26	TRCCM	Counter Clear Mode 0: Writing to a counter register leads to the register being cleared to 0. 1: Reading from a counter register leads to the register being cleared to 0.
b25, b24	—	Reserved These bits are read as 0. The write value should be 0.
b23	RCSC	Checksum Calculation 0: Checksums are not automatically calculated. 1: Checksums are automatically calculated.
b22	—	Reserved This bit is read as 0. The write value should be 0.
b21	DPAD	Data Padding 0: Padding to make up 60 bytes is inserted in data for transmission when fewer than 60 bytes are to be transmitted. 1: Padding is not inserted in data for transmission when fewer than 60 bytes are to be transmitted and the data are transmitted without being changed.
b20	RZPF	PAUSE Frame Reception with Time = 0 0: Reception of PAUSE frames with the TIME parameter value 0 is disabled. 1: Reception of PAUSE frames with the TIME parameter value 0 is enabled.
b19	ZPF	PAUSE Frame Usage with TIME = 0 Enable/Lost Carrier Error Detection Enable • PAUSE frame usage with TIME = 0 enable (in full-duplex mode) 0: Control in response to and for the sending of PAUSE frames with the TIME parameter value 0 is disabled. 1: Control in response to and for the sending of PAUSE frames with the TIME parameter value is 0 is enabled. • Lost carrier error detection enable (in half-duplex mode) 0: Checking for lost carrier errors proceeds during frame transmission. 1: Checking for lost carrier errors does not proceed during frame transmission

Table 24.57 ECMR register contents (2/2)

Bit Position	Bit Name	Function
b18	PFR	PAUSE Frame Receive Mode 0: PAUSE frames are not transferred to the AVB-DMAC. 1: PAUSE frames are transferred to the AVB-DMAC.
b17	RXF	Operating Mode for Flow Control in Reception 0: Detection of PAUSE frames is disabled. 1: Flow control for the receiving port is enabled.
b16	TXF	Operating Mode for Flow Control in Transmission 0: Flow control for the transmitting port is disabled (PAUSE frames are not automatically transmitted). 1: Flow control for the transmitting port is enabled (PAUSE frames are automatically transmitted as required).
b15 to b10	—	Reserved These bits are read as 0. The write value should be 0.
b9	MPDE	Magic Packet™ Detection Enable 0: Magic Packet™ detection is not enabled. 1: Magic Packet™ detection is enabled.
b8, b7	—	Reserved These bits are read as 0. The write value should be 0.
b6	RE	Reception Enable 0: Reception is disabled. 1: Reception is enabled.
b5	TE	Transmission Enable 0: Transmission is disabled. 1: Transmission is enabled.
b4 to b2	—	Reserved These bits are read as 0. The write value should be 0.
b1	DM	Mode selection 0: Half-duplex mode 1: Full-duplex mode
b0	PRM	Promiscuous Mode 0: Normal operation 1: Promiscuous mode operation

TRCCM Counter Clear Mode Bit

This bit sets the method for clearing the counter register. Refer to the descriptions of the counter registers.

RCSC Checksum Calculation Bit

Setting this bit to 1 enables automatic calculation of checksums for data in received frames.

Only the data field of an Ethernet frame is in the scope of checksum calculation. Specifically, the checksum is calculated from the data field, which follows the length/type field and is followed by the CRC field. Calculation only involves 16-bit addition; it does not involve bit inversion.

DPAD Data Padding Control Bit

This bit specifies padding or non-padding of data when less than 60 bytes are to be transmitted.

When this bit is set to 1, data are transmitted without padding; when it is set to 0, data are padded to make up 60-byte units for transmission.

RZPF (PAUSE Frame Reception with Time = 0) Bit

When the RZPF bit is set to 0, received PAUSE frames with the Timer value 0 are discarded.

When the RZPF bit is set to 1, release from the transmission wait state follows reception of a PAUSE frame with the Timer value 0.

ZPF (PAUSE Frame Usage with TIME = 0 Enable/Lost Carrier Error Detection Enable) Bit

PAUSE frame usage with TIME = 0 enable (In full-duplex mode)

When the ZPF bit is set to 0, the next frame to be transmitted is not transmitted until the time specified by the Timer value has elapsed.

When the ZPF bit is set to 1, if the amount of data in the reception FIFO becomes less than the setting of the receive FIFO critical level bits (RCR.RFCL[12:0]) before the time specified by the Timer value elapses, a PAUSE frame with a Timer value of 0 is automatically transmitted. If the interface is in the transmission wait state, it is released from that state on receiving a PAUSE frame with a Timer value of 0.

- Lost carrier error detection enable (in half-duplex mode)
  In half-duplex operation, this bit enables or disables lost carrier error detection.
- Lost carrier error detection enable (in half-duplex mode)
  signal (AVB_TX_EN) (active high) to detection of the carrier detection signal (AVB_CRS) being 1 is 63 BP* or less.
  If the time from activation of the transmit data enable signal (AVB_TX_EN) (active high) to detection of the carrier detection signal (AVB_CRS) being 1 is greater than 63 BP*, or if the timing of the carrier detection signal (AVB_CRS) is undefined, this bit should not be cleared to 0 (enabling error detection).

NOTE

BP: Bit period. 1 BP = 10 ns (100 Mbps), 1 BP = 100 ns (10 Mbps)

PFR PAUSE Frame Receive Mode Bit

This bit specifies whether PAUSE frames are transferred to the AVB-DMAC.

RXF (Operating Mode for Flow Control in Reception) Bit

When the RXF bit is set to 1 and a PAUSE frame is received, a next frame to be transmitted is not transmitted until the time indicated by the Timer value in the PAUSE frame has elapsed. However, the transmission of a current frame is continued. The number of received PAUSE frames is also counted. For details, see Section 24.3.56, PAUSE Frame Receive Counter (PFRCR).

Setting this bit to 0 disables PAUSE frame detection.

TXF (Operating Mode for Flow Control in Transmission) Bit

The TXF bit enables or disables flow control in transmission by the port.

Setting this bit to 0 disables PAUSE frame detection.

MPDE Magic Packet™ Detection Enable Bit

The MPDE bit enables or disables Magic Packet™ detection by hardware to allow activation via the Ethernet connection.

RE Reception Enable Bit

If this bit is switched from reception being enabled (RE = 1) to reception being disabled (RE = 0) while a frame is being received, reception will continue until reception of that frame is completed.

TE Transmission Enable Bit

If this bit is switched from transmission being enabled (TE = 1) to transmission being disabled (TE = 0) while a frame is being transmitted, transmission will continue until transmission of that frame is completed.

DM Duplex Mode Bit

This bit selects full- or half-duplex operation.

PRM Promiscuous Mode Bit

Setting the PRM bit enables all Ethernet frames to be received. All Ethernet frames means all receivable frames, irrespective of differences or enabled/disabled status (destination address, broadcast address, multicast bit, etc.).

24.3.49 Receive Frame Length Register (RFLR)

The RFLR register specifies the maximum length (in bytes) of frames that can be received by this LSI. Settings in this register must not be changed while reception is enabled (while the RE bit in the E-MAC mode register (EMCR) is 1).

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 508_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFL[17:16]	
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFL[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.58 RFLR register contents

Bit Position	Bit Name	Function
b31 to b18	—	Reserved These bits are read as 0. The write value should be 0.
b17 to b0	RFL[17:0]	Receive Frame Length H'00000 to H'005EE: 1,518 bytes H'005EF: 1,519 bytes H'005F0: 1,520 bytes : : H'007FF: 2,047 bytes H'00800: 2,048 bytes : : H'01000: 4,096 bytes : : H'10000: 65,535 bytes : : H'20000 to H'3FFFF: 131,072 bytes

RFL[17:0] Receive Frame Length Bits

Frame data described here refers to all fields from the destination address up to the CRC data. Frame contents from the destination address up to the data are actually transferred to memory. CRC data are not included in the transfer. When more data than the specified number of bytes are received, the portion of data that exceeds the specified value is discarded.

24.3.50 E-MAC Status Register (ECSR)

The ECSR register indicates the state of the E-MAC. The CPU can be notified of the state. For bits associated with interrupts, the interrupt can be enabled or disabled by the corresponding bit in the E-MAC Interrupt Permission Register (ECSIPR) described in **Section 24.3.51**.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 510_H

Initial value: B'0000 0000 0000 0000 0000 0000 0000 00xx

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MPD	ICD
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 24.59 ECSR register contents

Bit Position	Bit Name	Function
b31 to b2	—	Reserved These bits are read as 0. The write value should be 0.
b1	MPD	Magic Packet™ Detection 0: A Magic PacketTM has not been detected. 1: A Magic PacketTM has been detected.
b0	ICD	Illegal Carrier Detection 0: PHY-LSI has not detected an illegal carrier on the line. 1: PHY-LSI has detected an illegal carrier on the line.

MPD Magic Packet™ Detection Bit

This bit indicates that a Magic Packet™ has been detected on the line.

Writing 1 to this bit clears it to 0.

ICD Illegal Carrier Detection Bit

This bit indicates that the PHY-LSI has detected an illegal carrier on the line. If a change in the signal input from the PHY-LSI occurs in a period shorter than the software recognition period, the correct information may not be obtained. Refer to the timing specification for the PHY-LSI used.

Writing 1 to this bit clears it to 0.

24.3.51 E-MAC Interrupt Permission Register (ECSIPR)

The ECSIPR register enables or disables the states indicated by the ECSR register as interrupt sources. Each effective bit disables or enables interrupts corresponding to the bits in ECSR.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 518_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MPDIP	ICDIP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 24.60 ECSIPR register contents

Bit Position	Bit Name	Function
b31 to b2	—	Reserved These bits are read as 0. The write value should be 0.
b1	MPDIP	Magic Packet™ Detect Interrupt Enable 0: Interrupts on setting of the MPD bit is disabled. 1: Interrupts on setting of the MPD bit is enabled.
b0	ICDIP	False Carrier Detect Interrupt Enable 0: Interrupts on setting of the ICD bit is disabled. 1: Interrupt on setting of the ICD bit is enabled.

MPDIP Magic Packet™ Detect Interrupt Enable Bit

Setting this bit to 1 selects interrupt generation on setting of the Magic Packet™ detection bit (ECSR.MPD) in the E-MAC status register to 1.

ICDIP Illegal Carrier Detect Interrupt Enable Bit

Setting this bit to 1 selects interrupt generation on setting of the illegal carrier detection bit (ECSR.ICD) in the E-MAC status register to 1.

24.3.52 PHY Interface Register (PIR)

The PIR register provides a means of access to the PHY-LSI internal registers via the MII.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 520_H

Initial value: B'0000 0000 0000 0000 0000 0000 0000 x000

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	—	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 24.61 PIR register contents

Bit Position	Bit Name	Function
b31 to b4	—	Reserved These bits are read as 0. The write value should be 0.
b3	MDI	MII Management Data-In Indicates the level of the AVB_MDIO pin.
b2	MDO	MII Management Data-Out Holds data for output from the AVB_MDIO pin.
b1	MMD	MII Management Mode 0: Read direction is specified. 1: Write direction is specified.
b0	MDC	MII Management Data Clock The value set in this bit is output from the AVB_MDC pin, which supplies the management data clock for the MII.

MDI MII Management Data-In Bit

This bit indicates the level of the AVB_MDIO pin.

MDO MII Management Data-Out Bit

This bit holds data for output from the AVB_MDIO pin.

The AVB_MDIO pin outputs data when the MMD bit is set to 1 (to specify writing as the direction). Data are not output while the MMD bit is set to 0 (to specify reading as the direction).

MMD MII Management Mode Bit

This bit specifies the direction for data through MDIO (reading or writing).

MDC MII Management Data Clock Bit

Values set in this bit are output on the AVB_MDC pin to supply the MII with the management data clock. For the method of access to the MII registers, see Section 24.4.13, Connection to PHY-LSI.

24.3.53 Auto PAUSE Frame Time Parameter Register (APFTP)

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 554_H

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	APFTP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.62 APFTP register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to b0	APFTP[15:0]	Auto Pause Frame Time Parameter H'0000: — H'0001: 1 × 512 bit-period H'0002: 2 × 512 bit-period :: H'FFFF: 65535 × 512 bit-period **NOTE** A bit-period changes relative to the transfer speed as follows. • 100Mbps: 1 bit-period = 10 ns • 10Mbps: 1 bit-period = 100 ns **CAUTION** If ECMR.TXF set to 1, APFTP[15:0] should not be set to 0000_H.

APFTP[15:0] Auto Pause Frame Time Parameter

This bits configure the Timer value for the transmit of Auto Pause Frame.

The APFTP[15:0] register is used to set the value for the TIME parameter of automatically generated PAUSE

frames. When a PAUSE frame is automatically transmitted, the value set in this register is used as its TIME parameter.

Setting value is same as MP[15:0] bit of MPR register.

24.3.54 Manual PAUSE Frame Register (MPR)

The MPR register is used to set the value for the TIME parameter of manually generated PAUSE frames. When a PAUSE frame is manually transmitted, the value set in this register is used as its TIME parameter.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 558_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.63 MPR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to b0	MP[15:0]	Manual PAUSE These bits set the TIME parameter value of a manual PAUSE frame. Note H'0000: — H'0001: 1 × 512 bit-period H'0002: 2 × 512 bit-period :: H'FFFF: 65535 × 512 bit-period **NOTE** A bit-period changes relative to the transfer speed as follows. • 100Mbps: 1 bit-period = 10 ns • 10Mbps: 1 bit-period = 100 ns

MP[15:0] Manual PAUSE Bits

These bits set the value of the TIME parameter in manually generated PAUSE frames.

The unit for the setting is 512 bit periods.

24.3.55 PAUSE Frame Transmit Counter (PFTCR)

The PFTCR register is a counter that indicates the number of times PAUSE frames have been transmitted.

Access: This register is read-only in 32-bit units.

Address: <ETNBn_base> + $55C_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PFTXC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.64 PFTCR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0.
b15 to b0	PFTXC[15:0]	PAUSE Frame Transmit Counter Counter for counting the number of transmitted PAUSE frames

PFTXC[15:0] PAUSE Frame Transmit Counter Bits

These bits indicate the total number of PAUSE frames that have been transmitted (both manually and automatically).

The bits are cleared to 0 when they are read.

If counting up and clearing of the counter coincide, clearing the counter takes priority.

24.3.56 PAUSE Frame Receive Counter (PFRCR)

The RFRCR register is a counter that indicates the number of times PAUSE frames have been received.

Access: This register is read-only in 32-bit units.

Address: <ETNBn_base> + 560_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PFRXC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 24.65 RFRCR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0.
b15 to b0	PFRXC[15:0]	PAUSE Frame Receive Counter Counter for counting the number of received PAUSE frames

PFRXC[15:0] PAUSE Frame Receive Counter Bits

These bits indicate the number of PAUSE frames that have been received when flow control in reception is enabled (the RXF bit in ECMR = 1).

The bits are cleared to 0 when they are read.

If counting up and clearing the counter coincide, clearing the counter takes priority.

24.3.57 EthernetAVB Mode Register (GECMR)

The GECMR register specifies the operating mode for the EthernetAVB.

The setting in the ECMR1 register must not be changed while transmission or reception is enabled.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $5B0_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEED
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

CAUTION

The default value of the SPEED bit must not be changed to 1.

Table 24.66 ECMR1 register contents

Bit Position	Bit Name	Function
b31 to b1	—	Reserved These bits are read as 0. The write value should be 0.
b0	SPEED	Transfer Speed Setting 0: Transfer is at 100 Mbps. 1: Prohibited

SPEED Transfer Speed Setting Bit

This bit sets the transfer rate.

24.3.58 E-MAC Address High Register (MAHR)

The MAHR register specifies the 32 higher-order bits of the 48-bit E-MAC address. The settings in this register are normally made in the initialization process after a reset.

The settings in this register must not be changed while transmission or reception is enabled.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $5C0_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MA[47:32]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MA[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.67 MAHR register contents

Bit Position	Bit Name	Function
b31 to b0	MA[47:16]	E-MAC Address Bits 47 to 16 These bits are used to set the 32 higher-order bits of the E-MAC address.

MA[47:16] E-MAC Address Bits 47 to 16

These bits are used to set the 32 higher-order bits of the E-MAC address.

For example, if the E-MAC address is 01-23-45-67-89-AB (hexadecimal), set H'0123 4567 in the MAHR register.

24.3.59 E-MAC Address Low Register (MALR)

The MALR register specifies the 16 lower-order bits of the 48-bit E-MAC address. The settings in this register are normally made in the initialization process after a reset.

The settings in this register must not be changed while transmission or reception is enabled.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + $5C8_H$

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.68 MALR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to b0	MA[15:0]	E-MAC Address Bits 15 to 0 These bits are used to set the 16 lower-order bits of the E-MAC address.

MA[15:0] E-MAC Address Bits 15 to 0

These bits are used to set the 16 lower-order bits of the E-MAC address.

For example, if the E-MAC address is 01-23-45-67-89-AB (hexadecimal), set H'89AB in the MALR register.

24.3.60 Transmit Retry Over Counter Register (TROCR)

The TROCR register is a counter that indicates the number of frames the module was unable to transmit in 16 attempts at transmission including the first attempt and retries. When 16 attempts to transmit a frame fail, this register is incremented by 1. Counting up stops when the value in this register reaches H'0000 FFFF.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 700_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TROC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.69 TROCR register contents

Bit Position	Bit Name	Function
b31 to 16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to 0	TROC[15:0]	Transmit Retry Over Counter These bits indicate the number of frames the module was unable to transmit in 16 attempts at transmission, including the first attempt and retries.

TROC[15:0] Transmit Retry Over Counter Bits

These bits indicate the number of frames the module was unable to transmit in 16 attempts at transmission, including the first attempt and retries.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register (ECMR) is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

24.3.61 Lost Carrier Counter Register (LCCR)

The LCCR register is a counter that indicates the number of times the carrier was lost during data transmission. Counting up stops when the value in this register reaches H'0000 FFFF.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 710_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	LCC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.70 LCCR register contents

Bit Position	Bit Name	Function
b31 to 16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to 0	LCC[15:0]	Lost Carrier Counter These bits indicate the number of times the carrier was lost during data transmission.

LCC[15:0] Lost Carrier Counter Bits

These bits indicate the number of times the carrier was lost during data transmission.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register (ECMR) is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

24.3.62 CRC Error Frame Receive Counter Register (CEFCR)

The CEFCR register is a counter that indicates the number of times frames with CRC errors were received. Counting up stops when the value in this register reaches H'0000 FFFF.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 740_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	CEFC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.71 CEFCR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to b0	CEFC[15:0]	CRC Error Frame Counter These bits indicate the number of CRC error frames received.

CEFC[15:0] CRC Error Frame Counter Bits

These bits indicate the number of received frames having CRC errors.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

24.3.63 Frame Receive Error Counter Register (FRECR)

The FRECR register is a counter that indicates the number of frames for which receive errors were generated by input on the AVB_RX_ER pin from the PHY-LSI. Counting up stops when the value in this register reaches H'0000 FFFF.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 748_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FREC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.72 FRECR register contents

Bit Position	Bit Name	Function
b31 to 16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to 0	FREC[15:0]	Frame Receive Error Counter These bits indicate the number of errors during frame reception.

FREC[15:0] Frame Receive Error Counter Bits

These bits indicate the number of errors during frame reception.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

24.3.64 Too-Short Frame Receive Counter Register (TSFRCR)

The TSFRCR register is a counter that indicates the number of received frames that were fewer than 64 bytes in length. Counting stops when the value in this register reaches H'0000 FFFF.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 750_H

Initial value: $0000\ 0000_H$

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TSFRC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.73 TSFRCR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to b0	TSFRC[15:0]	Too-Short Frame Receive Counter These bits indicate the number of frames received with a length of less than 64 bytes.

TSFRCR[15:0] Too-Short Frame Receive Counter Bits

These bits indicate the number of received frames that were fewer than 64 bytes in length.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

24.3.65 Too-Long Frame Receive Counter Register (TLFRCR)

The TLFRCR register is a counter that indicates the number of received frames that were longer than the value specified in the receive frame length register (RFLR). Counting up stops when the value in the TLFRCR register reaches H'0000 FFFF.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 758_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	TLFC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.74 TLFRCR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to b0	TLFC[15:0]	Too-Long Frame Receive Counter These bits indicate the number of frames received with a length exceeding the value in RFLR.

TLFRCR[15:0] Too-Long Frame Receive Counter Bits

These bits indicate the number of received frames that were longer than the value in RFLR.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

24.3.66 Residual-Bit Frame Receive Counter Register (RFCR)

The RFCR register is a counter that indicates the number of received frames containing “residual bits” (trailing bits not making up an 8-bit unit). Counting up stops when the value in this register reaches H’0000 FFFF.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 760_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RFC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.75 RFCR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to b0	RFC[15:0]	Residual-Bit Frame Receive Counter These bits indicate the number of received frames containing residual bits.

RFC[15:0] Residual-Bit Frame Receive Counter Bits

These bits indicate the number of received frames containing residual bits.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

24.3.67 Multicast Address Frame Receive Counter Register (MAFCR)

The MAFCR register is a counter that indicates the number of received frames for which a multicast address was specified. Counting up stops when the value in this register reaches H'0000 FFFF.

Access: This register can be read/written in 32-bit units.

Address: <ETNBn_base> + 778_H

Initial value: 0000 0000_H

Bit	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MAFC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 24.76 MAFCR register contents

Bit Position	Bit Name	Function
b31 to b16	—	Reserved These bits are read as 0. The write value should be 0.
b15 to b0	MAFC[15:0]	Multicast Address Frame Counter These bits indicate the number of multicast frames that have been received.

MAFC[15:0] Multicast Address Frame Counter Bits

These bits indicate the number of multicast frames that have been received.

The bits are cleared to 0 when they are read while the counter clear mode bit (TRCCM) in the E-MAC mode register is set to 1.

When TRCCM = 0, they are cleared to 0 by the writing of any value to this register.

24.4 Operation

The EthernetAVB consists of the following functional units:

- DMA transfer controller (AVB-DMAC): Handles DMA transfer between the data storage areas for reception and transmission in the URAM and the reception and transmission FIFO buffers
- MAC controller (E-MAC): Handles transfer between the reception and transmission FIFO buffers and the MII

Using its direct memory access (DMA) function, the AVB-DMAC handles DMA transfer of frame data between the destinations for storing Ethernet frame data for transmission and reception in the URAM and the FIFO buffers for reception and transmission. Data cannot be directly read from or written to the FIFO buffers.

To handle DMA transfer, the AVB-DMAC requires information that includes the addresses for storage of data for transmission and received data. These data are referred to as descriptors. The AVB-DMAC reads data for transmission from the storage area for data to be transmitted according to the information in descriptors and writes received data to the storage area for received data accompanied by information in descriptors. The descriptors are placed in the URAM. Arranging multiple descriptors in descriptor lists allows the continuous reception or transmission of multiple Ethernet frames.

The E-MAC supports a MII, which provides an interface format for the externally connected PHY-LSI. The E-MAC constructs Ethernet frames from data written to the transmission FIFO and transmits these frames to the MII. It also performs CRC checking of Ethernet frames received from the MII and writes the frames to the reception FIFO.

24.4.1 AVB-DMAC Operating Modes

Figure 24.5 illustrates the operating modes of the AVB-DMAC.

Transitions of AVB-DMAC operating mode are under the control of the items listed below.

- CPU operating mode (hardware reset and power-down mode)
- Configuration of the operating mode configuration bits (CCC.OPC[1:0]) in the AVB-DMAC mode register

The current operating mode can be confirmed by reading the operating mode status bits in the AVB-DMAC status register (CSR.OPS[3:0]).

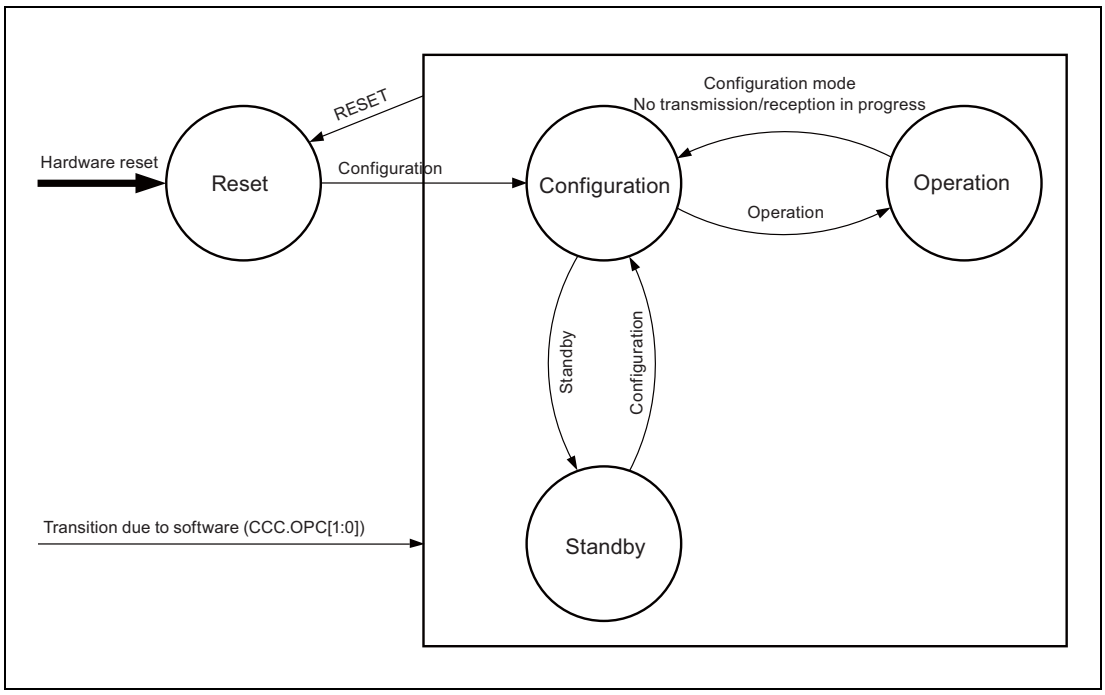


Figure 24.5 Operating Mode of AVB-DMAC

24.4.1.1 Operating Modes

(1) Reset mode

After a hardware reset, the AVB-DMAC enters reset mode.
In reset mode, only the AVB-DMAC operating mode control function is controllable and other functions are all stopped. This mode is designed for reduced power when the Ethernet function is not necessary.

(2) Configuration mode

In configuration mode, various settings for the AVB-DMAC can be made.
The operation of most functions is stopped and all status registers are initialized to their reset values.
The E-MAC functions in this mode.

(3) Operation mode

In operation mode, all functions of the AVB-DMAC can operate.
Ethernet communications can only proceed in this mode.

(4) Standby mode

In standby mode, the E-MAC can only be used to control the operating mode and to decode Magic Packets. Other functions cannot be used.

24.4.1.2 How to Set the Operating Mode

Set the operating mode configuration bits in the AVB-DMAC mode register (CCC.OPC[1:0]) to select the operating mode. Furthermore, the current operating mode can be checked by reading the operating mode status bits in the AVB-DMAC status register (CSR.OPS[3:0]).

Transitions other than from operation mode to configuration mode are made after the value is written to the operating mode configuration bits (CCC.OPC[1:0]) (**Figure 24.6**).

For transitions from operation mode to configuration mode, follow the procedure in **Figure 24.7** because any transmission and reception in progress will be executed before the transition to configuration mode.

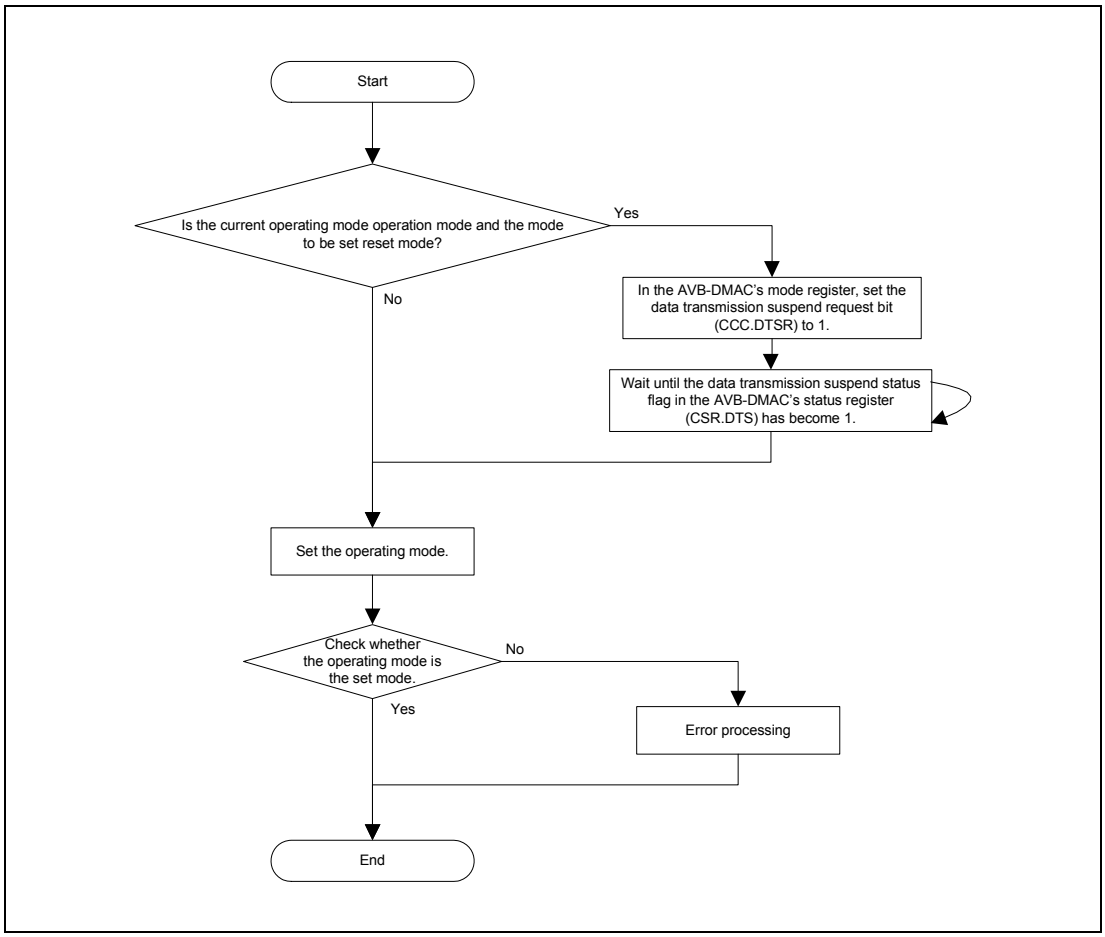


Figure 24.6 Flow for Transitions of Operating Mode (Other than from Operation Mode to Configuration Mode)

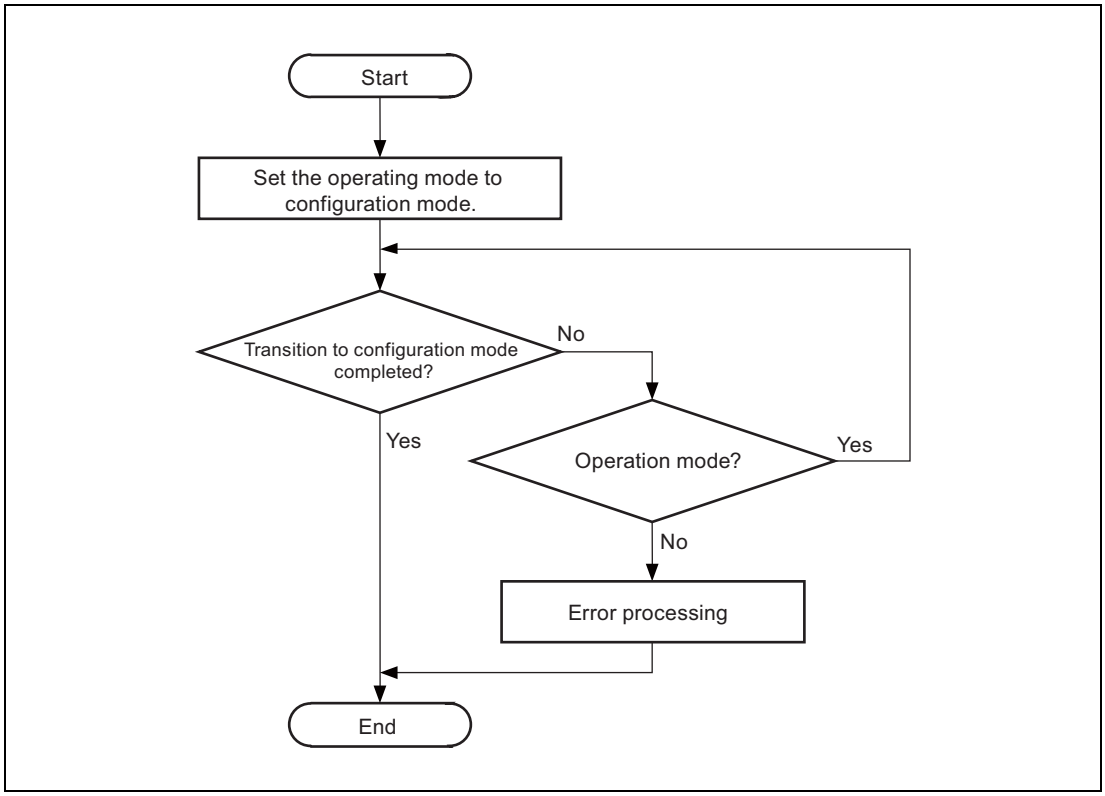


Figure 24.7 Flow for Transitions of Operating Mode (from Operation Mode to Configuration Mode)

In the transition from operation mode to configuration mode, the AVB-DMAC executes the following operations before the transition is completed. Read the operating mode status bits in the AVB-DMAC status register (CSR.OPS[3:0]) to check that the transition to configuration mode has been completed.

- If the transfer of a frame between the reception FIFO and URAM is in progress, this is completed (other received frames remaining in the FIFO and any frames that are subsequently received by the E-MAC are discarded).
- If the transfer of a frame is in progress between the transmission FIFO and URAM, this is completed
  (frames for transmission remaining in the URAM will not be transmitted).
- All frames for transmission in the transmission FIFO are transferred to the E-MAC.

Notes:

When the operating mode shifts to configuration mode, all status registers are cleared.

We recommend following the procedure below in the case of this transition.

1. Disable reception.
2. Since reception actually stopping after being disabled requires time, wait for an interval equivalent to that for reception of a maximum length packet.
3. Stop the software task that is generating data for transmission.
4. Wait until the receive process status bit (CSR.RPO) and the transmit process status bits (CSR.TPO0 to 3) in the AVB-DMAC status register are set to 0.
5. Capture all of the required status information.
6. Set the operating mode configuration bits in the AVB-DMAC mode register (CCC.OPC[1:0]) to initiate the transition to configuration mode.

24.4.1.3 Operating Mode Transitions Due to Hardware

The following hardware factors can also initiate transitions of the AVB-DMAC operating mode.

(1) Hardware reset

Resetting of the LSI chip leads to resetting of the entire EthernetAVB module. The operating mode shifts to reset mode.

(2) Transition during power-off

This transition is triggered by hardware during the power-off sequence under software control.

The AVB-DMAC completes the bus master access in progress (in the worst case, a 128-byte burst), and then shifts to reset mode. At this time, the operating mode configuration bits in the AVB-DMAC mode register (CCC.OPC[1:0]) are set to B’00.

24.4.2 Common Control for Transmission and Reception

24.4.2.1 Initialization Procedure

Figure 24.8 shows the overall initialization procedure in outline.

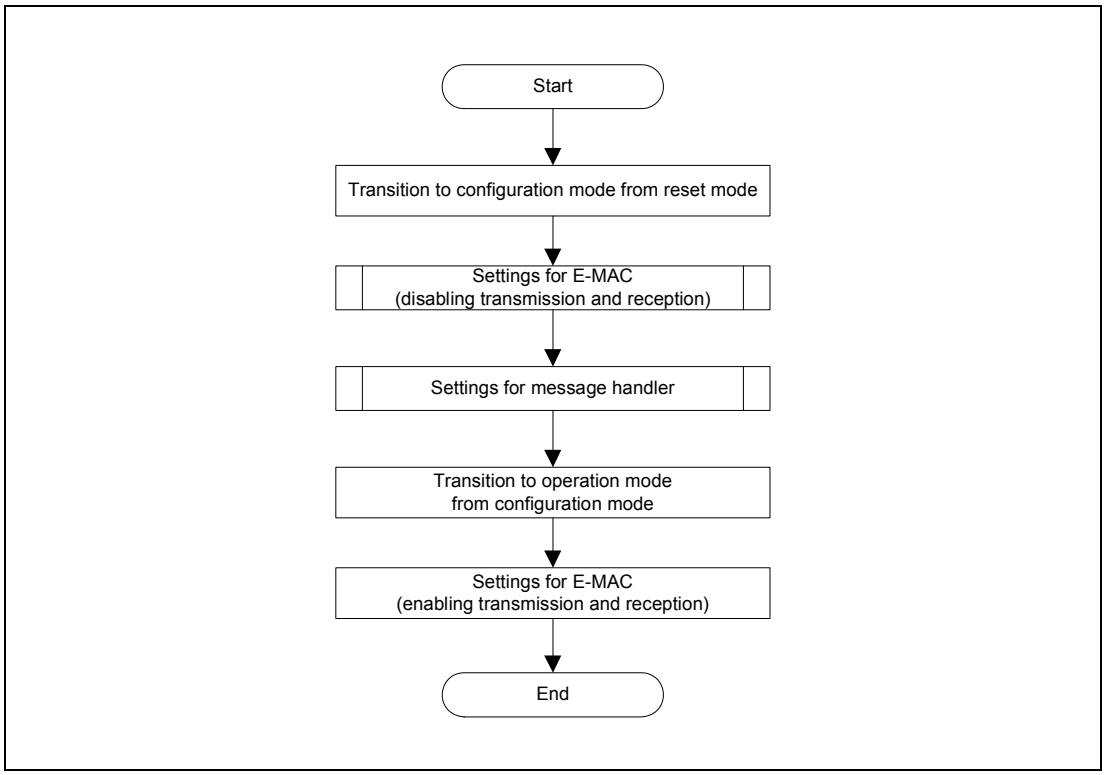


Figure 24.8 Outline of the Initialization Procedure

(1) Initializing the Receiver Section

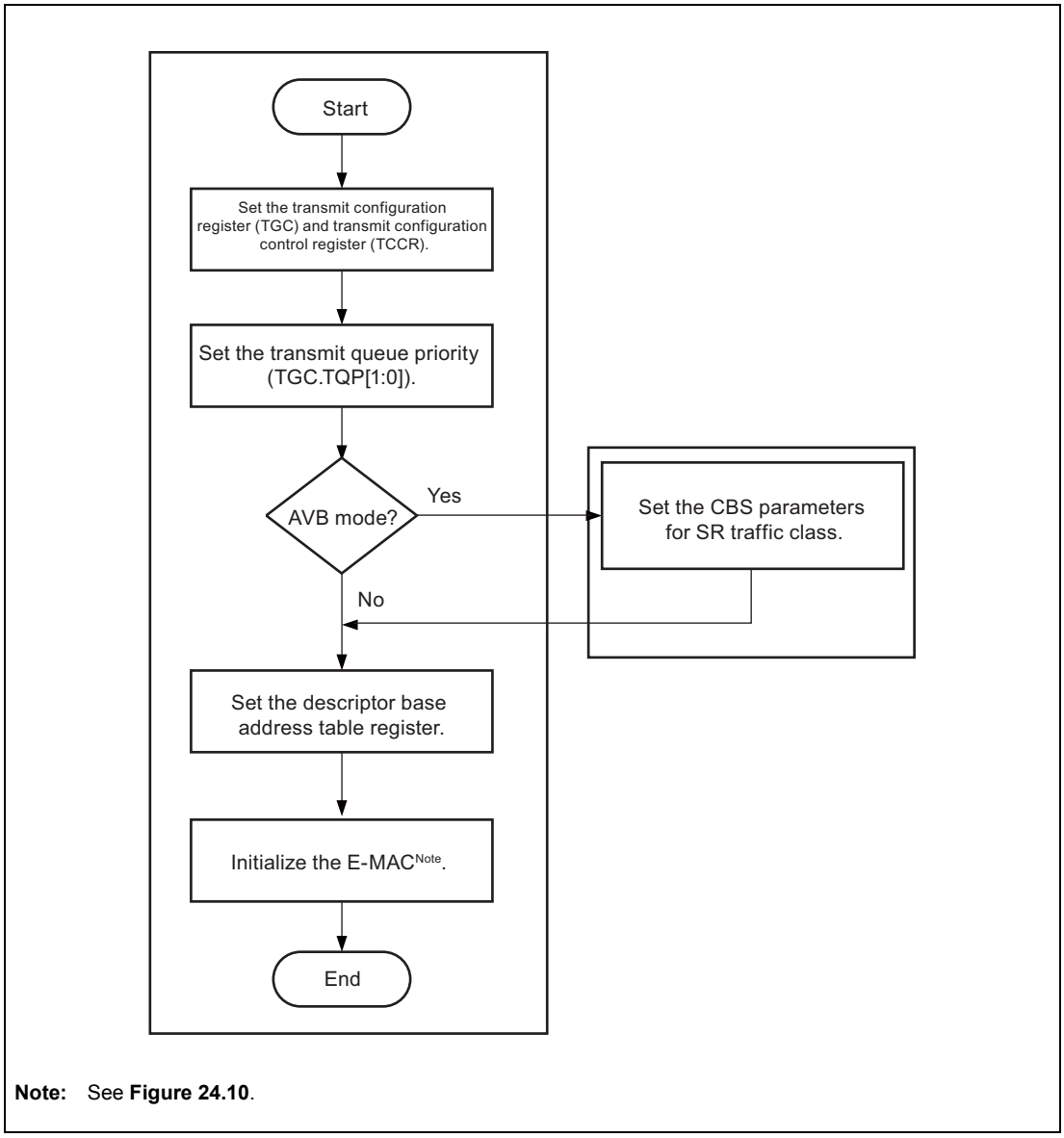
Before starting reception, follow the procedure below.

Keep the operating mode to configuration mode, and do not enable reception until the settings for the AVB-DMAC are completed.

- Set the operating mode to configuration mode.
- Set AVB filtering for network control frames and AVB stream frames to suit the specifications of the product the chip will be used in.
- Create a descriptor chain for each queue to be used.
- Set the base address for the descriptor table in the descriptor base address table register (DBAT).
- Specify the maximum frame length with the receive frame length upper limit register (RFLR).
- Specify whether padding is to be used with the receive padding configuration register (RPC).
- Set the unread frame counter for each queue with unread frame counter registers 0 to 4.

(2) Initializing the Transmitter Section

Figure 24.9 illustrates initialization of the transmitter section.



Note: See **Figure 24.10**.

Figure 24.9 Procedure for Initializing the Transmitter Section

(3) Initializing the E-MAC Section

Figure 24.10 illustrates initialization of the E-MAC section.

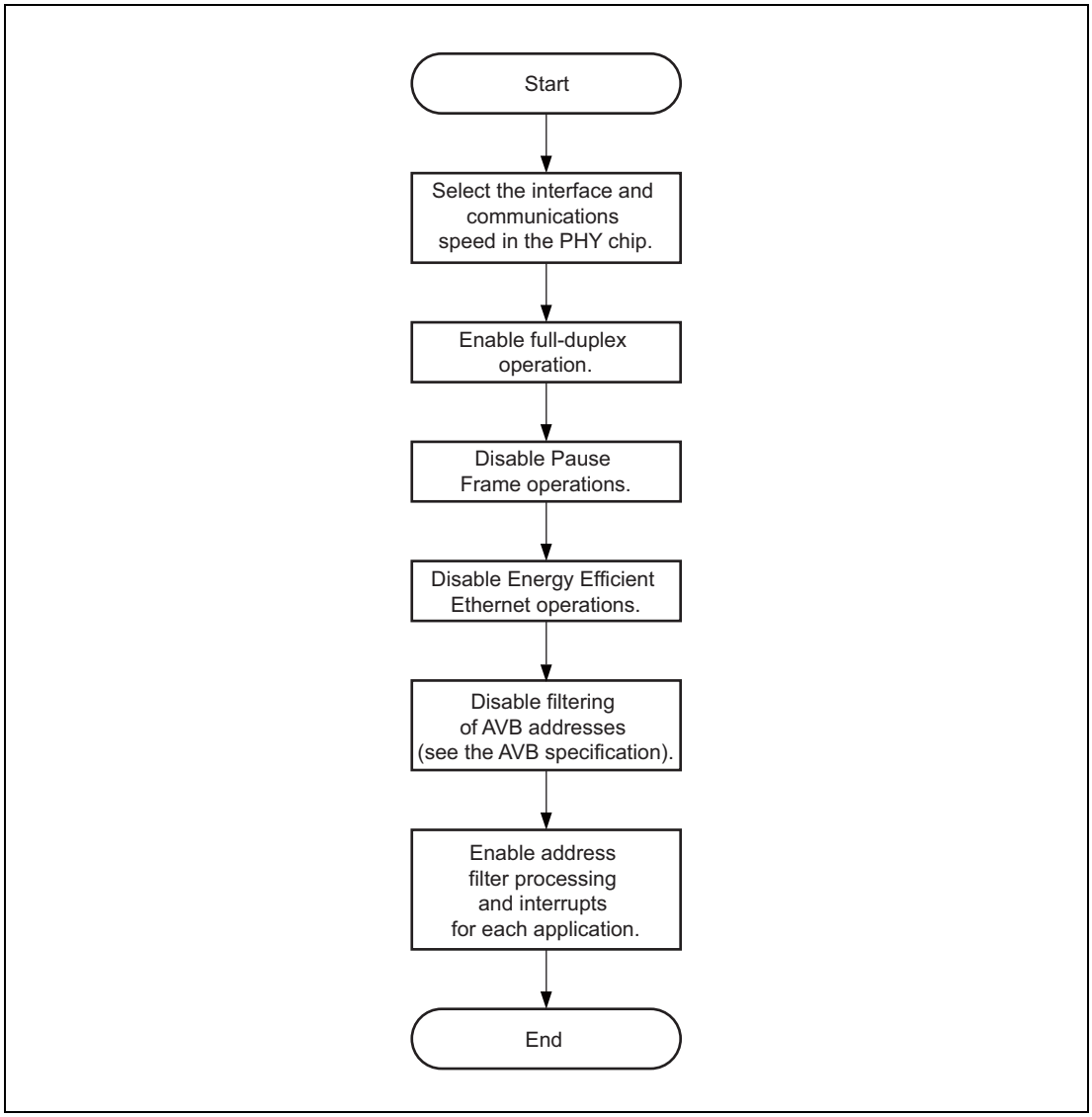


Figure 24.10 Procedure for Initializing the E-MAC Section

(4) Initialization of the Application Unit

Figure 24.11 illustrates initialization of the application unit.

For a description of how to set up the descriptors and the CBS traffic shaping parameters, see Section 24.4.3, Descriptors, and Section 24.4.6, CBS (Credit-Based Shaping).

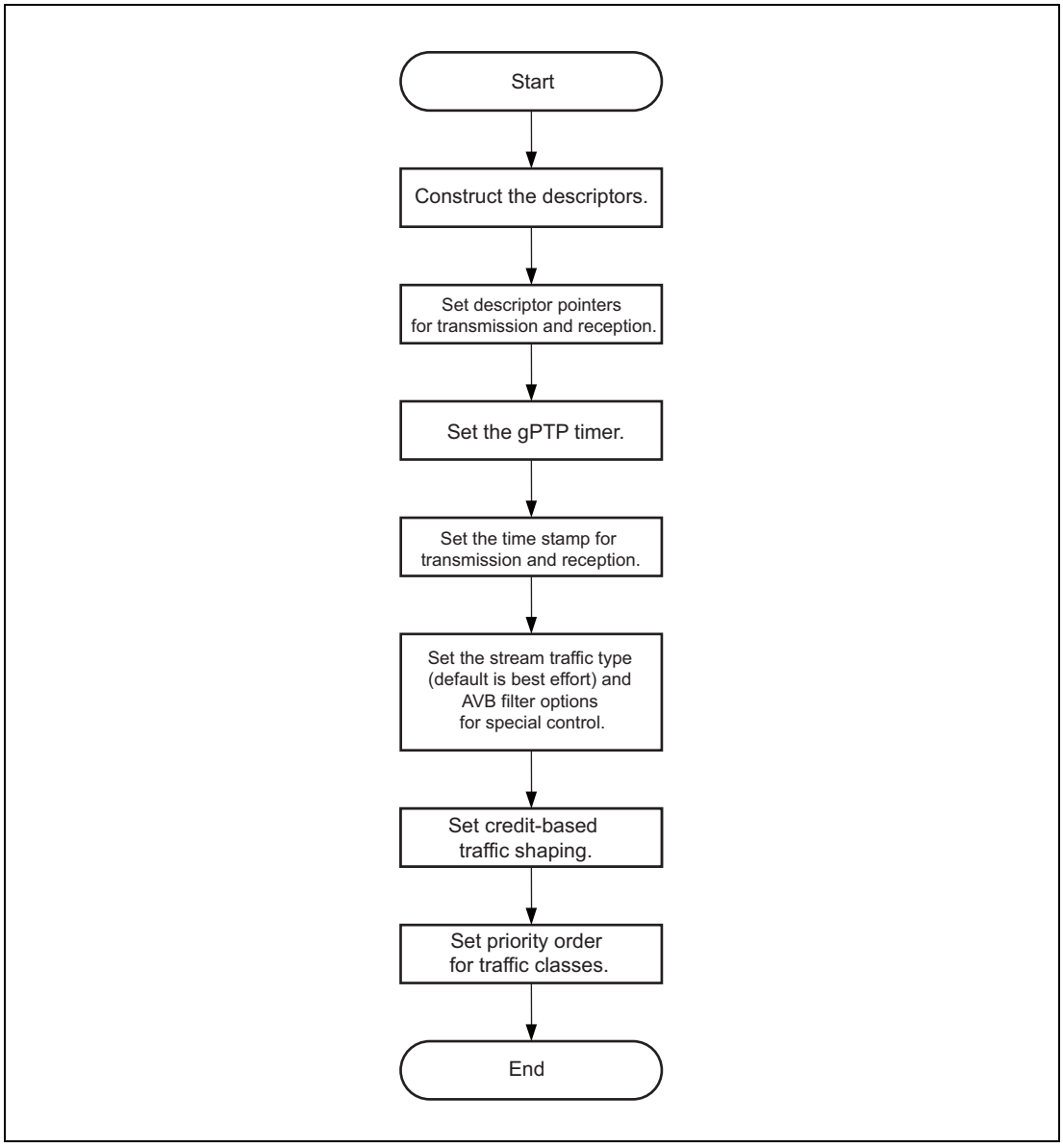


Figure 24.11 Initializing the Sections for Use by the Application

24.4.2.2 Scheduling Reception and Transmission

The AVB-DMAC normally has independent buses for transmission and reception. Furthermore, the four processes of fetching, storing, transmission and reception are basically independent of one another. Fetching and storing, however, share the same bus master so cannot be executed simultaneously. Access to the bus master is controlled by the scheduler.

Figure 24.12 is a schematic view of AVB-DMAC operations in transmission and reception.

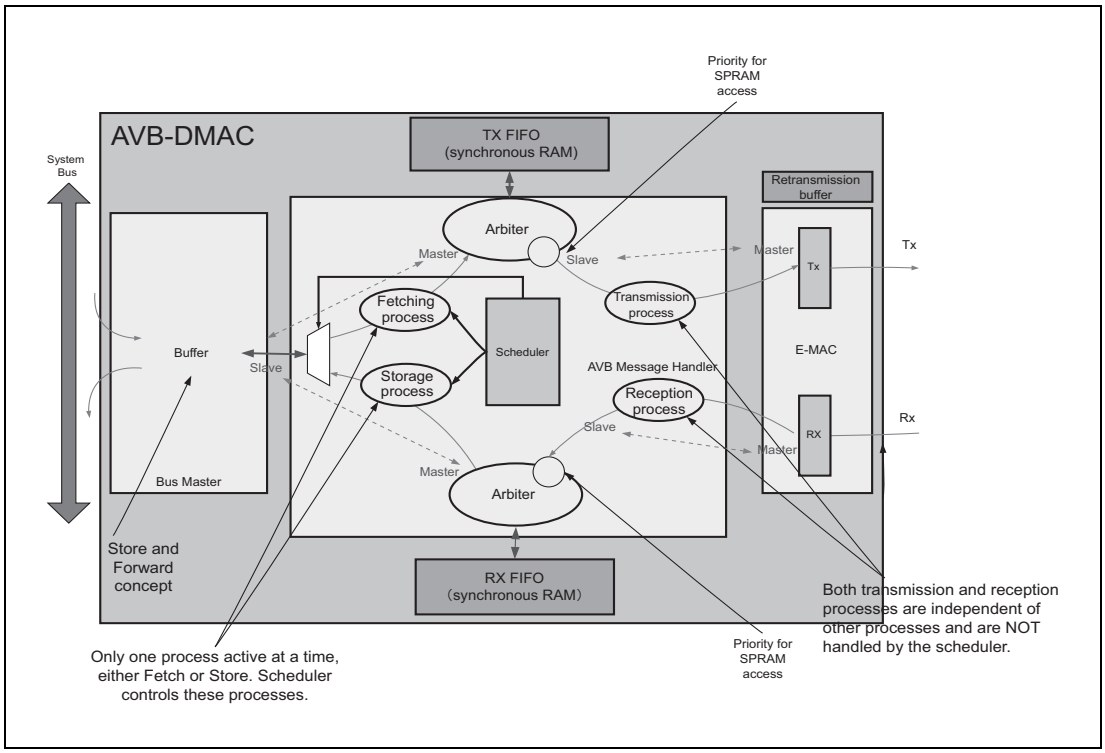


Figure 24.12 Schematic View of AVB-DMAC Operations in Transmission and Reception

Storing and fetching are alternately performed. When the number of frames held by the reception FIFO reaches the warning level, storing takes precedence over fetching.

(1) Relationship between Transmission Queue Numbers and Traffic Classes

In fetching, the relationships between the transmission queues and traffic classes are fixed, so the priority specified by the transmit queue priority bits in the transmit configuration register (TGC.TQP[1:0]) has no effect.

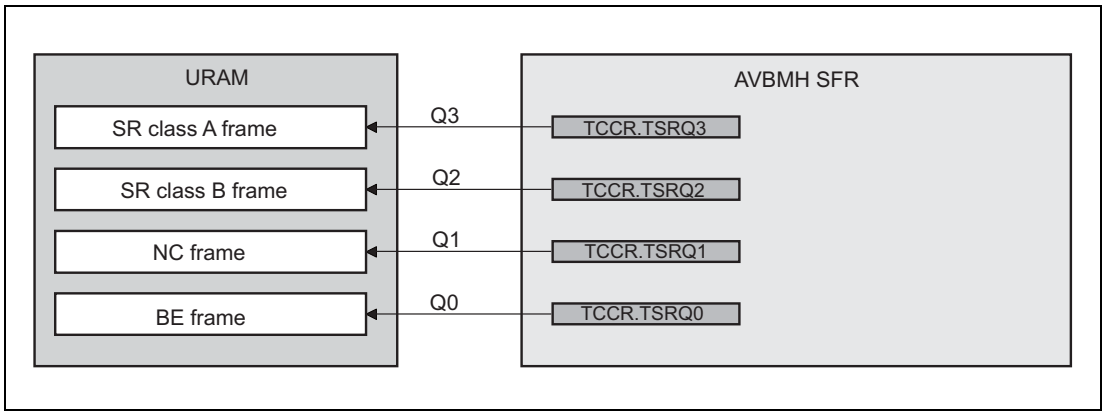


Figure 24.13 Class Associations of Queues for the Scheduler

In fetching, the credit values for stream classes A and B are not taken into account. Behavior depends on the setting of the transfer FIFO size configuration bits in the transfer configuration registers (TGC.TBDt) and on the frame size that can be fetched to the transmission FIFO.

When the transmit queue priority bits in the transmit configuration register (TGC.TQP[1:0]) are B’00 or B'01, the priority order is Q3 → Q 2 → Q1 → Q0.

When the transmit queue priority bits in the transmit configuration register (TGC.TQP[1:0]) are B’11, the priority order is Q1 → Q3 → Q2 → Q0.

24.4.2.3 Checking Integrity

The AVB-DMAC is capable of detecting and identifying errors produced in the processing of Ethernet frames and in the transfer of frame data for transmission and reception.

(1) Concept of Integrity Checking in Reception

The aim of integrity checking in reception is preventing the storage of error frames in the URAM.

If an error frame is stored, information to identify the frame as an error frame is appended to the data from the frame in the URAM.

CAUTION

If a special descriptor chain is to be used for separation of received headers from the associated data, an error that breaks the sequence may lead to storage space for synchronization running out. In such cases, software interaction or re-synchronization via the EOS descriptor is required.

(2) Concept of Integrity Checking in Transmission

The purpose of integrity checking in transmission is to prevent the transmission of broken frames.

Since transmission of a frame by the E-MAC can neither be stopped nor disabled once it has started, this check involves intensive monitoring for problems that can arise during fetching.

(3) Items for Monitoring in Both Reception and Transmission

(a) Errors in access to the URAM for reading of descriptors

The same descriptor may be processed again because the current descriptor address (CDARq.CDA[31:0]) was not changed.

If this problem occurs in a divided frame, the sequence may be broken.

In reception

- The received frame will be lost.
- The same problem will occur for the next frame of data received for the same queue.

In transmission

- The transmit start request bit in the transmit configuration control register (TCCR.TSRQt) is set to 0.
- The frame will be lost from the transmission FIFO.

Errors in access to read descriptors from the URAM are detected from the response signal of the AXI-Bus.

(b) Illegal configuration of a descriptor by an application

The same descriptor may be processed again because the current descriptor address (CDARq.CDA[31:0]) was not changed.

If this problem occurs in a divided frame, the sequence may be broken.

In reception

- The received frame will be lost.
- The same problem will occur for the next frame of data received for the same queue.

In transmission

- The transmit start request bit in the transmit configuration control register (TCCR.TSRQt) is set to 0.
- The frame will be lost from the transmission FIFO.

(c) Errors in access to the URAM for writing of descriptors

As in the case where no error occurs, the current descriptor address (CDARq.CDA[31:0]) and the transmit start request bit in the transmit configuration control register (TCCR.TSRQt) are updated.

As DESCR.DT was not updated, hardware and software synchronization may have been destroyed.

Errors in access to write descriptors to the URAM are detected from the response signal of the AXI-Bus.

(4) Items for Monitoring in Reception

(a) Errors in access to the URAM for writing of data or time stamps

- As in the case where no error occurs, the current descriptor address (CDARq.CDA[31:0]) is updated.
- DESCR.EI is set to indicate incorrect contents.
- This problem occurring in a divided frame may break the descriptor sequence, making the queue unusable.

Errors in access to write data or descriptors to the URAM are detected from the response signal of the AXI-Bus.

(b) Errors in Access for Reading from the Reception FIFO

- As in the case where no error occurs, the current descriptor address (CDARq.CDA[31:0]) is updated.
- DESCR.EI is set to indicate incorrect contents.
- This problem occurring in a divided frame may break the descriptor sequence, making the queue unusable.
- Errors of this type are detected by the ECC checker for the reception FIFO.

(c) Damaged Data in the Reception FIFO

- Received frames are all invalidated.
- All frames stored as received frames are discarded. At this time, the number of frames and queue information cannot be captured.

If damaged data in the reception FIFO is an error due to the reception FIFO, this is detected by the AVB-DMAC.

(5) Items for Monitoring in Transmission

(a) Errors in Access for Reading Data from the URAM

- Data that have already been fetched are discarded from the transmission FIFO.
- When an error of this type occurs during processing of an FSINGLE or FEND descriptor:
  As in the case where no error occurs, the current descriptor address (CDARq.CDA[31:0]) and the transmit start bit in the transmit configuration control register (TCCR.TSRQt) are updated.
  Fetching resumes after the error frame.
- When an error of this type occurs during processing of an FSTART or FMID descriptor:
  - The current descriptor address (CDARq.CDA[31:0]) is not updated.
  - The transmit start bit in the transmit configuration control register (TCCR.TSRQt) is set to 0.

Errors in access to read data from the URAM are detected from the response signal of the AXI-Bus.

(b) Overflow of the Transmission FIFO

- As in the case where no error occurs, the current descriptor address (CDARq.CDA[31:0]) and the transmit start bit in the transmit configuration control register (TCCR.TSRQt) are updated.
  Fetching resumes after the error frame.
- The frame will be discarded from the FIFO.

(c) Errors in Access for Reading of the Transmission FIFO

The AVB-DMAC is incapable of detecting an error in reading of the transmission FIFO, resulting in the transmission of a broken frame.

Errors of this type are detected by the ECC checker for the transmission FIFO.

(d) Frame size error during transmission

- As in the case where no error occurs, the current descriptor address (CDARq.CDA[31:0]) and the transmit start bit in the transmit configuration control register (TCCR.TSRQt) are updated.
  Fetching resumes after the error frame.

A transmit frame size error is detected when the size setting in one or more (in the case of a divided frame) descriptors for frame transmission is 1966 or more bytes. Such frames are cut out and transmitted.

(e) Damaged Data in the Transmission FIFO

- Fetching is not affected by damaged data.
- Since damaged data from the FIFO is only detected during frame transmission, an error frame may be transmitted.

If damaged data in the transmission FIFO is an error due to the transmission FIFO, this is detected by the AVB-DMAC.

24.4.3 Descriptors

24.4.3.1 Data Representation in URAM

The AVB-DMAC transfers data for transmission and received data to and from the application software via the URAM.

The memory in the URAM for use by the AVB-DMAC is configured with control structures referred to as descriptors and associated areas to which the frame data are allocated. Dividing the memory into a control area and data area allows the flexible allocation of frame data to the URAM. This enables sharing of the areas to which frame data are allocated and the use of non-contiguous areas. Frame data can be copied without using the CPU. Arbitration that ensures hardware and software access to the memory area is also available without using the AVB-DMAC.

Figure 24.14 shows an example of the memory maps for descriptors and the descriptor data area in the URAM.

A descriptor consists of its type (DESCR.DT), which controls the descriptor functions, a descriptor pointer (DESCR.DPTR) indicating the start address for storage of the frame data in the descriptor area, and the data size field (DESCR.DS), indicating the amount of frame data. Post-processing interrupt generation can be set up for each descriptor. Enabling and disabling of the interrupt is controlled by the descriptor interrupt enable bits (DESCR.DIE).

The descriptor may also hold information related to content. This information does not affect general descriptor functions. It provides information other than the frame data proper, such as on the state of reception.

For details, see Section 24.4.4.2, Setting Up Reception Descriptors, and **Section 24.4.5.2, Setting Up Transmission Descriptors**.

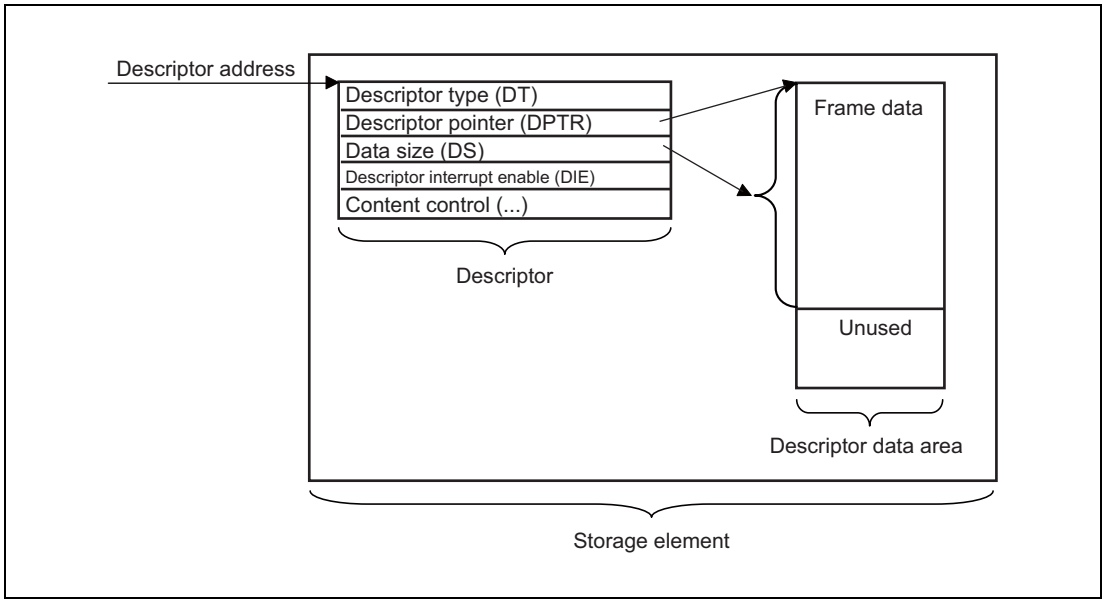


Figure 24.14 Example of URAM Memory Map

The descriptor must be aligned with a 32-bit boundary in the URAM.

Descriptors are generally configured of 64 bits, but are configured of 160 bits when reception and storage of gPTP time stamps is enabled.

The frame data must also be aligned with a 32-bit boundary in the URAM.

The amount of data in the frame is defined by the data size bits (DESCR.DS). In reception, these bits indicate the upper limit on the size of frames to be received. If the data size is not aligned with a 32-bit boundary, the bytes to the next 32-bit boundary in the data area will be an unused area.

24.4.3.2 Using Descriptor Chains in Queues

Transmission and reception descriptors in the URAM are grouped into queues. Each queue handles frames so that they are transmitted in order of priority and received separately. A queue is capable of controlling one or more frames. Accordingly, multiple descriptors can be assigned to one queue. A combination of multiple descriptors is referred to as a descriptor chain.

For a descriptor chain, the three general descriptor types listed below are defined. For details on these descriptor types, see Section 24.4.3.6, Descriptor Type.

- Descriptors that define frame data
- Descriptors that control the descriptor chain itself (e.g. LINK, EOS).
- Descriptors that arbitrate access by hardware or software

Figure 24.15 shows the two basic topologies for descriptor chains. In the simplified examples in the figure, all descriptors allocated to the chain are stored in the array.

- For a linear descriptor chain, the last descriptor in the array is a control descriptor indicating the end of the descriptors (e.g. EEMPTY).
- For a cyclic descriptor chain, the last descriptor in the array is a control descriptor that returns to the first descriptor in the array (e.g. LINK).

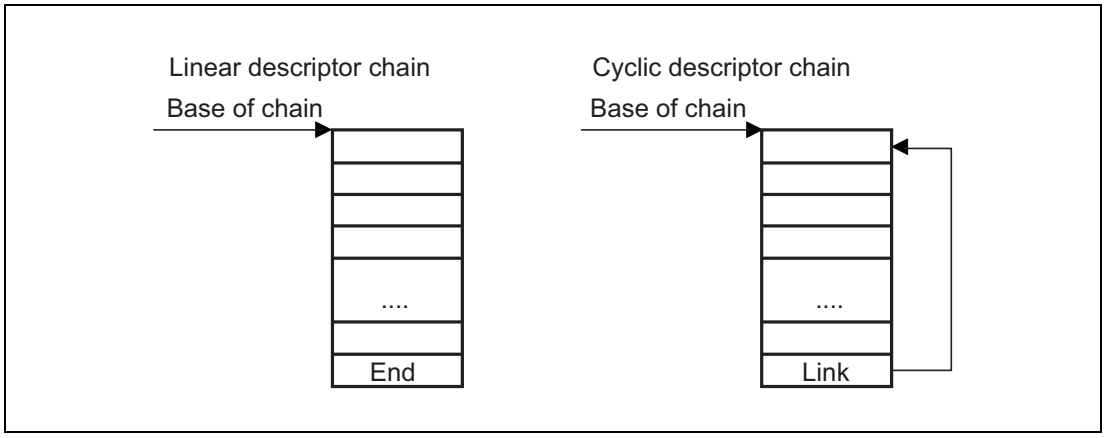


Figure 24.15 Outline of the Basic Descriptor Chains

The relationship between queues and descriptor chains is defined by the base addresses of chains. A queue is connected to one descriptor chain over one round of processing. There is also a method of switching to a different chain while in operation mode.

There are no restrictions on the number of link descriptors and their locations within the chain. The last descriptor of a designed chain determines the topology.

Which chain structure is to be used or which topology is suitable depends on the application. A description of how to design descriptor chains to suit various applications is given in **Section 24.4.4.2**. Procedure for Setting Reception Descriptors, and Section 24.4.5.2, Setting Up Transmission Descriptors.

24.4.3.3 Descriptor Base Address Table

The base address table in the URAM contains the address of the first descriptor of all chains to be handled by the respective queues.

Entries 0 to 3 are used to access transmission queues 0 to 3. Subsequent entries are used to access reception queues. Entry 4 thus corresponds to reception queue 0.

The configuration of entries in the base address table is the same as the configuration of link descriptors. We recommend using the descriptor type (DESCR.DT) LINKFIX. Processing of this link descriptor does not change it, so it does not require updating. The first descriptor of a chain performs hardware and software synchronization. If the application requires hardware and software synchronization for the base addresses, use the descriptor type (DESCR.DT) LINK.

The CPU is only capable of using LINKFIX and LINK as descriptor types (DESCR.DT) of descriptors in the base address table.

Set the location of the base address table in the URAM in the descriptor base address table register (DBAT).

Figure 24.16 shows an example of a base address table for controlling four transmission and three reception queues. The boxes to the right of the table represent descriptor chains with the desired topologies.

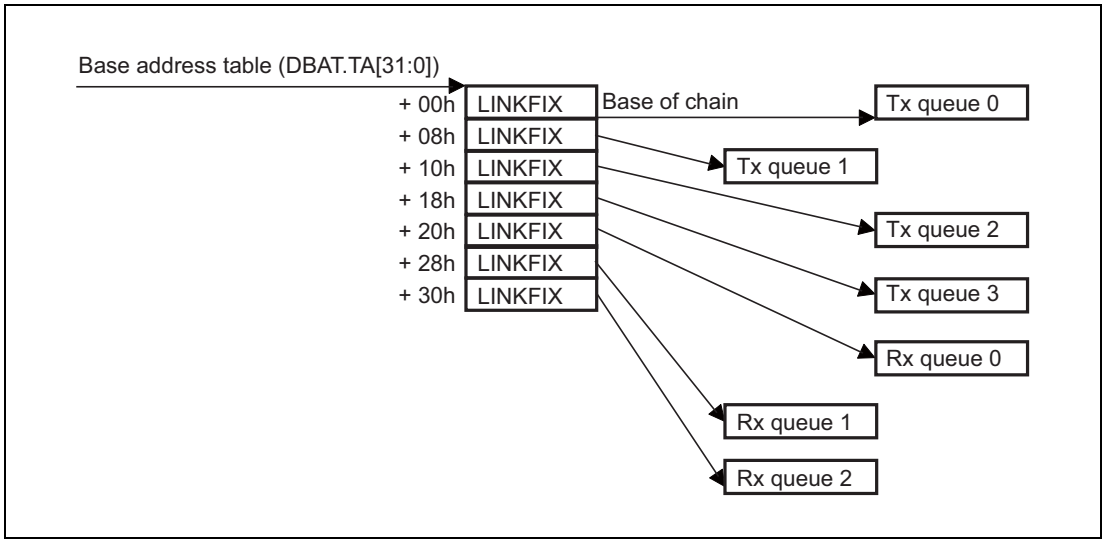


Figure 24.16 Example of a Base Address Table for Reception and Transmission Queues

CAUTION

The size of the descriptors in the base address table is always eight bytes even if the queue itself includes extended descriptors.

24.4.3.4 Descriptor Chain Processing

The descriptor that is currently processed or will be processed when the related queue gets active is the current descriptor. The current descriptor address for use by a queue q can be checked in the current descriptor address register q (CDARq).

Current descriptors are stored in registers or in descriptors as described below in the given situations.

- In the descriptor base address table registers for all q queues (DBAT) (DBAT.TA[31:0]+8*q) when the operating mode shifts to operation mode.
- In the descriptor base address table register (DBAT) (DBAT.TA[31:0]+8*q) when a base address load request is issued for a queue q by setting the corresponding bit (DLR.LBAq) in the descriptor base address load request register (DLR).
- In DESCR.DPTR for a link descriptor (LINK, LINKFIX) to be processed.

After a descriptor has been processed, the current descriptor for the same queue is incremented by the size of the descriptors being handled by the queue (8 bytes for normal descriptors and 20 bytes for extended descriptors). The AVB-DMAC updates the descriptor type and informs the CPU that the descriptor has been processed.

24.4.3.5 Descriptor Interrupts

A descriptor is able to issue a descriptor interrupt on completion of its processing. The setting of the descriptor interrupt enable bits (DESCR.DIE) in each descriptor selects disabling or generation of the descriptor interrupt.

The descriptor interrupt is a common resource that is shared between reception and transmission queues. Software control of the descriptor interrupt provides a flexible method of application-specific flag processing.

Figure 24.17 illustrates the way in which the AVB-DMAC generates descriptor interrupts (or sets bits in the descriptor interrupt status register (DIS.DPFi)). Processing of a descriptor with the value i in the descriptor interrupt enable bits (DESCR.DIE) leads to the corresponding bit in the descriptor interrupt status register (DIS.DPFi) being set.

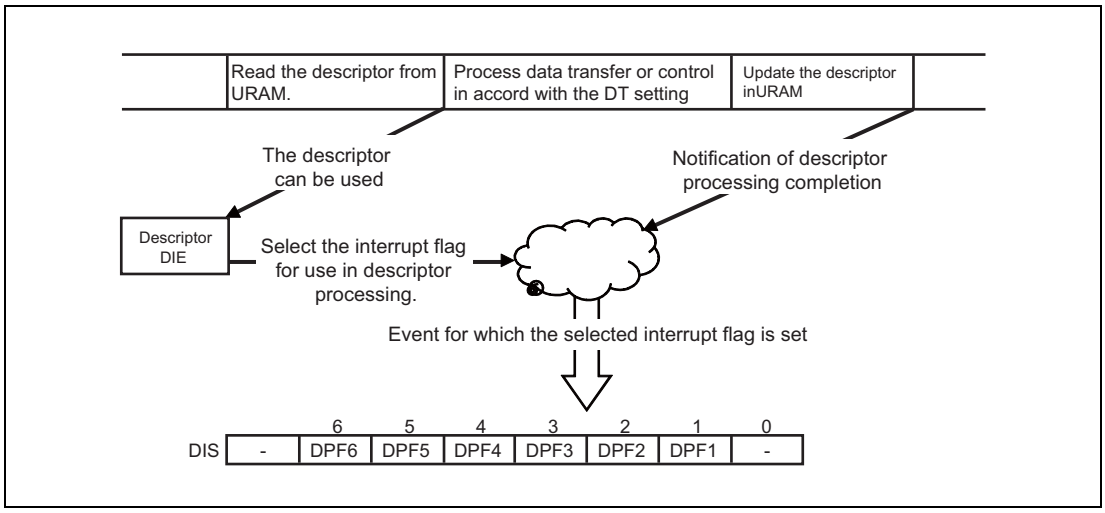


Figure 24.17 Method of Descriptor Interrupt Generation

24.4.3.6 Descriptor Type

The descriptor types (indicated by the DESCR.DT bits) supported by the AVB-DMAC fall into the following three categories.

- Definitions of frame data
- Control of descriptor chains
- Hardware and software arbitration

Table 24.77 is a summary of the descriptor types available for the AVB-DMAC. Entries under “Name” are the names of the descriptor types and the values under “DT” are the corresponding values to be set in the descriptor type field (DESCR.DT). A given descriptor may be handled differently according to whether it is in a transmission or reception queue, so the transmission and reception columns list the scopes of control and processing of the descriptor types.

The abbreviations defined below are used in the transmission and reception columns.

Definition of SW:

- The descriptor is processed by software.
- Software has access to and may modify the descriptor and descriptor data area.
- This descriptor cannot be changed by hardware (AVB-DMAC).

Definition of HW:

- The descriptor is processed by hardware (AVB-DMAC).
- Software must modify neither the descriptor nor the descriptor data area.
- Hardware (AVB-DMAC) processes this descriptor and subsequently changes the descriptor type.

Invalid:

This descriptor type is not used in transfer in the given direction (transmission or reception).

Do not write this value to the descriptor type (DESCR.DT) field for transfer in the given direction.

Hardware does not process these descriptor types in the cases listed as invalid. The current descriptor address (CDARq.CDA[31:0]) will not be changed when processing of a queue for the given direction arrives at a descriptor with this type setting.

Table 24.77 Summary of Descriptor Types

Name	DT	Description	Reception	Transmission
Frame data				
FSTART	5	Frame Start The descriptor points to valid data for a frame. The frame starts with the given data and continues with that indicated by the next descriptor.	SW	HW
FMID	4	Frame Middle The descriptor points to valid data for a frame. The frame started with a previous descriptor and continues to the data indicated by the next descriptor.	SW	HW
FEND	6	Frame End The descriptor points to valid data for a frame. The frame continues from the previous descriptor and ends with the data indicated by in this descriptor.	SW	HW
FSINGLE	7	Frame Single The descriptor points to valid data for a complete frame.	SW	HW
Chain control				
LINK	8	Link Defines the next descriptor in the chain.	HW	HW
LINKFIX	9	Fixed Link Defines the next descriptor in the chain.	SW	SW
EOS	10	End Of Set Defines the next descriptor in the chain.	HW	HW
HW/SW arbitration				
FEMPTY	12	Frame Empty A descriptor related to frame data but not containing valid data for a frame	HW	SW
FEMPTY_IS	13	Frame Empty Incremental Start A descriptor related to frame data but not containing valid data for a frame DESCR.DPTR sets the base address of an "incremental data area" in the URAM.	HW	Invalid
FEMPTY_IC	14	Frame Empty Incremental Continue A descriptor related to frame data but not containing valid data for a frame Data is stored to the incremental data area in the URAM.	HW	Invalid
FEMPTY_ND	15	Frame Empty No Data storage A descriptor related to frame data but not containing valid data for a frame The descriptor is processed in the same way as FEMPTY but data are not stored in the URAM.	HW	Invalid
LEMPTY	2	Link Empty A link descriptor for processing by the AVB-DMAC	SW	SW
EEMPTY	3	EOS Empty An EOS descriptor for processing by the AVB-DMAC	SW	SW
DT0	0	Reserved	Invalid	Invalid
DT1	1	Reserved	Invalid	Invalid
DT11	11	Reserved	Invalid	Invalid

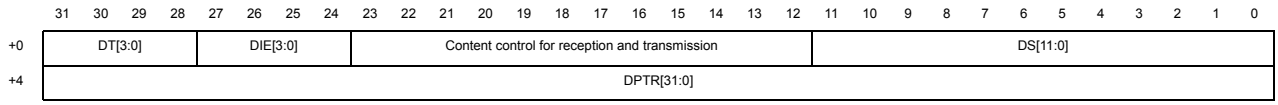
(1) Layout of General Descriptors in the URAM

The AVB-DMAC updates processed descriptors in the URAM. The field to be changed in a descriptor being updated depends upon whether the direction is transmission or reception and the queue mode. Other fields will not be changed. There are no restrictions on the values set in unused descriptor fields (indicated by "—" in the figure).

(2) Frame Data Descriptors

The allocation of bits in the frame data descriptors (FSTART, FMID, FEND, and FSINGLE) is shown below.

- Normal descriptor (usable in both reception and transmission)



- Extended descriptor (usable only in reception)

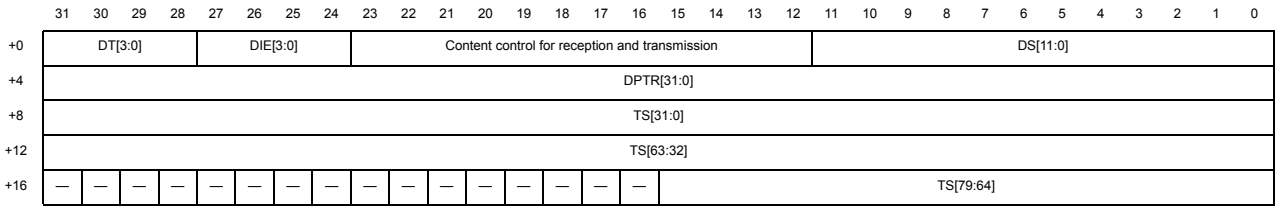


Table 24.78 Contents of Frame Data Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 5: FSTART 4: FMID 6: FEND 7: FSINGLE For details, see Section 24.4.4.2, Setting Up Reception Descriptors, and Section 24.4.5.2, Setting Up Transmission Descriptors.
DIE[3:0]	Descriptor Interrupt Enable B'0000: Descriptor interrupt is disabled. B'0001 to B'1111: The corresponding descriptor interrupt is generated (DIS.DPFi).
—	Content Control For details, see Section 24.4.4.2, Setting Up Reception Descriptors, and Section 24.4.5.2, Setting Up Transmission Descriptors.
DS[11:0]	Data Size Size of the data area/frame data for the descriptor (in bytes)
DPTR[31:0]	Descriptor Pointer Pointer to the data area for the descriptor Register an address on a 32-bit boundary.
TS[79:0]	Time Stamp Time stamp of the received frame (only available in extended descriptors)

CAUTION

Register an address aligned with a 32-bit boundary as the descriptor pointer (DESCR.DPTR).

Some bits in extended descriptors are reserved (the DESCR.TS[79:64] bits in an

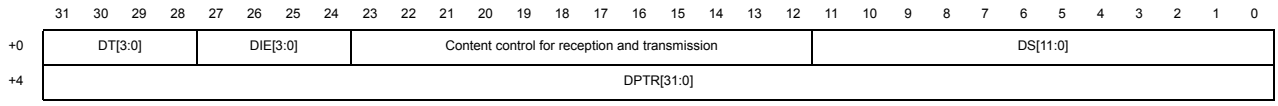
extended descriptor are set to H'0000 after the time stamp is stored).

(3) Hardware/Software Arbitration Descriptors (Only for Reception)

The allocation of bits in the descriptors for hardware/software arbitration (FEMPTY, FEMPTY_IS, FEMPTY_IC, and FEMPTY_ND) is shown below.

The allocation of bits in the arbitration descriptors for use in reception is the same as in frame data descriptors.

- Normal descriptor



- Extended descriptor (usable only in reception)

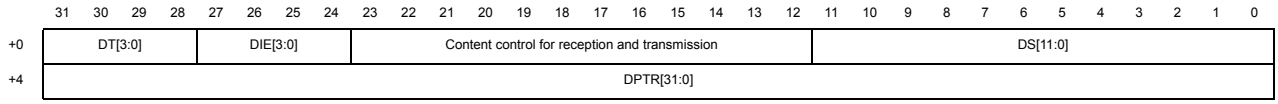


Table 24.79 Contents of Hardware/Software Arbitration Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 12: FEMPTY 13: FEMPTY_IS 14: FEMPTY_IC 15: FEMPTY_ND For details, see Table 24.77, Summary of Descriptor Types.
DIE[3:0]	Descriptor Interrupt Enable B'0000: Descriptor interrupt is disabled. B'0001 to B'1111: The corresponding descriptor interrupt is generated (DIS.DPFi).
—	Content Control For details, see Section 24.4.4.2, Setting Up Reception Descriptors, and Section 24.4.5.2, Setting Up Transmission Descriptors.
DS[11:0]	Data Size Size of the data area/frame data for the descriptor (in bytes)
DPTR[31:0]	Descriptor Pointer Pointer to the data area for the descriptor Register an address on a 32-bit boundary.

CAUTIONS

Register an address aligned with a 32-bit boundary as the descriptor pointer (DESCR.DPTR).

When the descriptor is an extended descriptor, it has a 12-byte unused area.

In an FEMPTY descriptor, the descriptor type (DT), descriptor interrupt enable (DIE), data size (DS), and descriptor pointer (DPTR) fields are used.

In an FEMPTY_IS descriptor, the descriptor type (DT), descriptor interrupt enable (DIE), and descriptor pointer (DPTR) fields are used.

In an FEMPTY_IC descriptor, the descriptor type (DT) and descriptor interrupt enable (DIE) are used.

In an FEMPTY_ND descriptor, the descriptor type (DT), descriptor interrupt enable (DIE), and data size (DS) are used.

(4) Link Descriptors

The allocation of bits in the link descriptors (LINK and LINKFIX) is shown below.

- Normal descriptor

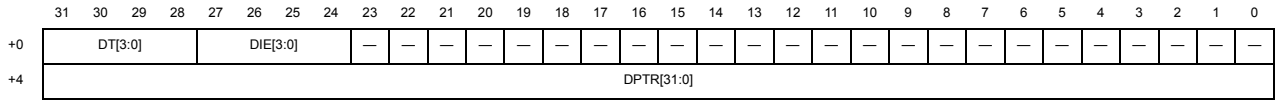


Table 24.80 Contents of Link Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 8: LINK 9: LINKFIX For details, see Table 24.77, Summary of Descriptor Types.
DIE[3:0]	Descriptor Interrupt Enable B'0000: Descriptor interrupt is disabled. B'0001 to B'1111: The corresponding descriptor interrupt is generated (DIS.DPFi).
—	Content Control For details, see Section 24.4.4.2, Setting Up Reception Descriptors, and Section 24.4.5.2, Setting Up Transmission Descriptors.
DPTR[31:0]	Descriptor Pointer Pointer to the data area for the descriptor Register an address on a 32-bit boundary.

CAUTION

Register an address aligned with a 32-bit boundary as the descriptor pointer (DESCR.DPTR).

(5) Other Descriptors

The allocation of bits in the other descriptors (EOS, FEMPTY (only for transmission), LEMPTY, and EEMPTY) is shown below.

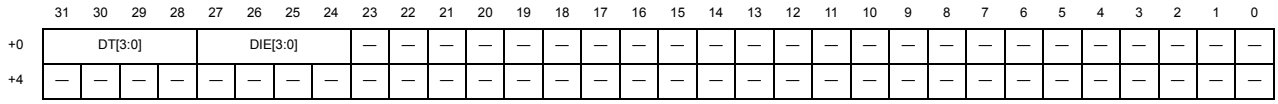


Table 24.81 Contents of Other Descriptors (DESCR)

Bit Name	Function
DT[3:0]	Descriptor Type 10: EOS 12: FEMPTY (only for transmission) 2: LEMPTY 3: EEMPTY For details, see Table 24.77, Summary of Descriptor Types.
DIE[3:0]	Descriptor Interrupt Enable B'0000: Descriptor interrupt is disabled. B'0001 to B'1111: The corresponding descriptor interrupt is generated (DIS.DPFi).

(6) How to Use Frame Data Descriptors

The descriptor data area size field (DESCR.DS) can specify up to 2048 bytes of Ethernet frame data per data area. Settings higher than 2048 (bytes) cannot be made.

In general, Ethernet frames are not of uniform length. The AVB-DMAC is capable of dividing frame data into multiple descriptors in order to minimize the memory capacity for frame data. This function allows processing of frames that are longer than the limit for descriptor data areas. Division can also be applied to frames on the basis of their data structures.

To handle both frames divided up into multiple data areas and descriptors for complete single frames, four types (DESCR.DT) FSTART, FEND, FMID, and FSINGLE are defined.

Figure 24.18 shows the mapping of frame data by frame data descriptors. The descriptor data areas are allocated to the URAM. For frames that require division into four or more data areas, additional FMID descriptors can be added as required.

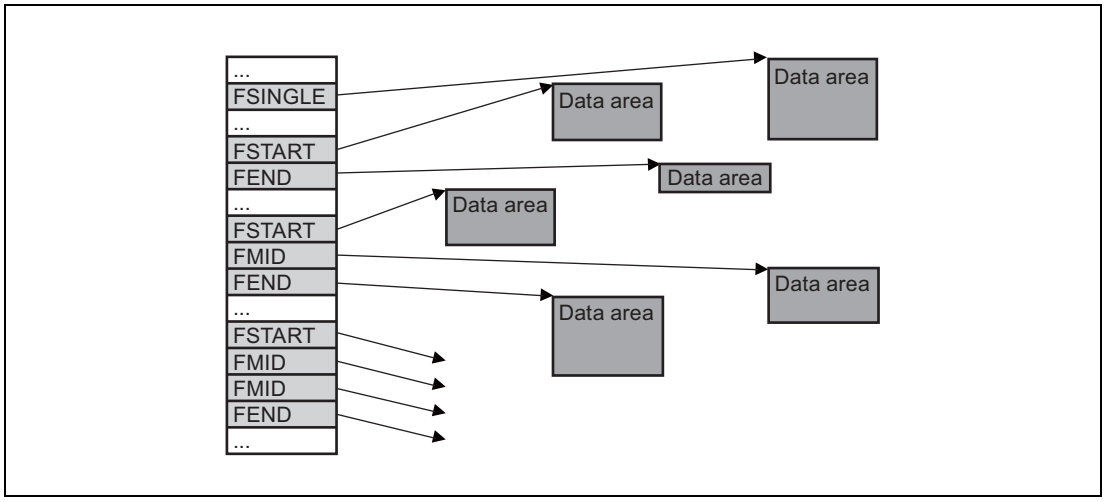


Figure 24.18 Mapping of Frame Data

For reception, set the descriptor data areas to the maximum size (i.e. give DESCR.DS its maximum value). The AVB-DMAC will store received frame data in the given area. If a received frame has more data than the maximum size, the AVB-DMAC will divide the data up.

For transmission, set the frame data size to the actual data size. The AVB-DMAC modifies the descriptor type (DESCR.DT) to FEMPTY after processing the relevant descriptor. The data size (DESCR.DS) and descriptor pointer (DESCR.PTR) fields retain their settings.

A descriptor data area including unused space produces an empty space between data areas. In reception, an "incremental data area" can be used to prevent empty spaces. For incremental data areas, see **Section 24.4.4.3** (2) Incremental Data Areas.

As well as reducing the memory capacity taken up by the descriptor area in the URAM, division into frames can be used to identify different sections of data (e.g. for separating a header and data).

(7) How to Use Chain Control Descriptors

(a) Link Descriptors

The link descriptors can be used to set up cyclic descriptor chains (for details, see **Section 24.4.3.2, Using Descriptor Chains in Queues**)

After a LINK descriptor is processed, its descriptor type (DESCR.DT) is changed to LEMPTY. The descriptor pointer (DESCR.PTR) retains its setting.

After processing of a LINKFIX descriptor, the descriptor type (DESCR.DT) is not updated. Software can change the descriptor type (DESCR.DT), descriptor interrupt enable (DESCR.DIE), and descriptor pointer (DESCR.DPTR). However, DESCR.DT should not be modified by software. Take care to check the current descriptor address register (CDARq.CDA[31:0]) before changing the descriptor pointer (DESCR.DPTR).

(b) EOS Descriptor

Use the EOS descriptor to divide a descriptor chain into various segments. The queue can continue even after an EOS descriptor.

In transmission, the response to an EOS descriptor is clearing of the transmit start request bit in the transmit configuration control register (TCCR.TSRQq) to 0.

In reception, the response is generation of a receive queue full interrupt (RIS2.QFFr), although if the frame currently being received is being divided for storage (received data such as those where some storage is in FMID- or FEND-type frames), the data are not completely stored.

(8) How to Use Hardware and Software Arbitration Descriptors

In hardware processing of descriptors, the empty descriptor types (FEMPTY, LEMPTY, and EEMPTY) are used to distinguish various descriptors. For software, they can be used to initiate checking for empty spaces, etc.

(a) FEMPTY, FEMPTY_IS, FEMPTY_IC, and FEMPTY_ND

These descriptor types (DESCR.DT) are used for descriptors that do not contain effective data. Of these, only FEMPTY is used in transmission.

(b) LEMPTY

This descriptor type (DESCR.DT) is assigned to LINK descriptors after they have been processed.

The descriptor pointer (DESCR.DPTR) of an LEMPTY descriptor still points to the linked descriptor.

(c) EEMPTY

This descriptor type (DESCR.DT) is assigned to EOS descriptors after they have been processed.

The descriptor pointer (DESCR.DPTR) of an EEMPTY descriptor is not used.

(9) Synchronization between Descriptor Access by Hardware and Software

The allocation of descriptor types (DESCR.DT) to the URAM can be used to set up the primary synchronization between hardware and software. This makes it possible to minimize access by the AVB-DMAC to the SFR via the CPU, leading to higher performance.

Basic concepts of synchronization:

- Each descriptor type in the set is exclusively for processing by hardware or software, depending on the direction of transfer (see Table 24.77, Summary of Descriptor Types).
- Software must not change a descriptor assigned to hardware processing (the hardware does not change descriptors assigned to software processing).

In the case of software processing, the software must process the information in the descriptor and the corresponding frame data before changing the descriptor type. If a descriptor type for hardware is set in DESCR.DT, the software should not change any part of the descriptor or of the corresponding frame data.

24.4.3.7 Tips for Optimizing Performance in Handling Descriptors

The following items are recommended as ways to ensure the optimal use of data structures in the URAM.

They are not requirements, but using a different approach may increase the load on the system bus within the LSI chip.

- Register descriptors with 64-bit alignment (this does not apply to extended descriptors).
- While in operation mode, use LINKFIX instead of LINK whenever a descriptor need not be changed. Hardware modifies the descriptor type (DESCR.DT) fields of LINK descriptors.
- Restrict frame data to a maximum size of 128 bytes.
- Design the descriptor chains in ways that minimize parallelism of processing.
  This helps in dividing the chains into segments allocated to different cache pages, and in arranging the different segments exclusively for access by software or hardware.
- Minimize the number of divided frames. This can reduce the overhead of descriptor handling.

24.4.4 Control in Reception

The point of the AVB-DMAC is to transfer data between the E-MAC and URAM without intervention by the CPU.

Create descriptors that define the amounts of frame data to be stored and the locations. After the E-MAC receives a frame, it stores the received frame data and the conditions of reception as the MAC state. If the descriptor is extended, the time stamp is also stored. For a description of how to set up descriptors for use in reception, see Section 24.4.4.2, Setting Up Reception Descriptors.

The AVB-DMAC filters received frames to separate them into various classifications (separation filtering). More specifically, this is done to separate received frames into the various reception queues and to set the priorities of different classes of received frames. For more on separation filtering, see **Section 24.4.4.1** (1) Separation Filtering.

Figure 24.19 shows the reception data bus and the selection of queues for use in reception.

Each frame received from the E-MAC is stored in the reception FIFO; in parallel with this, the frame is analyzed to identify its type and the target queue number. After the E-MAC completes reception, the target queue number is generated and stored in the reception FIFO. Appending of a reception flag depends on the storage of one frame among the reception queues in the URAM, and the unread frame counter (UFC) is also associated with frame storage.

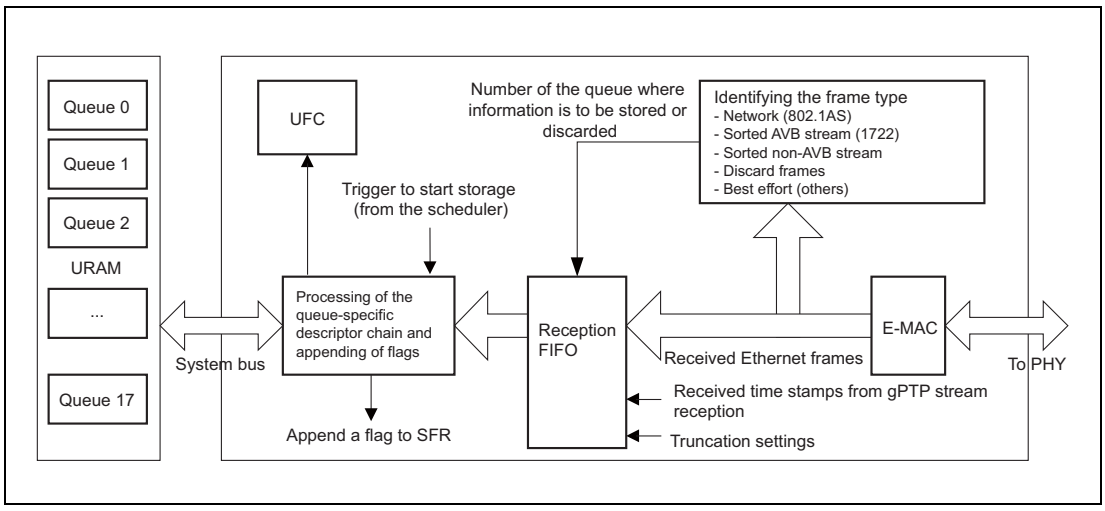


Figure 24.19 Mechanism of General Reception Queue Selection

24.4.4.1 Reception Queues

The AVB-DMAC applies its separation filtering mechanism to select the reception queue for storing a received frame. The AVB-DMAC stores all received frames in the URAM.

There are two conditions for the AVB-DMAC to discard a received frame.

- Detection of an error during reception by the E-MAC
  - Whether error frames are discarded or stored in reception queue 0 (best effort) depends on the setting of the error frame enable bit in the receive configuration register (RCR.EFFS). If error frames are to be stored (RCR.EFFS = 1), they are always stored in queue 0 (best effort). In this case, characteristics specific to the queue (e.g. truncation) will vary. If the storage of time stamps for reception queue 0 (best effort) is enabled (the time stamp enable bit in the receive configuration register RCR.ETS0 = 1), time stamps are stored even for error frames.
  - The separation filter is unable to determine where the frame data should be stored.
- Frame fails the separation filter
  - It depends on RCR.ESF[1:0] if such frame is discarded or stored in receive queue 0 (best effort).

The flowchart in **Figure 24.20** shows how the AVB-DMAC selects the reception queue in accord with the frame type, including judgment by the separation filter. Selection of the queue starts when the E-MAC completes frame reception. The result is storage of the frame in the proper queue or the frame being discarded.

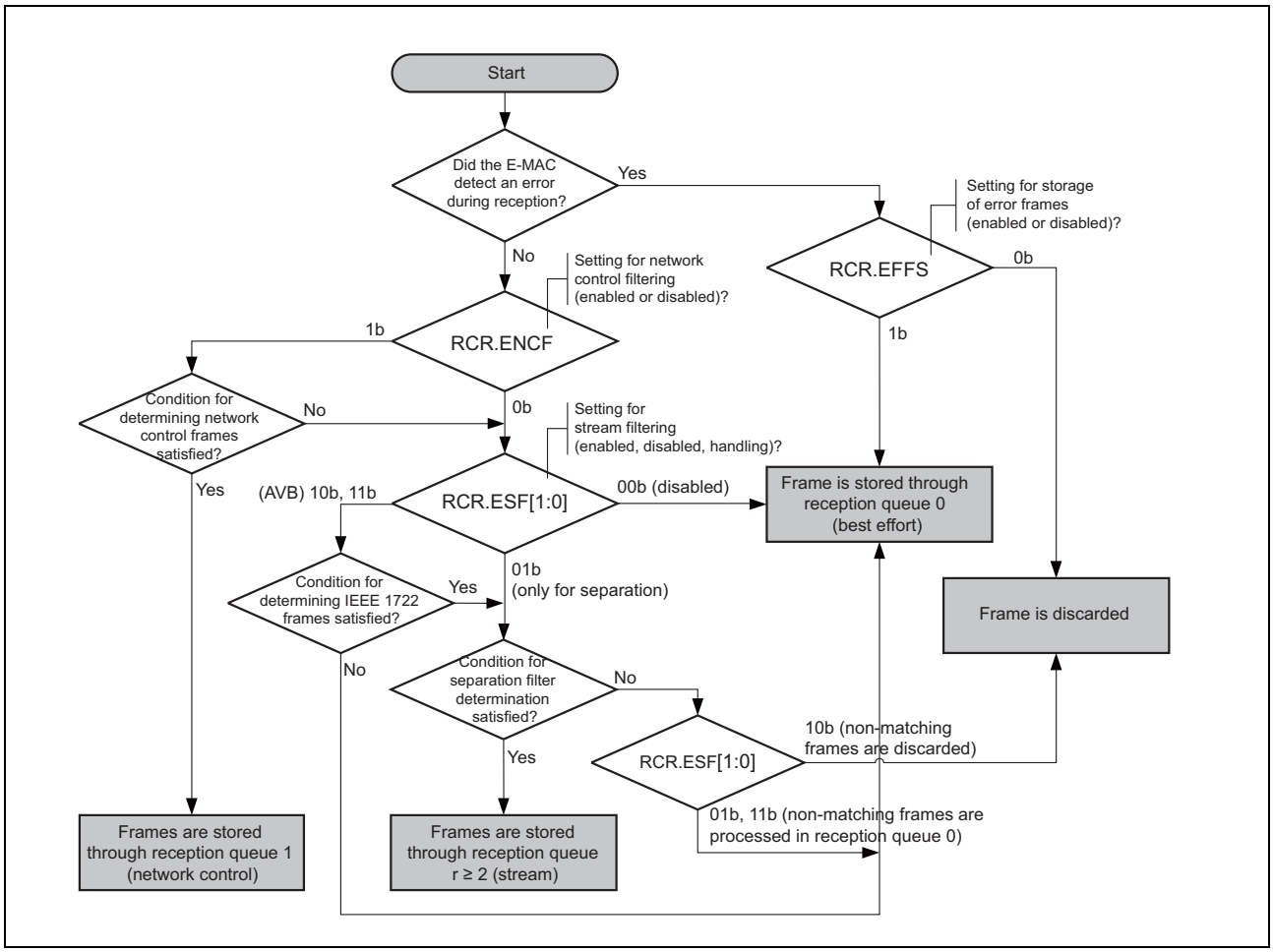


Figure 24.20 Mechanism of Reception Queue Selection

Notes on the meanings of entries in the flowchart

- "Condition for determining network control frames"
  The Ethernet destination address (DA) is 01:80:C2:00:00:0E.
  The Ethernet type (ET) is 88:F7.
- "Condition for determining IEEE 1722 frames"
  The Ethernet destination address (DA) is within the range from 91:E0:F0:00:00:00 to 91:E0:F0:00:FE:FF.
  The VLAN tagged TPID (tag protocol identifier) field (VL) is 81:00.
  The Ethernet type (ET) is 22:F0.
- "Condition for separation filter determination"
  See **Section 24.4.4.1** (1) Separation Filtering.

Figure 24.21 shows the allocation of bits related to the network and stream types in Ethernet frames. The preambles of Ethernet frames are not taken into account.

Data bytes	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Network type	DA1	DA2	DA3	DA4	DA5	DA6	SA1	SA2	SA3	SA4	SA5	SA6	ET1	ET2	...	...	...	...	...	...
Stream type	DA1	DA2	DA3	DA4	DA5	DA6	SA1	SA2	SA3	SA4	SA5	SA6	VL1	VL2	-	-	ET1	ET2	...	...

Figure 24.21 Data Bytes of Ethernet Frames Used in Classification

(1) Separation Filtering

Separation filtering involves the checking of up to 64 bits (eight successive bytes) in received Ethernet frames. The setting for the first byte (i.e. the setting of the separation filter offset configuration register (SFO.FBP[5:0])), selects the part of frames to be used in separation filtering. There is also a common filter mask (set in the separation filter mask configuration register (SFMi.CFM[31:0])) that can be freely set to reduce the number of bytes used in separation filtering or to mask particular bits.

Examples

To use one byte in separation, set separation filter mask configuration register 0 (SFM0.CFM[31:0]) to H'0000 00FF and separation filter mask configuration register 1 (SFM1.CFM[31:0]) to H'0000 0000.

To use seven bytes in separation, set separation filter mask configuration register 0 (SFM0.CFM[31:0]) to H'FFFF FFFF and separation filter mask configuration register 1 (SFM1.CFM[31:0]) to H'00FF FFFF.

CAUTION

If bits at some positions are set to 0b in the separation mask, in order to match with the pattern, the bits at the corresponding positions of the pattern must also be set to 0b. Only those bits in which the separation filter pattern configuration register (SFPi.FPs) setting is equal to the separation filter mask configuration register (SFMi.CFM[31:0]) are sorted by matching with received data.

Figure 24.22 shows separation filtering. The selected data from a received frame (Rx_Frame[63:0]) are masked by the common filter mask. As a result, the selected frame data can be obtained. This value is compared with all filter patterns. The separation filter circuit in the AVB-DMAC selects the filter pattern that matches the queue having the lowest index s or selects a flag to indicate that there is no matching separation pattern.

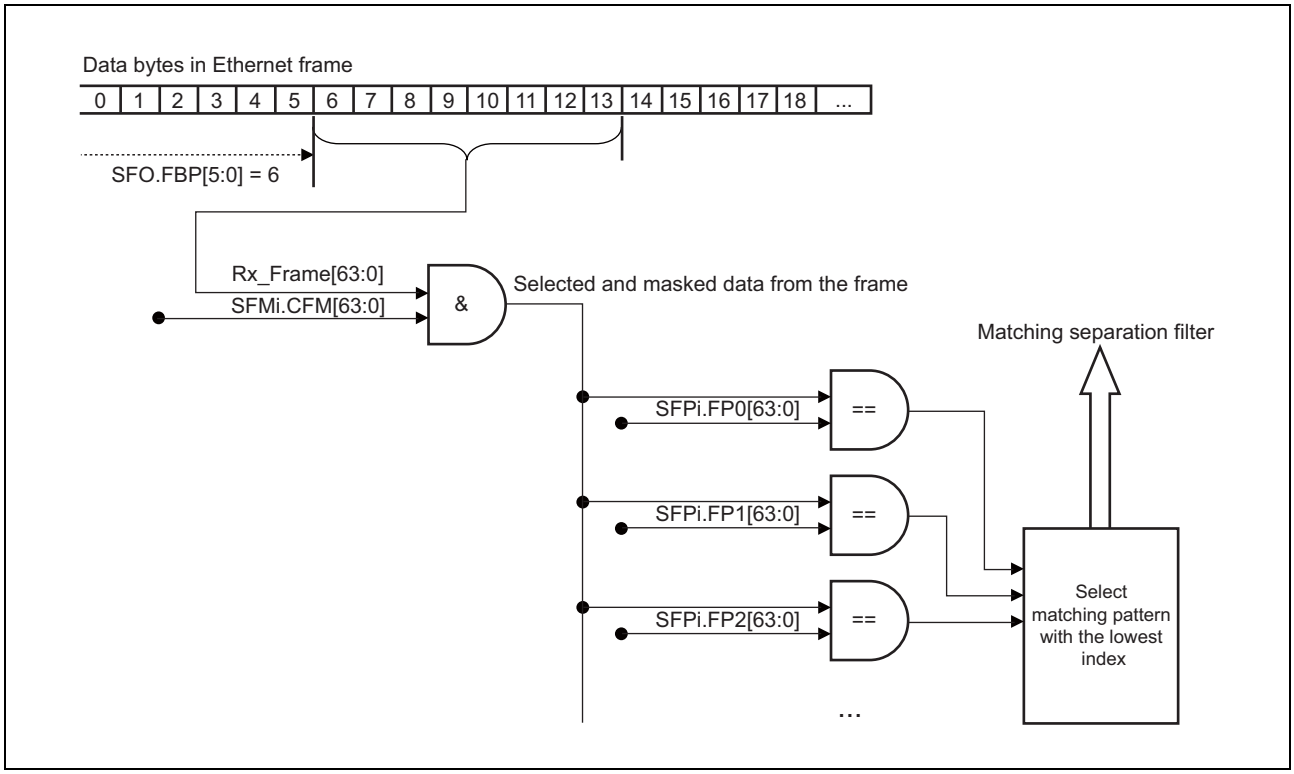


Figure 24.22 Separation Filtering

(2) Separating Streams

The AVB-DMAC applies separation filtering to sort frames received in streams. An AVB network has a concept of “Talker” and “Listener”. A Talker is an end station that generates one or more streams. A Listener is an end station that has the role of being a sink for at least one stream. The various A/V streams are identified by 8-byte stream IDs.

The number of end stations within an AVB network and their roles differ with the application.

The stream ID is a general pattern of the AVB network for identifying one stream. **Figure 24.23** shows the bit allocation of bits in IEEE1722 Ethernet frames and stream ID fields.

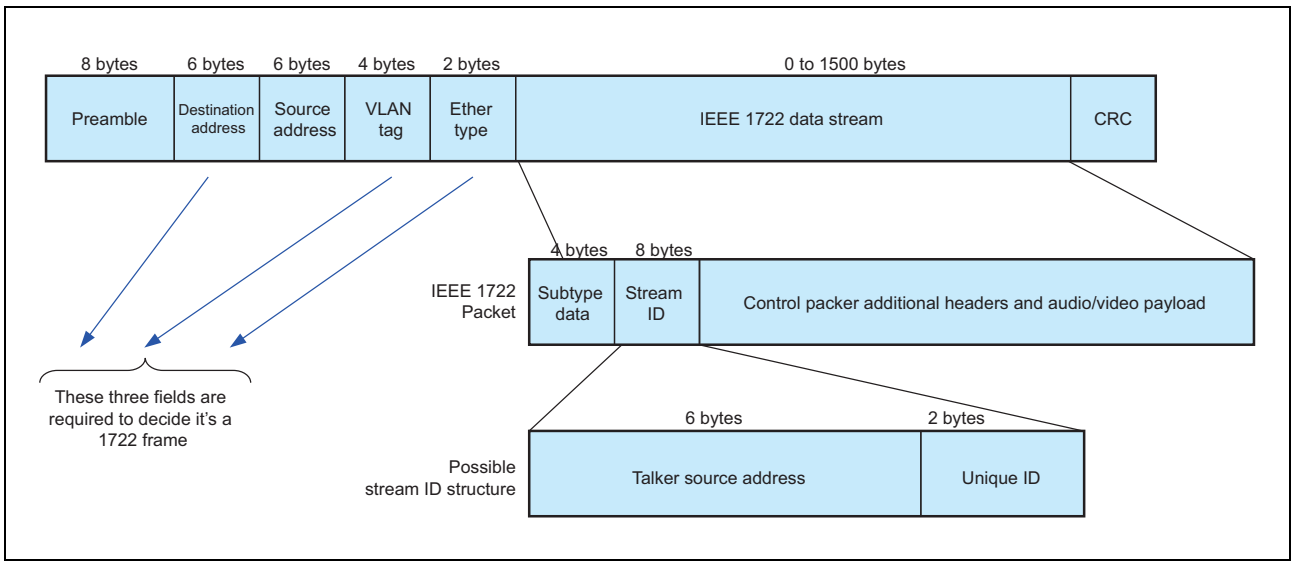


Figure 24.23 IEEE 1722 Frame Layout and Stream ID

The IEEE 1722 standard stipulates that the stream ID field starts from the 23rd byte (not counting the preamble). Accordingly, set the separation filter offset (SFO.FBP[5:0]) to 22 in operations on IEEE 1722 streams. Set the separation filter mask (SFMi) and separation filter pattern (SFPi) in accord with the specification of the product in which the chip is being used.

Example: In the example of a stream ID shown in **Figure 24.23**, the current application divides the field into the talker source address and the unique stream ID. The unique ID is used to differentiate between multiple streams from the same talker. Based on this, there are two settings for separation filter masking:

- To divide various streams into individual queues, set SFM0.CFM[31:0] to H’FFFF FFFF and SFM1.CFM[31:0] to H’FFFF FFFF.
- To divide streams from various talkers into individual queues, set SFM0.CFM[31:0] to H’FFFF FFFF and SFM1.CFM[31:0] to H’0000 FFFF. This excludes the unique ID from the filter condition.

24.4.4.2 Setting Up Reception Descriptors

For reception, the descriptor mechanism is essentially as described in Section 24.4.3, Descriptors.

This section describes memory operations that are especially required in handling reception queues.

(1) Reception Descriptor Type

The type of a descriptor is defined by the descriptor type (DESCR.DT) field.

Table 24.82 shows the descriptor types used in reception.

Table 24.82 Descriptor Types in Reception

Descriptor Type (DESCR.DT)	Operation	Write-back
Frame Start (FSTART)	Condition for data not being stored in a reception queue: The RIS2.QFFr bit indicates that queue r is full and the received frame is not stored. Descriptor processing proceeds again in response to further reception.	Not changed
Frame Middle (FMID)	Same as FSTART	Not changed
Frame End (FEND)	Same as FSTART	Not changed
Frame Single (FSINGLE)	Same as FSTART	Not changed
Link (LINK)	Processing proceeds to the descriptor specified by DESCR.DPTR.	LEMPTY
Fixed Link (LINKFIX)	Same as LINK	Not changed
End Of Set (EOS)	A stop point defined by software has been reached. A frame of this type within a divided frame (writing of FMID or FEND) stops the frame being stored and the frame is lost. RIS2.QFFr indicates that the frame has been lost. If this happens at the start of a frame (writing of FSTART or FSINGLE), storing of frames starts from the next descriptor. In either case, processing shifts to the next descriptor in the chain.	EEMPTY
Frame Empty (FEMPTY)	The descriptor can be used to store received data. Up to DESCR.DS bytes are stored in the descriptor data area. For details, see **Section 24.4.4.3** (1) Storing Frame Data in the Descriptor Data Area.	FSTART, FMID, FEND, or FSINGLE
Frame Empty Incremental Start (FEMPTY_IS)	The descriptor can be used to store received data. All data for the frame are stored in the descriptor's data area. DESCR.DPTR indicates the base address of the incremental data area. For details, see **Section 24.4.4.3** (2) Incremental Data Areas.	FEND or FSINGLE
Frame Empty Incremental Continue (FEMPTY_IC)	The descriptor can be used to store received data. The remaining bytes of frame data are stored in the descriptor's data area. DESCR.DPTR is undefined, but is written back at the start position within the incremental data area after processing. For details, see **Section 24.4.4.3** (2) Incremental Data Areas.	FEND or FSINGLE
Frame Empty No Data storage (FEMPTY_ND)	The descriptor can be used to store received data. Up to DESCR.DS bytes are captured from the reception FIFO but not stored. After processing, DESCR.DS is written back as 0. For details, see **Section 24.4.4.3** (2) Incremental Data Areas	FSTART, FMID, FEND or FSINGLE
Link Empty (LEMPTY)	Same as FSTART	Not changed
EOS Empty (EEMPTY)	Same as FSTART	Not changed

(2) Configuration of Reception Frame Data Descriptors

Figure 24.24shows the configuration of descriptors for use with reception queues. The reception-specific fields are the same whether the descriptor is normal or extended. The reception-specific fields (DESCR.MSC, DESCR.PS, DESCR.EI, and DESCR.TR) are described in **Table 24.83**.

For the other fields and the descriptor types, see Section 24.4.3.6, Descriptor Type.

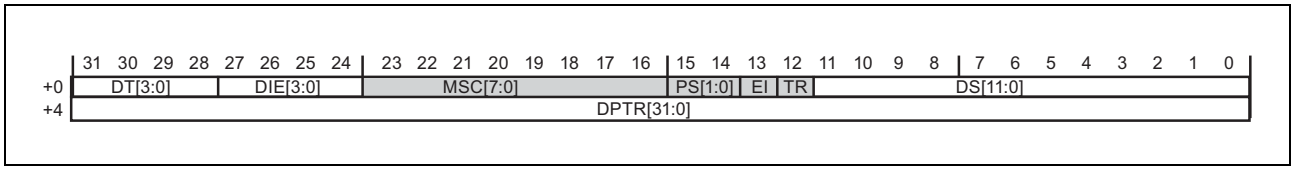


Figure 24.24 Configuration of Descriptor for a Received Frame

Table 24.83 Configuration of a Received Descriptor

Bit Name	Function
MSC	MAC Status Code These bits indicate errors in reception detected by the E-MAC. In the case of a divided frame, these bits are set to the same value within all descriptors for the frame data. Details of the bits are as follows. MSC[7]: Received frame has a multicast address. MSC[6]: Carrier expansion error MSC[5]: Carrier expansion loss MSC[4]: Received frame has residual bits. MSC[3]: Received frame is too long. MSC[2]: Received frame is too short MSC[1]: Error in frame reception MSC[0]: Received frame has a CRC error.
PS	Padding Selection These bits specify whether frame data are to be padded when stored in the incremental data area. Insertion of padding data is in accord with the settings in the RPC register. B'00: Padding is not to be inserted. B'01: Padding data may be inserted. This depends on the RPC settings. Other settings are not effective.
EI	Error Indication This bit indicates the detection of an error in frame data while a frame was being stored. The bit is set to 1 for a descriptor in which an error has been detected. If the descriptor is for a divided frame, storage of the frame is aborted. "0": No error "1": Error is detected
TR	Truncation Indication This bit indicates whether frame data received from the E-MAC have been truncated before being stored. These bits are set to the same value within all frame data descriptors for a divided frame. 0b: Data have not been truncated. 1b: Data have been truncated.

CAUTION

The RCR.EFFS bit specifies whether or not frames with errors detected by the E-MAC are to be stored in the URAM. When the storing of error frames is disabled, error codes are not written to DESCR.MSC.

24.4.4.3 Reception Processing

After initialization, the AVB-DMAC is able to select the proper reception queue and store received frames in the data area in the URAM as indicated by the descriptor. The AVB-DMAC continues to store received data in the URAM as long as space is available for descriptors and data areas.

Received frames are classified and stored in the reception FIFO in accord with the algorithm described in **Section 24.4.4.1** (1) Separation Filtering. A frame will already have been sorted by separation filtering, truncated, or discarded based on reception by the MAC before being stored in the reception FIFO. The following data are stored in the reception FIFO.

- MAC status of received frames
- Length of received frames
- Time stamp of received frames
- Target reception queue
- Received frame data

If the reception FIFO contains even one frame, the scheduler executes storing in the reception queue (see Section 24.4.2.2, Scheduling Reception and Transmission).

If there is even one empty data descriptor in a queue for which reception has started, the storage of frame data starts. Received frames for a queue that is already full (there is no empty frame descriptor or the UFC stop level has been reached) are discarded from the reception FIFO. This ensures that one queue being full does not prevent the storage of data in the other queues.

(1) Storing Frame Data in the Descriptor Data Area

Frame data for storage are assumed to be in either of the two patterns described below.

- The data for an entire frame will fit in the descriptor data area.
  - In this case, the descriptor type (DESCR.DT) is FSINGLE.
- Frame data to be stored in the descriptor data area arrive in divided form.
  - In this case, FSTART is written to the descriptor type (DESCR.DT) bits of the first of the frame data to arrive and FMID and FEND are written to the type bits of descriptors for subsequent data.

The descriptor type is updated by the AVB-DMAC in the last step of descriptor processing, so software can always access the descriptor assigned to DESCR.DT.

When normal synchronization mode is used, the CPU can write FEMPTYxxx directly to the descriptor type field after processing the stored element. Do not change the descriptor or any part of the descriptor data area after FEMPTYxxx is written to DESCR.DT.

(a) Storing Frame Data for a Whole Single Frame

For a frame with an FSINGLE descriptor, all data for the frame are held at the position defined by DESCR.DPTR. DESCR.DS indicates the length of the received frame.

If DESCR.DS is bigger than the actual size of a received frame, the FEMPTY or FEMPTY_ND descriptor is stored in place of the FSINGLE descriptor after processing.

Also, the FEMPTY_IS and FEMPTY_IC descriptors, which always hold the full frame data for the reception FIFO, are stored in place of the FSINGLE descriptor after processing.

(b) Storing Frame Data as Divided Frames

Divided frames are handled in the same way as a single frame. A frame stored with divided descriptors must be recombined before use. DESCR.EI and DESCR.TS are only valid in the last descriptor of the sequence for a divided frame.

CAUTION

If the data area size setting in DESCR.DS is not a multiple of four, the number of bytes set in DESCR.DS is fetched from the reception FIFO and the remaining bytes are used as the next storage area.

After a received frame is divided into different descriptors, each storage element is handled separately, and the descriptor type is assigned by software after processing. Accordingly, an error frame (FEMPTYxxx instead of FMID or FEND) may exist while a descriptor chain is being processed. In such a case, the CPU must postpone processing of the error frame to the next trigger point.

(c) No Data are Stored

The application specification may lead to some types of received frames being unimportant (for example, when the application only requires stream data from the Ethernet frames). Storing frames in divided form makes separating out the unnecessary parts of Ethernet frames possible.

If part of a divided frame is not required, use the FEMPTY_ND descriptor for that part so that it is not stored in the URAM. Not storing the data negates the need for bandwidth on the data bus, improving the overall performance.

When an FEMPTY_ND descriptor is processed, DESCR.DS is set to 0. This brings the frame data section of the descriptor into agreement with the FEMPTY type. DESCR.DS = 0 is for the unique identification of the descriptor after writing.

(2) Incremental Data Areas

Secure space in the URAM for storing received data. Even when data are placed in the URAM area such that all descriptor data areas of a chain are contiguous, a received frame being shorter than the descriptor data area will lead to an empty space. **Figure 24.25** shows an example of settings and the memory map.

Certain applications require that data areas be contiguous (e.g. when received data are to be processed other than by hardware as the A/V codec module). When the length of received frames differs (e.g. when payloads vary between having one or two A/V packages), the use of a static pointer in the descriptor produces empty spaces in the data area. This may necessitate direct additional processing to remove the empty spaces.

Accordingly, and to reduce the CPU load imposed by copying data, the AVB-DMAC supports an "incremental data area" function.

When incremental data areas are in use, all descriptors use a common data area for storage. One descriptor (FEMPTY_IS) defines the base address of the incremental data area and the next descriptor (FEMPTY_IC) within the descriptor chain holds received data. **Figure 24.26** shows an example of settings and the memory map.

Use of an incremental data area does not reduce the memory space in the individual descriptor data areas.

The hardware and software synchronization strategy and performance are also not changed.

It is also possible to divide a frame up among various descriptors in a way that reflects its structure (e.g. one descriptor for the Ethernet header and one for the data payload).

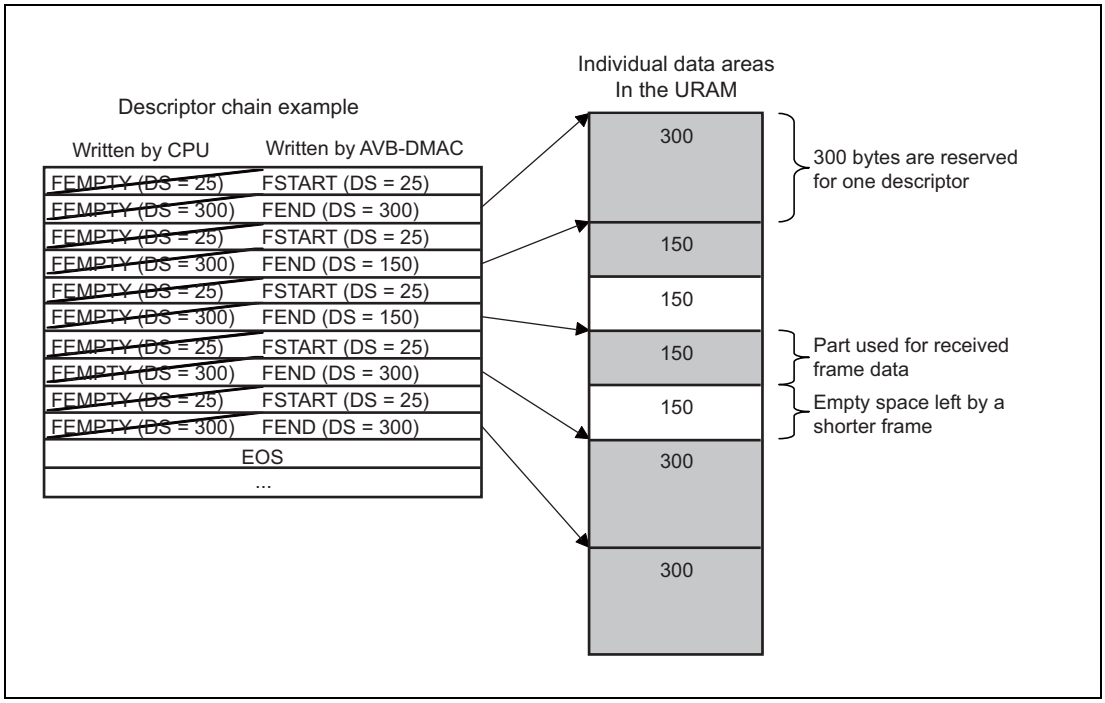


Figure 24.25 A Reception Queue Using Individual Descriptor Data Areas

Figure 24.25 and **Figure 24.26** show how control of the data storage areas by a descriptor chain varies according to whether individual or incremental data areas are in use. The chains are configured for storing received frames consisting of a 25-byte header (which is treated as one descriptor; and is outside the scope) and a 150- or 300-byte payload (whether one or two 150-byte payload packages are transmitted with one Ethernet frame depends on the data source).

In **Figure 24.25**, the EOS descriptor is added as an example of a re-synchronization point. If the frame source transmits a frame containing more than 325 bytes, the frame will be divided among three descriptors, meaning that synchronization of the header and data sequences is lost. Despite this, however, the frame is not divided across the EOS descriptor, so recovery is automatic without software interaction. The EOS is not required with the incremental descriptors because all data being processed are always stored while an incremental data area is in use.

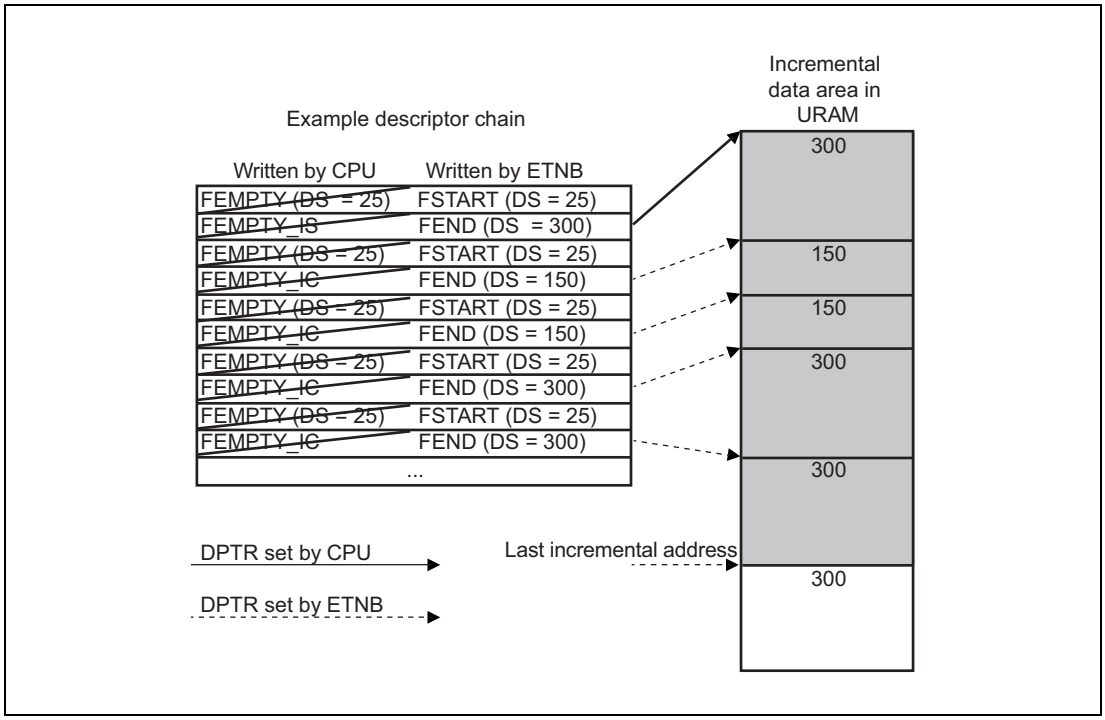


Figure 24.26 Reception Queue Using a Common Incremental Data Area

As **Figure 24.26** shows, when data are stored in an incremental data area, the descriptor pointers in the FEMPTY_IC descriptors (DESCR.DPTR) are updated. Accordingly, the resulting FEND or FSINGLE descriptor is in the same format as after writing to an FEMPTY descriptor.

Software captures received data from an incremental data area, which has no empty storage areas between frame data. The only empty space is that at the end of the incremental data area. The sizes of incremental data areas and of blocks of data for storage in incremental data areas must be multiples of four bytes. When the amount of data for storage in an incremental data area is not a multiple of four bytes, from one to three bytes of empty space will be produced. DESCR.DS can be read to check for such empty spaces.

Directly controlling the amount of received data to be stored from the incremental descriptors FEMPTY_IS and FEMPTY_IC is not possible because storage due to the DESCR.DS of other descriptors (FEMPTY and FEMPTY_ND) is also possible. All received data in the chain are always stored in an incremental descriptor.

(a) Setting Up an Incremental Data Area

A descriptor chain in the incremental data area having N descriptors (one FEMPTY_IS and N-1 FEMPTY_IC) means that a storage area for the maximum of N times the capacity must be prepared.

As **Figure 24.26** shows, DESCR.DPTR of an FEMPTY_IS descriptor indicates the base address of the incremental data area. The next FEMPTY_IS descriptor in the chain indicates the processing step where data must be stored in the incremental data area.

(b) Processing an Incremental Data Area Based on Descriptors

Since data processing by the CPU is the same regardless of how the AVB-DMAC stores the data, data stored in an incremental data area do not require any special handling.

(c) Padding

Use padding for received frame data that are not aligned correctly in the specified memory structure. Padding can be set individually for each descriptor. Accordingly, in the reception of divided frames, padding can be restricted to only those frames that require it (e.g. A/V payload data.)

Padding can also be used to optimize system performance in an incremental data area (e.g. to prevent inefficient access by aligning received data with 32-byte boundaries in the incremental data area), as well as to fulfill application-specific requirements for specified memory structures (e.g. formats required by other modules that will be processing the received data).

Padding can only be used in an incremental data area.

The value H'0000 0000 is always used in padding.

Padding is the addition of the number of words (from one to seven 32-bit words) set in the stored padding counter in the receive padding configuration register (RPC.PCNT[2:0]). This padding is repeatedly inserted in accord with the value in the stored data counter (RPC.DCNT[7:0]) (from one to 255 32-bit words). When the stored data counter (RPC.DCNT[7:0]) reaches 0, however, padding is not repeated.

The first word of padding is always inserted at the position specified by DESCR.DPTR. When divided frames are in use, a padding word can be inserted at any byte position, and padding is handled on a 32-bit basis (e.g. an incremental data area where the first descriptor is for a 42-byte header data and the second descriptor holds padded payload data).

The next figure shows a general example of how padding is inserted and an example of setting up padding. A indicates frame data A received from the E-MAC, while B indicates frame data already stored in the descriptor data area (32-bit word units).

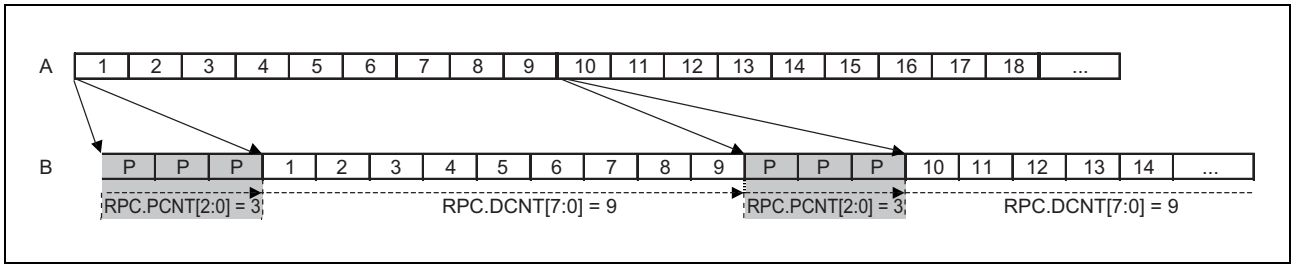


Figure 24.27 Example of a Padding Setting

Both padding and received frame data are counted in the descriptor size (DESCR.DS).

(3) Mode with Write-Back

Constructing a descriptor chain requires software (see **Figure 24.28**).

In the example in the figure, the variable SWdescr (software descriptor pointer) is a structure to identify a descriptor being processed. SWdescr must be initialized after operation mode is entered and a descriptor base address load request (DLR.LBAq) is executed (condition for starting the flow of software operations).

The frame_processing() function processes the stored data. The function can use SWdescr.DT to check whether processing of a frame is completed. How frame data are processed differs with the application, so create functions that handle processing in accord with the specification.

The processing section is common to all modes of reception. The number of frames processed in response to each trigger can be restricted. When multiple frames have to be processed in a batch, waiting for individual trigger boxes must be skipped for these frames.

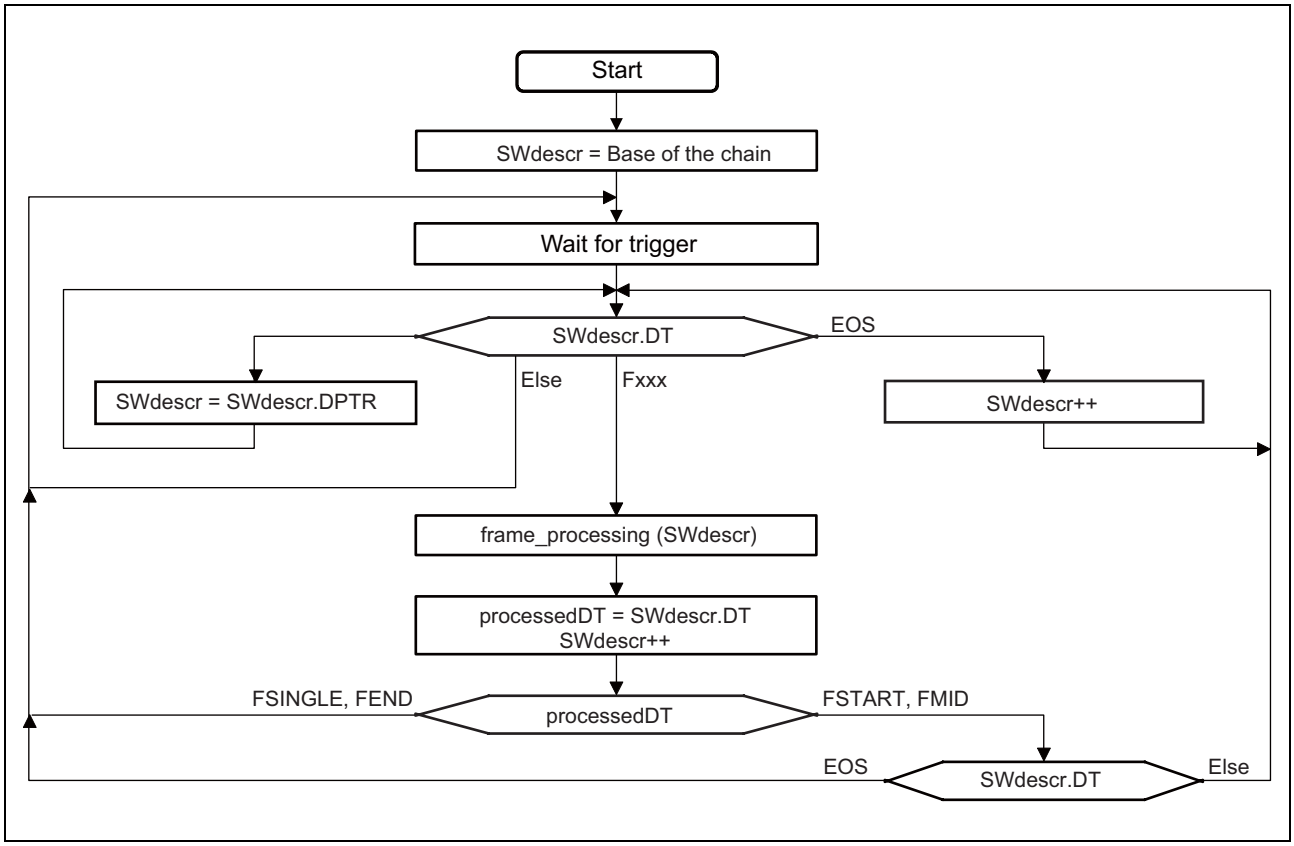


Figure 24.28 Flow of Reception Descriptor Processing (with Write-Back)

(4) Support for Reception Time Stamps

Capturing reception time stamps is essential for IEEE 802.1AS time synchronization. Other types of received frames may also require that a reception time stamp be appended; this depends on the application. The AVB-DMAC supports reception time stamps based on the gPTP timer by storing time stamps, which have been captured when Start Frame Delimiter (SFD) of a received frame is arrived, in the last frame data descriptor (FEND or FSINGLE). For the gPTP timer, see Section 24.4.7.1, gPTP Timer.

When time stamps are to be stored, use extended descriptors for the entire reception queue. Furthermore, time stamps are always stored for reception queue 1 (network control). Time stamps for reception queue 0 (best effort) and reception queue r ($r \geq 2$; for stream data) can be selected by the time stamp enable bits in the receive configuration register (RCR.ETS0 or RCR.ETS2).

24.4.4.4 Unread Frame Counters

Each reception queue has an unread frame counter (UFCVi). Use the unread frame counter configuration bits in the receive queue configuration register (RQCi.UFCCr) to select from among the four warning and stop levels for each unread frame counter. The 0 setting disables the stop and warning functions. For how to set this up, see **Figure 24.29**.

Operations of the AVB-DMAC (hardware) and CPU (software) drive an unread frame counter (UFC) in the following ways.

- The hardware indicates that it has added a new frame to the descriptor chain for the queue (this increments the counter).
- Software indicates how many frames from the descriptor chain it has processed by writing to the corresponding bits of the unread frame counter decrement register for the queue (this decrements the register by the number written).

The unread frame counter is based on the number of frames stored in the URAM and is only incremented by one even when a received frame is divided into different descriptors. Failure in storing a descriptor chain requires care because this may unread frame counter may fail in synchronization as described in **Section 24.4.4.4** (1) Unread Frame (UFC) Synchronization Failure.

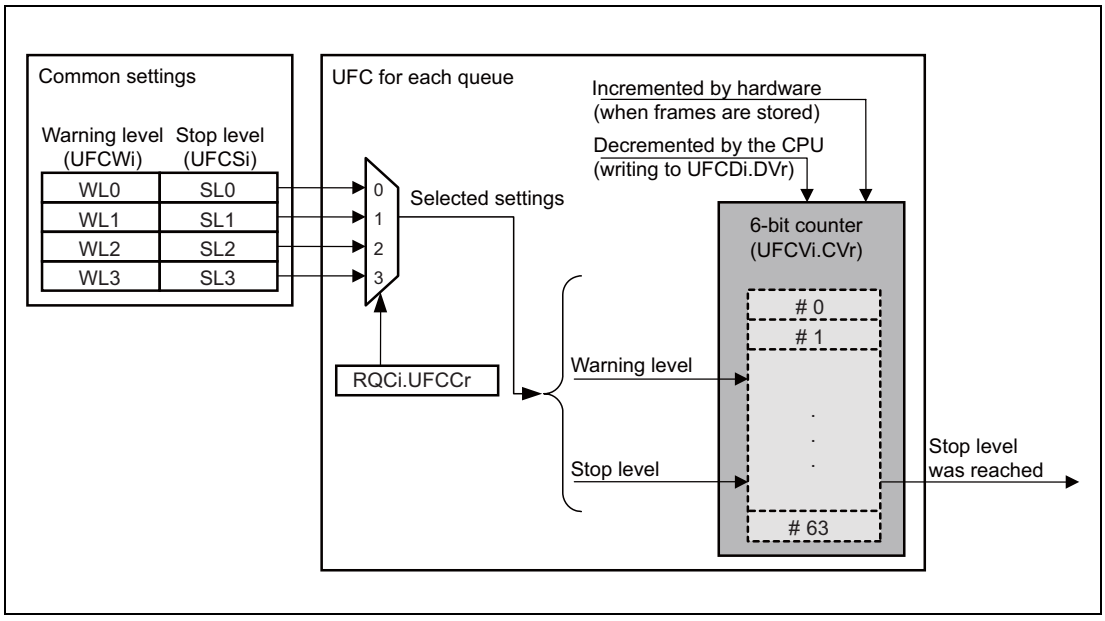


Figure 24.29 Overview of an Unread Frame Counter

Unless synchronization of hardware and software is lost, the current unread frame counter value (UFCVi.CVr) indicates the number of unread frames in the queue.

The indicator that the stop level has been reached prevents the storage of further received frames in the descriptor chain. Selecting 0 as the stop level disables this function. Otherwise, further received frames for the queue are discarded once its unread frame counter reaches the stop level. Activation of the unread frame counter stop function is indicating by setting of the receive queue full interrupt flag in the receive interrupt status register 2 (RIS2.QFFr).

Set the unread frame counter warning level configuration register (UFCW) and the unread frame counter stop level configuration register (UFCS) for each reception queue that will use the unread frame counter function while the current operating mode is configuration mode.

(1) Unread Frame (UFC) Synchronization Failure

The unread frame counters do not recognize failure to store a frame in the URAM. In other words, the AVB-DMAC increments the counter for a queue each time it captures a frame for that queue from the reception FIFO whether or not it succeeds in storing the frame normally in the descriptor chain.

In general, synchronization of hardware and software fails under the following conditions.

- An unread frame counter reaching its maximum value
  When the value of a counter in an unread frame counter register i (UFCVi) (i = 0 to 4) reaches 63, synchronization for the corresponding queue can fail.
  The CPU can only judge that a failure in synchronization has not occurred when the stop level is set to 63.
- A queue not having enough space for a descriptor or the associated data
  In this case, the corresponding receive queue full interrupt flag (RIS2.QFFr) in receive interrupt status register 2 (RIS2) is set.
  If an unread frame counter reaches its stop level while synchronization remains normal, the receive queue full interrupt flag (RIS2.QFFr) in the receive interrupt status register 2 (RIS2) is set. Software must respond to this.
- A problem occurring during access to memory

The result of a failure in synchronization is the unread frame counter indicating that the corresponding descriptor chain contains more available frames than it actually does. To retrieve the correct starting point for operations, use the descriptor base address load request (DLR.LBAq) for the given queue.

24.4.5 Transmission Control

Areas in the URAM for storing transmission descriptors must also be secured (for descriptors, see Section 24.4.3, Descriptors).

The AVB-DMAC fetches data from the URAM in accord with the procedure the descriptor describes. The descriptor also retains tag information once the frame has been fetched for transmission. The tag information is used to maintain the relationships between state information and time stamps for the software and the AVB-DMAC. The status and time stamp information for transmitted frames remains accessible after their transmission is completed.

24.4.5.1 Transmission Modes

The AVB-DMAC has two modes of transmission.

- AVB transmission mode
  This mode is selected by the priority level setting for the transmission queue in the transmit configuration register (setting of the TGC.TQP[1:0] bits) being B'01 or B'11.
- Non-AVB transmission mode
  This mode is selected by the priority level setting for the transmission queue in the transmit configuration register (setting of the TGC.TQP[1:0] bits) being B'00.

(1) AVB Transmission Mode

AVB transmission supports the control of traffic through the output port to implement various traffic classes.

(a) Support for Traffic Classes and Associated Priority

When transmission is in AVB transmission mode, streams of traffic are transmitted in accord with the part of the AVB specification called Forwarding and Queuing for Time Sensitive Streams (FQTSS; for details on this, see the IEEE 802.1Q standard).

In the AVB specification, at least one queue for a reserving stream under the Stream Reservation Protocol (SR stream) and at least one queue for a non-SR stream are present, and the queues for SR traffic are highest priority queues.

The AVB-DMAC supports four traffic classes: SR class A, SR class B, network control (NC) traffic (gPTP frames), and best effort (BE) traffic. Allocating a specific queue to network control (NC) frames ensures the control of synchronization.

The AVB-DMAC realizes compliance with the AVB standards by handling queues with the following architecture (in terms of traffic classes).

- Four transmission queues (Q3, Q2, Q1, and Q0) are available.
- Q3 and Q2 are for SR streams (one each for class A and class B).
- Q1 is for low-bandwidth network control (NC) traffic (gPTP frames)
- Q0 is for other types of traffic (MSRPDU*1, MVRPDU*2, best effort (BE), etc.)

CAUTIONS

1. **MSRPDU: Multiple Stream Registration Protocol Data Unit**
2. **MVRPDU: Multiple VLAN Registration Protocol Data Unit**

Fetching from queues proceeds in order of priority of the above traffic types. Three systems of priority are available through the setting of the transmit queue priority bits in the transmit configuration register (TGC.TQP[1:0]). In the default priority scheme, which is called AVB mode 1 (selected by TGC.TQP[1:0] = B'11), operation of the AVB-DMAC is fully in accord with the AVB specification. AVB mode 2 (transmit queue priority bits (TGC.TQP[1:0] = B'11) is an alternative priority scheme and varies from the AVB specification. Using this scheme thus requires more care. The other setting is for non-AVB-mode transmission.

Table 24.84 Default and Alternative Priority Orders in AVB Transmission Mode

Priority Schemes (AVB Mode)	Priority Order of Queues
Default	Q3 (SR class A) > Q2 (SR class B) > Q1 (NC) > Q0 (BE)
Alternative	Q1 (NC) > Q3 (SR class A) > Q2 (SR class B) > Q0 (BE)

(b) Transmission Selecting Algorithm and CBS

The algorithm the AVB-DMAC applies to select frames for transmission is in accord with the specifications under section 8.6.8, Transmission selection, of the IEEE 802.1Q standard. For AVB mode, the CBS (credit-based shaping) algorithm is applied to the class A and class B SR queues (Q3 and Q2). Use of the CBS enables correct handling of the priorities of transmission from the SR queues. For the CBS algorithm, see Section 24.4.6, CBS (Credit-Based Shaping).

When the following conditions both hold, transmission from an SR queue (Q3 or Q2) proceeds at the specified time.

- The queue contains at least one frame ready for transmission.
- The queue has available credit.
- Unless an SR queue satisfies the above conditions, a higher priority queue is not present (not ready for transmission).

A non-SR queue (Q1 or Q0) is selected if the conditions below both hold.

- The queue contains at least one frame ready for transmission.
- As well as the above condition, a higher priority queue is not present (not ready for transmission).

Figure 24.30 and **Figure 24.31** are flowcharts of selection for transmission in AVB mode 1 (default) and AVB mode 2 (alternative).

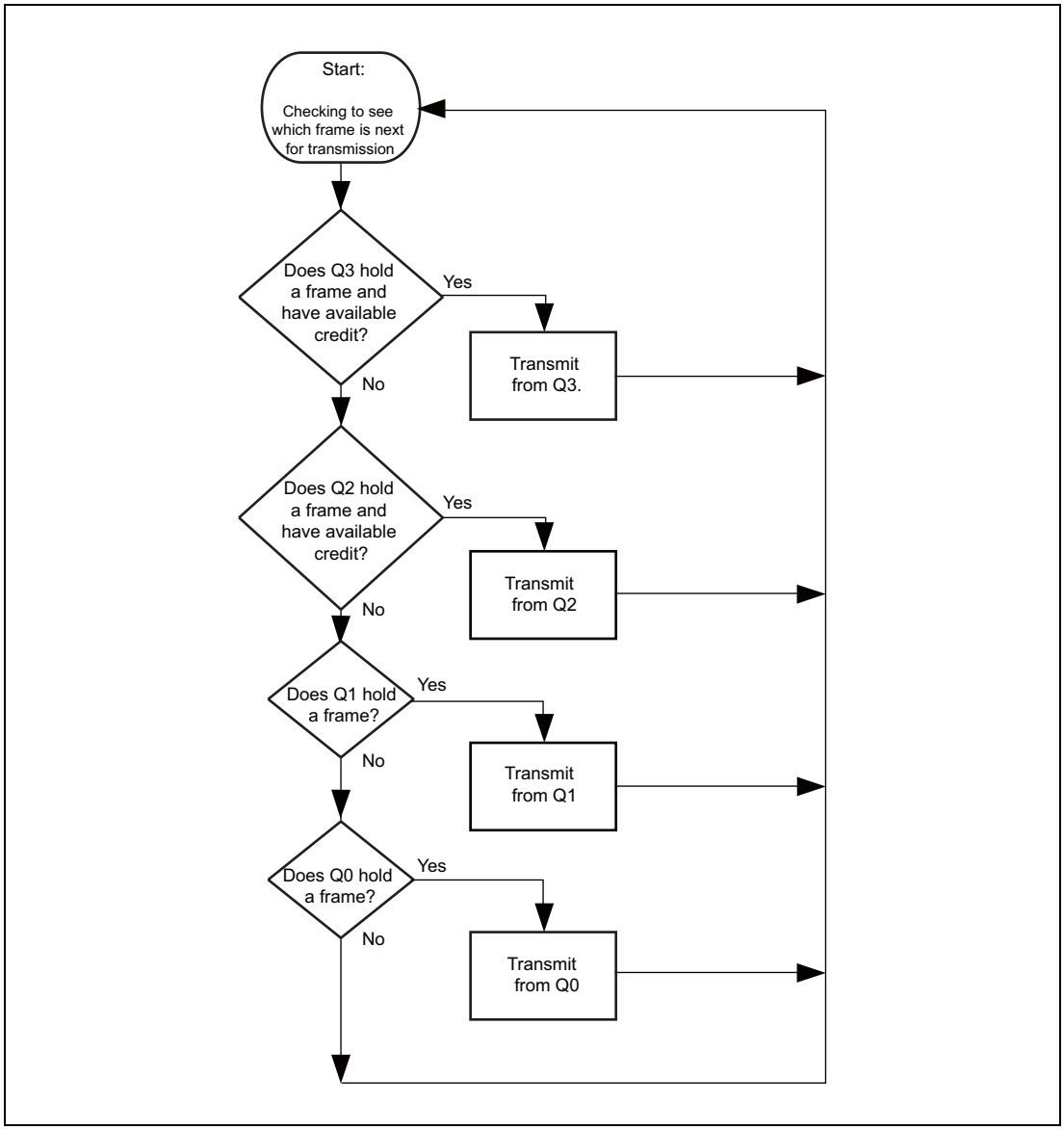


Figure 24.30 Flow of Selection for Transmission in AVB Mode 1 (Default)

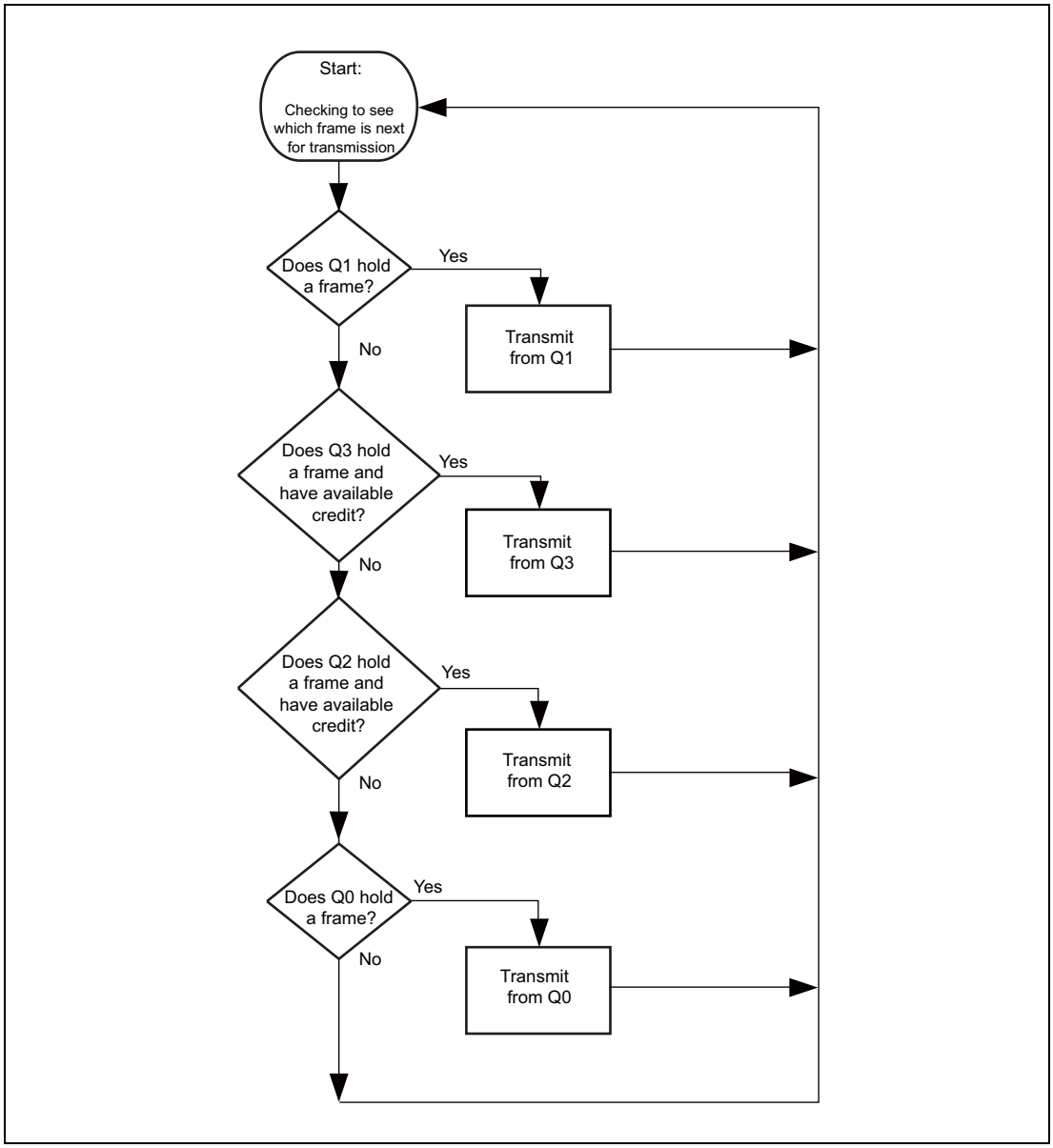


Figure 24.31 Flow of Selection for Transmission in AVB Mode 2 (Alternative)

(2) Non-AVB Transmission Mode

In non-AVB transmission mode, an absolute priority scheme is used. The SR class is not supported and the CBS algorithm is not used.

In non-AVB transmission mode (when the transmit queue priority bits in the transmit configuration register (TGC.TQP[1:0]) are B'00), data is fetched for transmission in a strict order of priority (Q3 > Q2 > Q1 > Q0).

Figure 24.32 shows the flow of selection in non-AVB transmission mode.

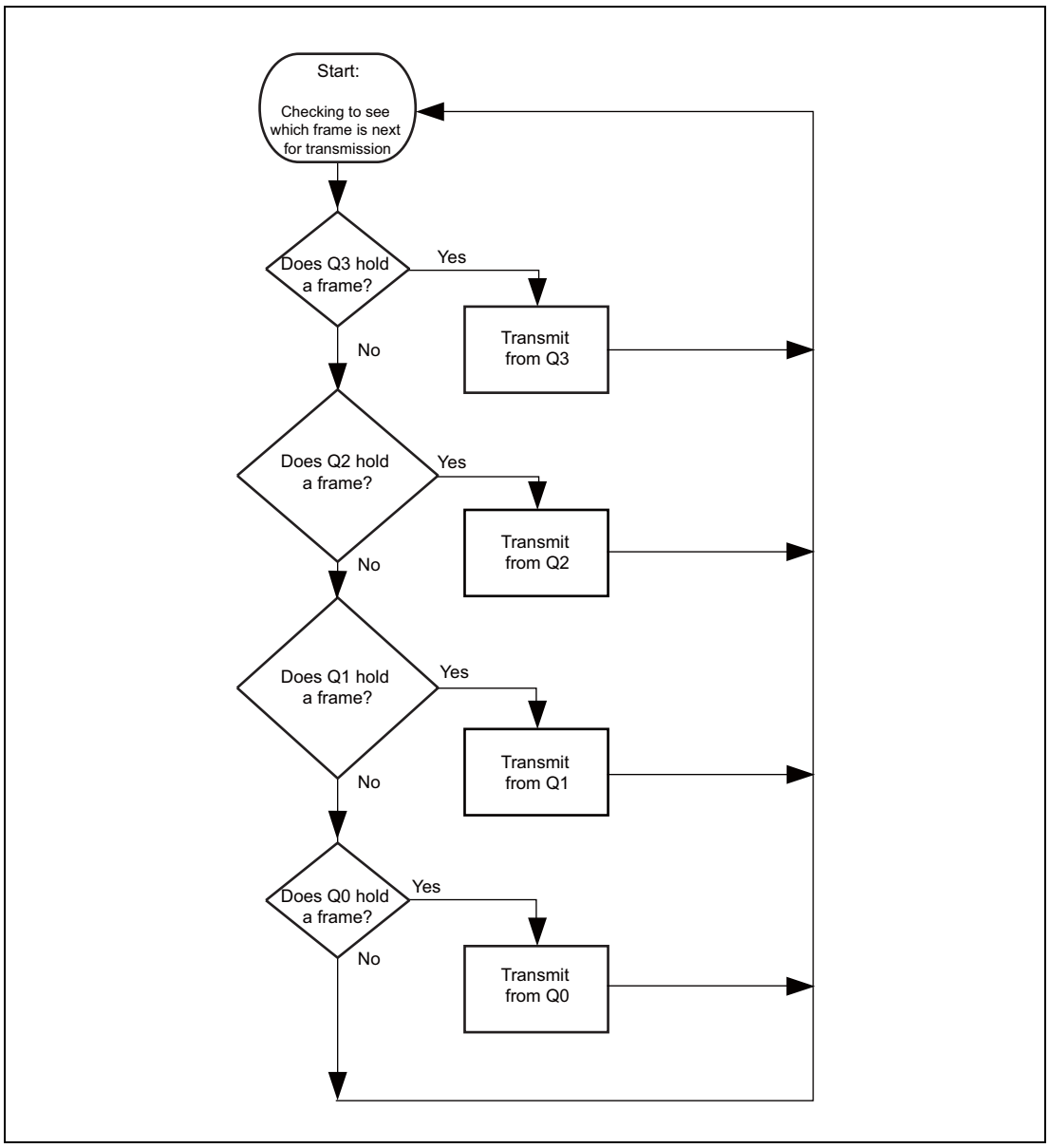


Figure 24.32 Flow of Selection for Transmission in Non-AVB Mode

(3) Setting the Size of the Transmission FIFO

The transmission FIFO is made up of 124 clusters. Each cluster can hold up to 128 bytes.

The size of the part of the transmission FIFO for use by each of the four transmission queues can be set by the corresponding transmit queue configuration q bits in the transmit control register (TGC.TBDq). The setting of TGC.TBDq is fixed to 2.

General Usage Examples:

Q0: Frames containing up to 1500 bytes → 1500/128 = 11.7 → 12 clusters

Q1: Frames containing up to 1024 bytes → 1024/128 = 8.0 → 8 clusters

Q3: Frames containing up to 1996 bytes → 1996/128 = 15.6 → 16 clusters

Q4: Frames containing up to 1996 bytes → 1996/128 = 15.6 → 16 clusters

When the depth of all transmission queues is 2, only the following number of clusters is required.

2 * (12 + 8 + 16 + 16) + 16 = 2 * 52 + 16 = 120

In the worst case (each of the transmission queues holds 1996 bytes), 2 * 64 + 16 = 144 clusters are required, so 1 must be set in TGC.TBDt for at least two transmission queues (e.g. Q0 and Q1).

24.4.5.2 Setting Up Transmission Descriptors

(1) Transmission Descriptor Type

The type of a descriptor is defined by the descriptor type (DESCR.DT) field.

Table 24.85 shows the descriptor types used in transmission.

Table 24.85 Descriptor Types in Transmission

Descriptor Type (DESCR.DT)	Operation	Write-back
Frame Start (FSTART)	The AVB-DMAC fetches the first of the data for the divided frame and proceeds to processing of the next descriptor.	FEMPTY
Frame Middle (FMID)	The AVB-DMAC fetches the second or subsequent data for the divided frame and proceeds to processing of the next descriptor.	FEMPTY
Frame End (FEND)	The AVB-DMAC fetches the last of the data for the divided frame. When the frame of data that has been fetched to the transmission FIFO is ready for transmission by the E-MAC, the AVB-DMAC proceeds to processing of the next descriptor.	FEMPTY
Frame Single (FSINGLE)	The AVB-DMAC fetches the frame of data. When the frame of data that has been fetched to the transmission FIFO is ready for transmission by the E-MAC, the AVB-DMAC proceeds to processing of the next descriptor.	FEMPTY
Link (LINK)	Processing proceeds to the descriptor specified by DESCR.DPTR.	LEMPTY
Fixed Link (LINKFIX)	Same as LINK	Not changed
End Of Set (EOS)	This is a transmission stop point defined by software This leads to clearing of the transmit start request bit (TCCR.TSRQt), which stops transmission. When the TCCR.TSRQt is again set to 1 (a new transmission start request is issued), processing proceeds to the next descriptor.	EEMPTY
Frame Empty (FEMPTY)	No frame data are ready for transmission This leads to clearing of the transmit start request bit (TCCR.TSRQt), which stops transmission. When the TCCR.TSRQt is again set to 1 (a new transmission start request is issued), processing starts at this descriptor.	Not changed
Link Empty (LEMPTY)	Same as FEMPTY	Not changed
EOS Empty (EEMPTY)	Same as FEMPTY	Not changed

(2) Configuration of Transmission Frame Data Descriptors

Figure 24.33 shows the configuration of descriptors for use with transmission queues. The transmission-specific fields (DESCR.TSR, and DESCR.TAG) are described in **Table 24.86**.

For the other fields and the descriptor types, see Section 24.4.3.6, Descriptor Type.

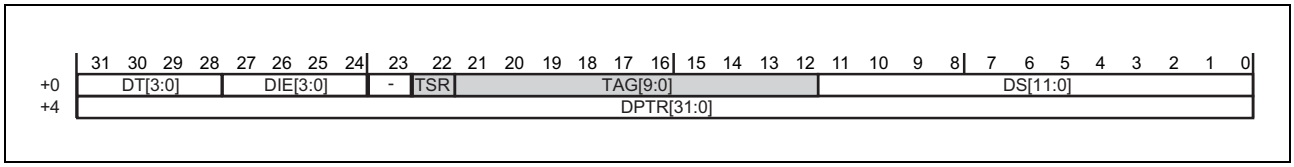


Figure 24.33 Configuration of Descriptor for a Transmitted Frame

Table 24.86 Configuration of a Transmission Descriptor

Bit Name	Function
TSR	Time Stamp Store Request This bit specifies whether the transmission time stamp is to be stored within the EthernetAVB module. 0: The time stamp status FIFO within the EthernetAVB module does not retain a transmission time stamp. 1: The time stamp status FIFO within the EthernetAVB module retains a transmission time stamp. Only control this bit while the current DESCR.DT is FEND or FSINGLE.
TAG	Frame Tag This TAG field is used to associate each frame data with a time stamp. Frame TAG is not required but is recommended. Only control this bit while the current DESCR.DT is FEND or FSINGLE.

For the time stamp FIFO function, see Section 24.4.5.4, Time Stamping in Transmission.

24.4.5.3 Transmission

(1) Transmitting Frames

Setting the transmit start request bit in the transmit configuration control register (TCCR.TSRQt) starts the transfer of frames from the corresponding transmission queue.

The descriptor in the current descriptor address (CDARq.CDA[31:0]) for the queue t (with q = t) is read first.

If this descriptor is a descriptor for frame transmission (FSINGLE, etc.), the AVB-DMAC fetches the frame data from the data area indicated by the descriptor, writes FEMPTY back to the descriptor type (DESCR.DT) bits to indicate completion of this processing, then proceeds to processing of the next descriptor.

If the descriptor is not for transmission, processing is as dictated by the given descriptor (for these descriptors, see the descriptions in Section 24.4.3, Descriptors).

If a base address load request is issued for a descriptor chain while it is being processed (by setting 1 in the LBAq bit for transmission queue q that is currently being processed in the descriptor base address load request register, DLR), processing proceeds to the new descriptor chain. Changing the chain does not interrupt frame fetching, but note that frames that have not been fetched from the old chain remain where they are.

Figure 24.34 shows descriptor processing during transmission.

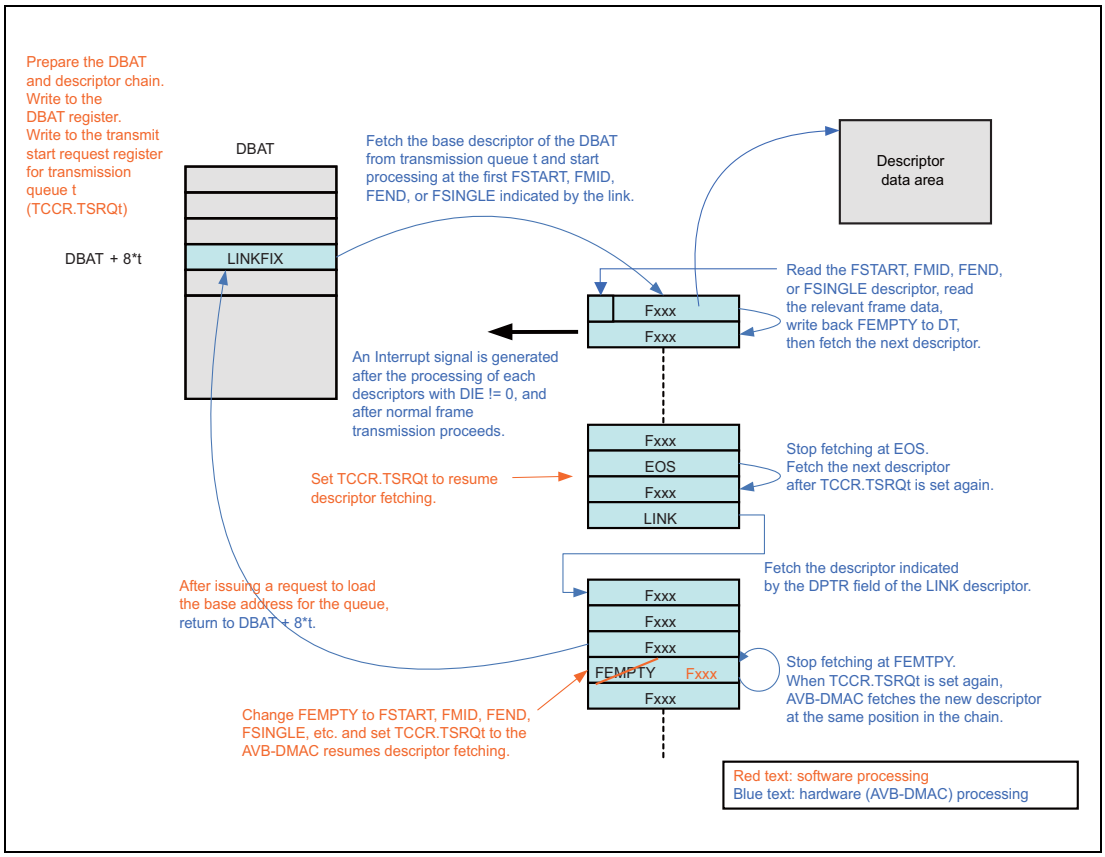


Figure 24.34 Descriptor Processing During Transmission

(2) Examples of Descriptor Usage

(a) Immediate Frame Transmission

Immediate frame transmission is a pattern in which fetching by the AVB-DMAC starts whenever software adds data to a queue. FEMPTY descriptors are used as stop points to keep the hardware and software in synchronization.

Create descriptor chains that have FEMTPY descriptors at the stop points.

Figure 24.35 shows the flow for software implementing this pattern.

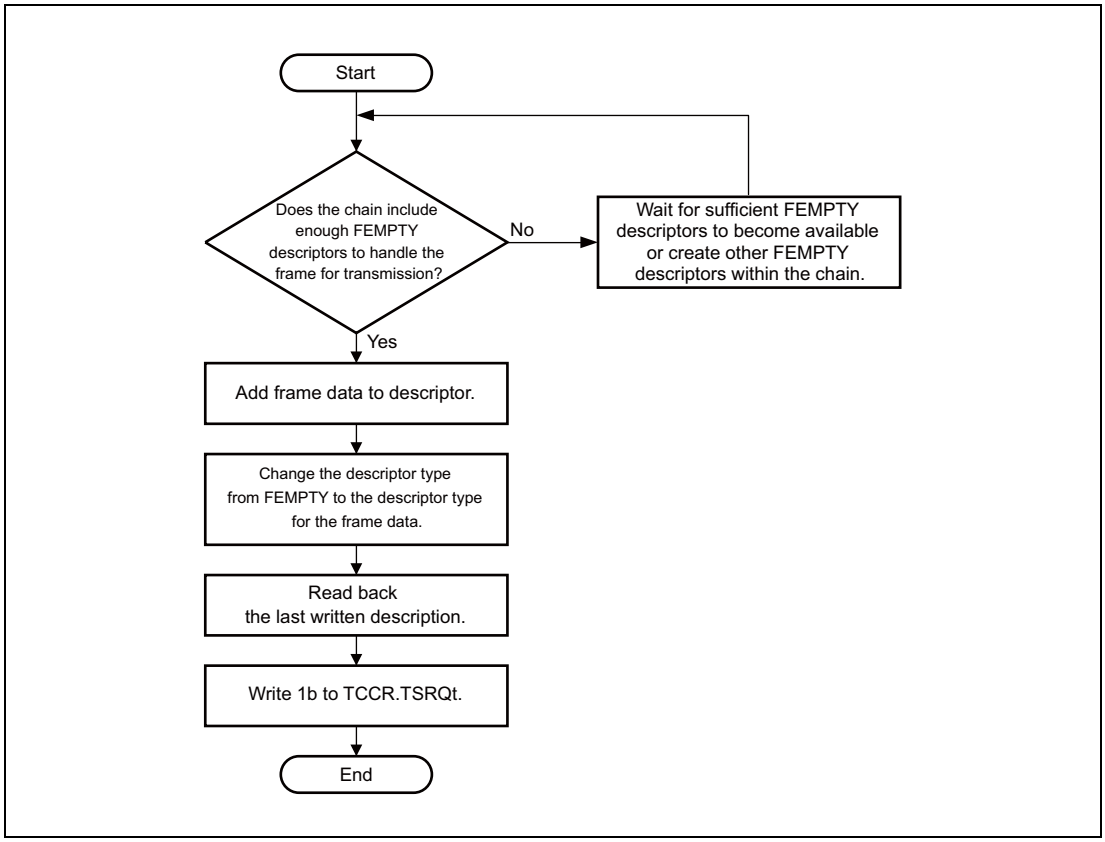


Figure 24.35 Software Flow for Immediate Frame Transmission

Figure 24.36 shows software and AVB-DMAC operations for immediate frame transmission.

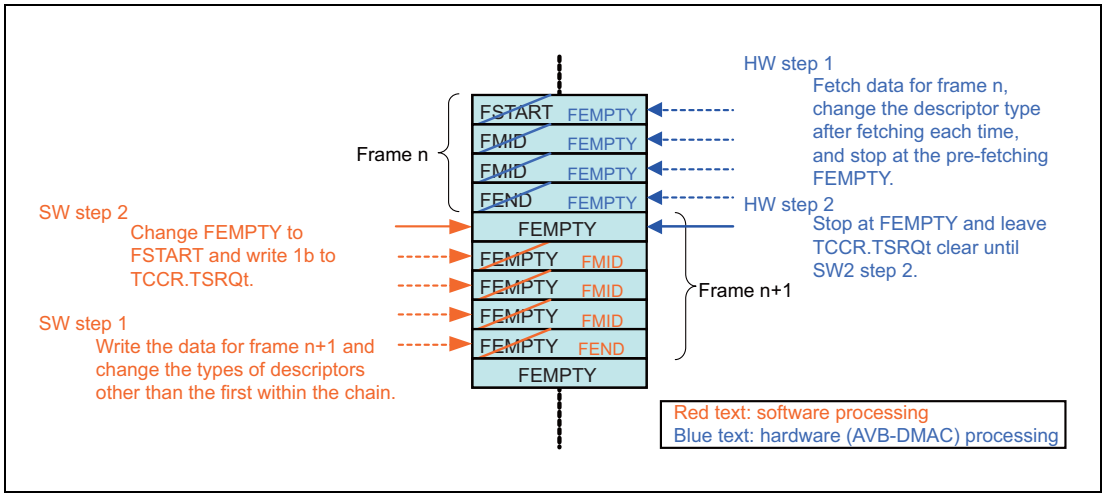


Figure 24.36 Software and AVB-DMAC Operations for Immediate Frame Transmission

(b) Frame Set Transmission with Changing of the Active Descriptor Chain

This pattern is used when data are transmitted with a delay for software control to secure bandwidth or for other reasons, rather than immediately transmitted. EOS descriptors are used for the stop points.

Start by creating a descriptor chain that has a FEMPTY descriptor as its stop point.

Figure 24.37 shows the software flow in this pattern.

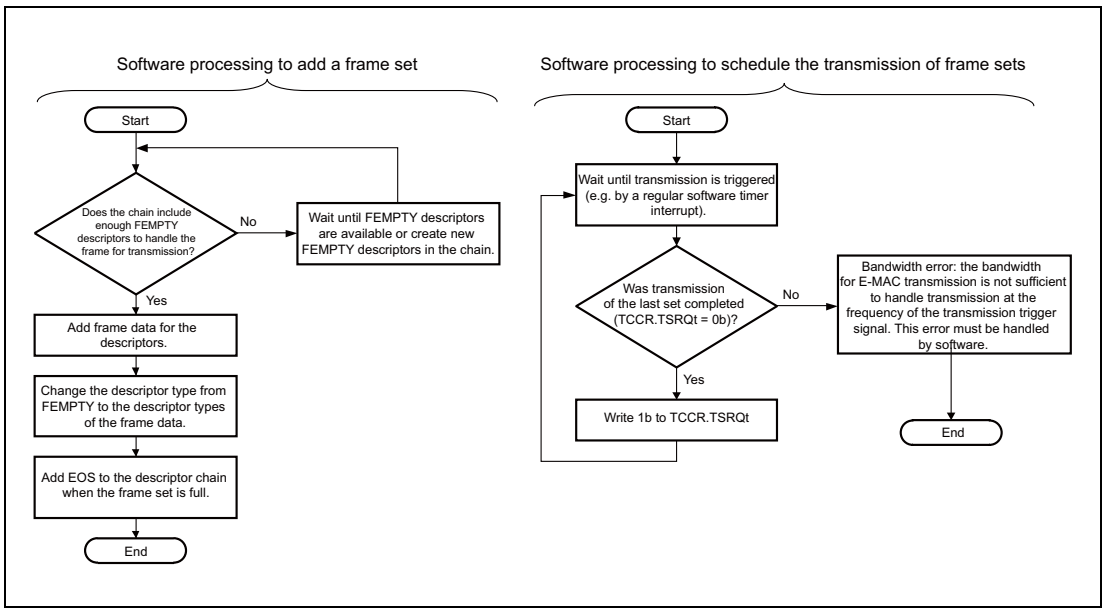


Figure 24.37 Software Flow for Frame Set Transmission with Changing of the Active Descriptor Chain

Figure 24.38 shows software and AVB-DMAC operations for frame set transmission.

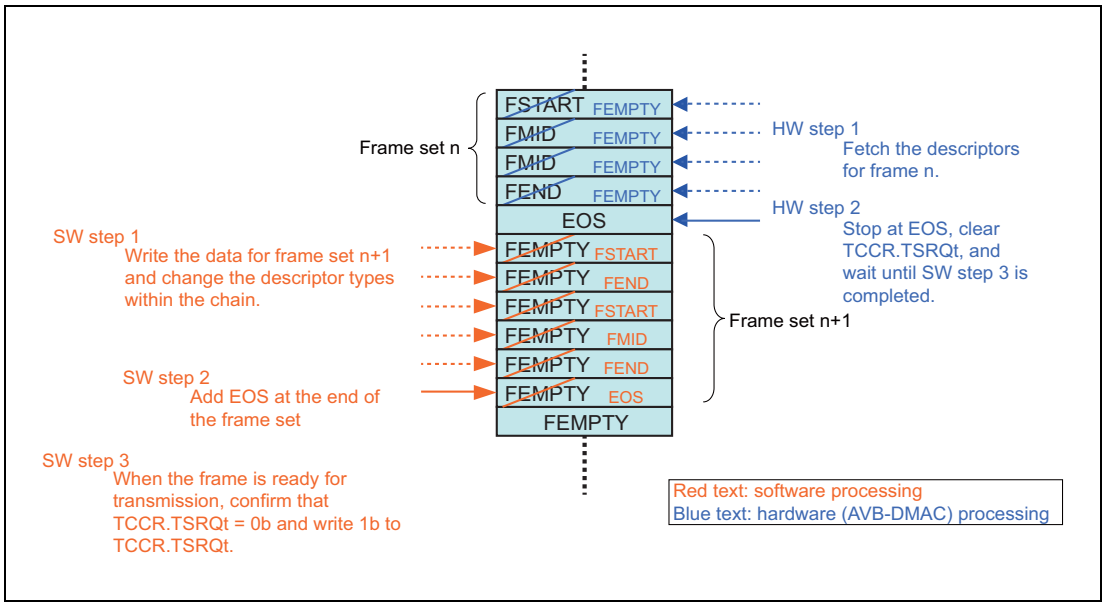


Figure 24.38 SW and AVB-DMAC Operations for Frame Set Transmission with Changing of the Active Descriptor Chain

(c) Frame Set Transmission Using a Shadow Descriptor Chain

This pattern is used when data are transmitted with a delay for software control to secure bandwidth or for other reasons, rather than immediately transmitted. Two or more descriptor chains are used. The chains are classified as active or shadow chains. EOS descriptors are used for the stop points.

Create descriptor chains that have FEMTPY descriptors at the stop points.

Figure 24.39 shows the flow for software implementing this pattern.

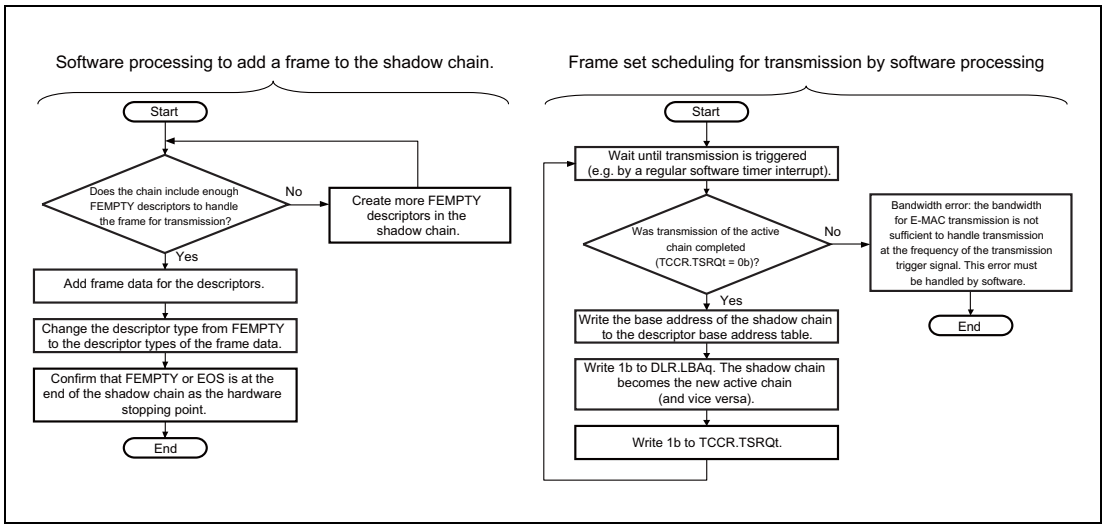


Figure 24.39 Software Flow for Frame Set Transmission Using the Shadow Descriptor Chain

Figure 24.40 shows software and AVB-DMAC operations for frame set transmission.

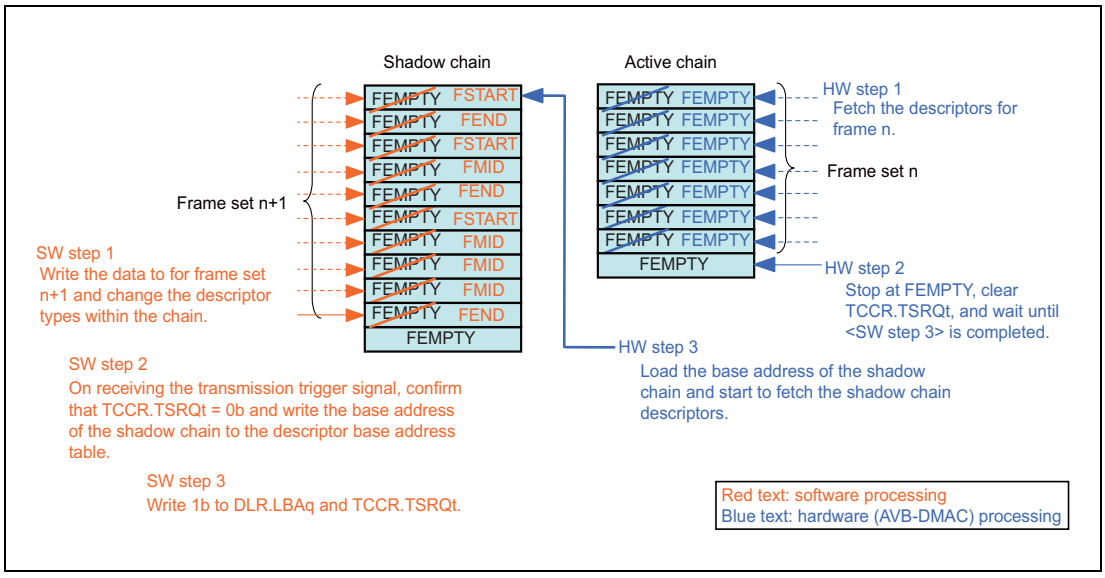


Figure 24.40 SW and AVB-DMAC Operations for Frame Set Transmission Using the Shadow Descriptor Chain

24.4.5.4 Time Stamping in Transmission

Transmission time stamps are important in satisfying the requirements for timing and synchronization of the IEEE 802.1AS standard. Reference to this information can also be useful to other applications and in testing. The AVB-DMAC supports the storage of time stamps for transmitted frames. The time-stamp values are based on the gPTP timer and are captured at the same time as sending of the Start of Frame Delimiter (SFD) for transmitted frames.

When the time stamp storage request field (DESCR.TSR) is set to 1, selecting storage of a time stamp, the tag number defined in the tag field (DESCR.TAG) of the last descriptor in a set (FEND) or of an FSINGLE descriptor for the frame being transmitted is stored with the time stamp. This eases identification and association. The time-stamp FIFO is accessible at any time.

Figure 24.41 shows the mechanism supporting transmission time stamping.

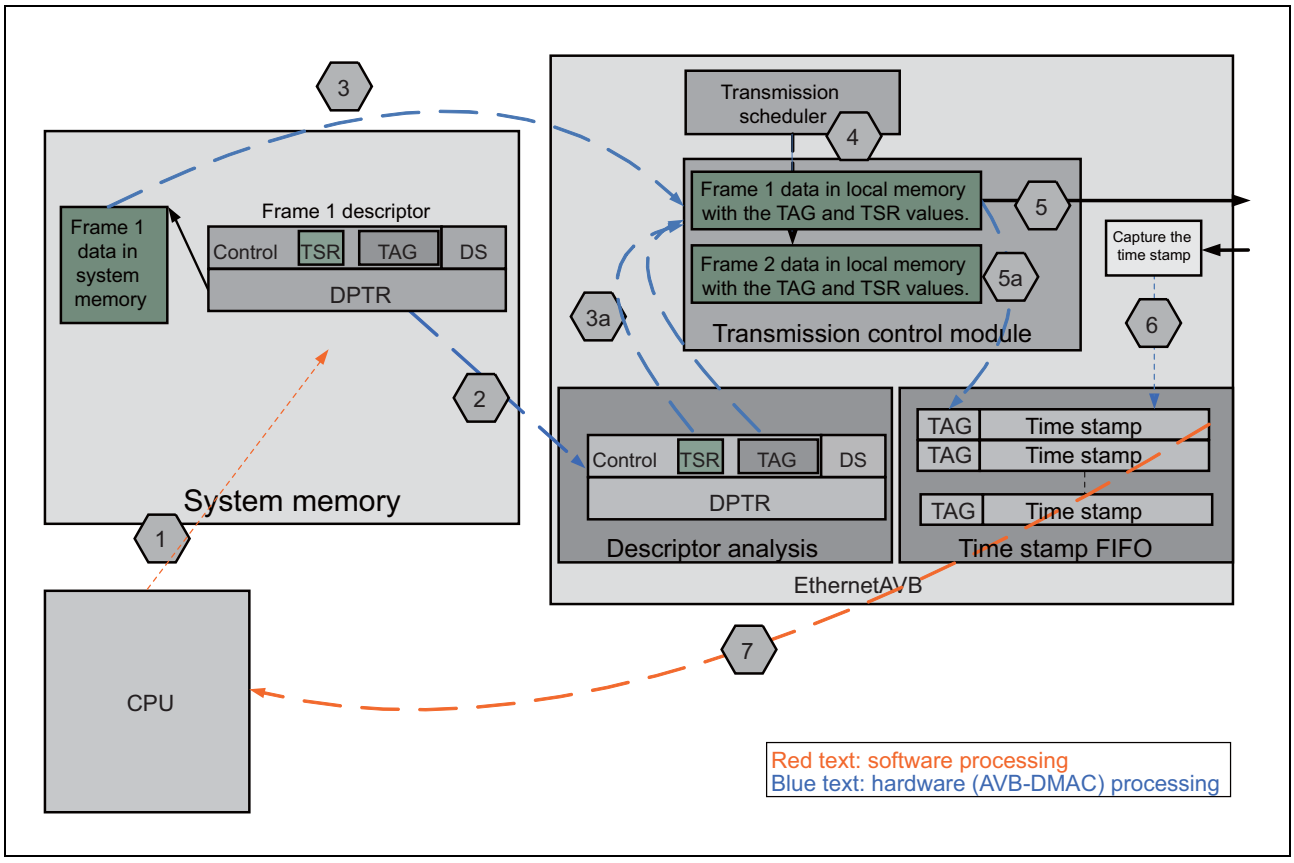


Figure 24.41 Mechanism to Support Transmission Time Stamps

The method of using this function is described below:

1. Secure space in the URAM for the frame requiring time stamping.
   Write the tag number of the frame to the frame tag field (DESCR.TAG) and set the time stamp storage request field (DESCR.TSR) to 1.
2. The AVB-DMAC fetches and analyzes the descriptor. The time stamp storage request field (DESCR.TSR) is 1, so it recognizes that transmitting this frame also requires storage of the time stamp.
3. The AVB-DMAC fetches the data for frame 1 and temporarily stores the frame in internal memory for scheduling.

   3a: The frame tag field (DESCR.TAG) and time stamp storage request field (DESCR.TSR) are stored with the fetched data.)
4. Under the control of priority settings according to credit-based shaping (CBS) or another scheme, the transmission scheduler decides it is time to transmit frame 1.
5. Transmission of frame 1 starts.

   5a: Frame 1's tag is stored in the time-stamp FIFO.
6. The gPTP time stamp is captured at the start of sending the frame delimiter (SFD) for transmission and stored with the tag in the time-stamp FIFO. On completion of the transmission, an interrupt is generated. For this to happen, the descriptor interrupt control register (DIC) must be set beforehand.
7. The entry can now be read from the time-stamp FIFO.

Use the time-stamp FIFO for the timing and synchronization of frames with IEEE 802.1AS compliance.

Time stamping can also be used with other frames, but take care not to allow the time-stamp FIFO to overflow. When the FIFO is full, further time stamps supplied to it are lost.

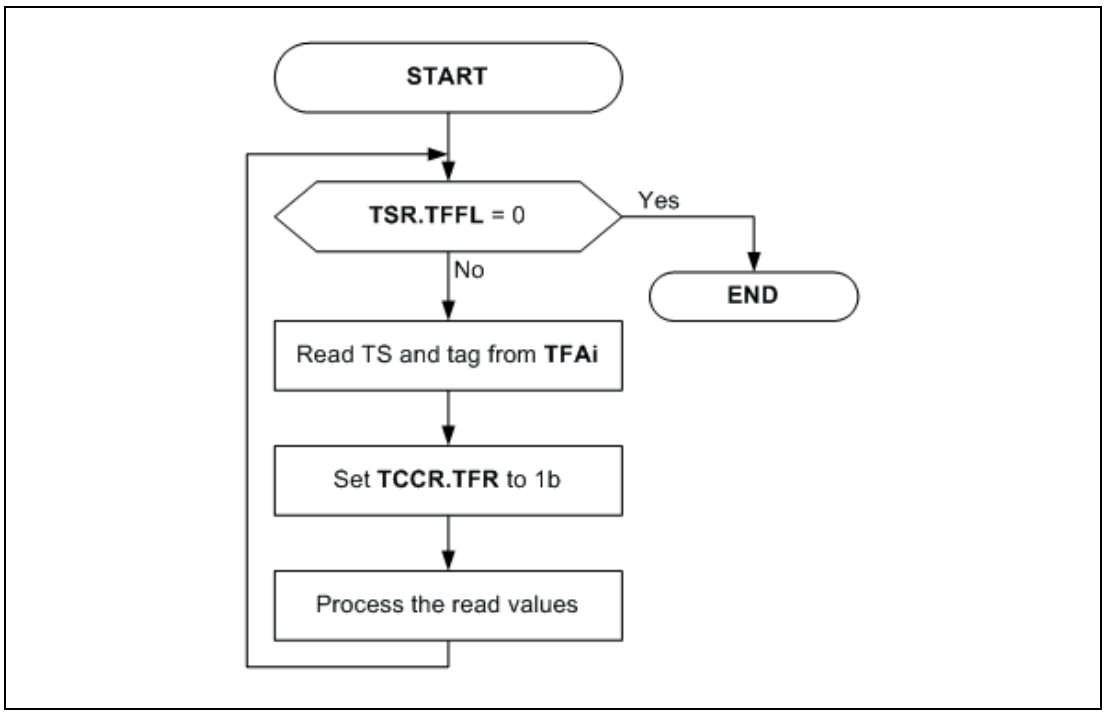


Figure 24.42 Flow of Transmission Time Stamping

(1) Ending Transmission

Figure 24.43 shows the procedure for ending transmission.

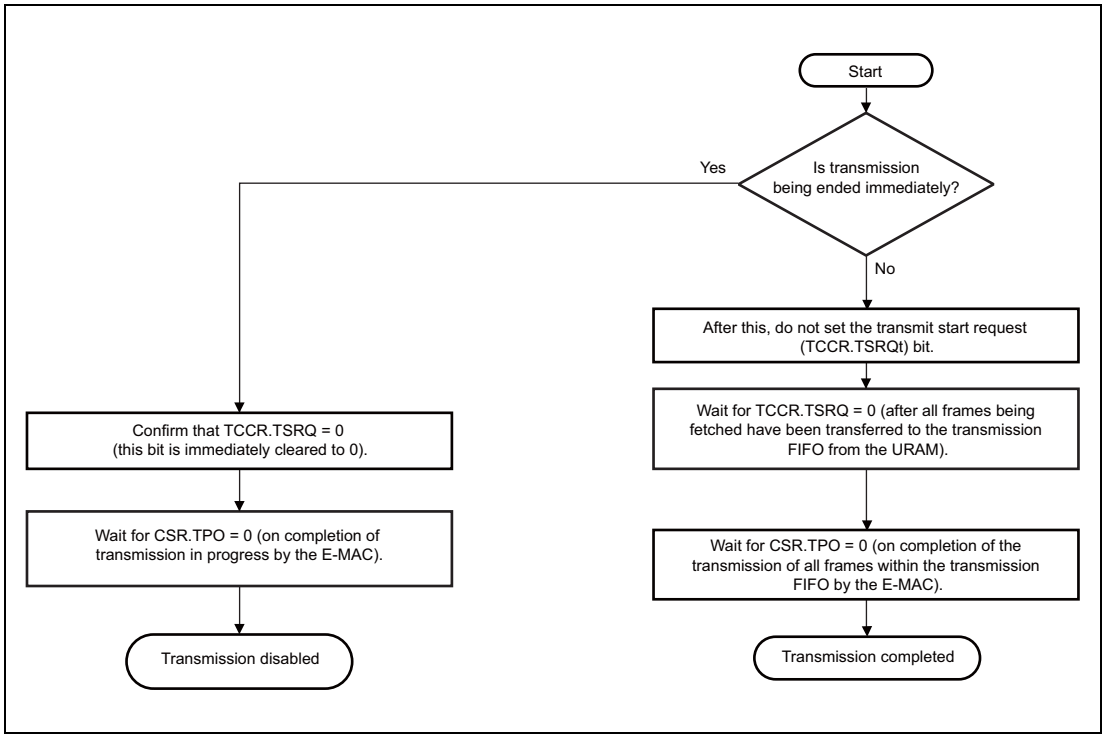


Figure 24.43 Procedures for Ending Transmission

24.4.6 CBS (Credit-Based Shaping)

In AVB transmission mode (i.e. when the transmit queue priority field in the transmit configuration register (TGC.TQP[1:0]) is B'01 or B'11), transmission queues Q3 and Q2 are respectively assigned to class A and class B stream traffic and the CBS (Credit Based Shaping) algorithm is used to select the transmission queues in order to satisfy the Forwarding and Queuing for Time Sensitive Streams (FQTSS) specification (see section 8.6.8 or section 34 in IEEE 802.1Q).

The CBS algorithm is based on the concept of transmission credit for each queue. Credit can be thought of as the degree to which a queue has the "right" to transmit at a given time. Actually, in AVB transmission mode as specified in IEEE 802.1Q, queues that are subject to the CBS algorithm are able to transmit when the following conditions are met.

- At least one frame is stored in the queue.
- The credit for the queue is 0 or a positive value.

The credit for a transmission queue is incremented while one or more frames from the queue are present in the transmission FIFO but transmission of these frames is not proceeding. This state is indicated by the transmission process status bit for queue t in the AVB-DMAC status register (CSR.TPOt). The credit is decremented while transmission of a frame from the queue is in progress. This mechanism is used to control transmission so that the transmission of frames from the queues for each of the traffic classes does not exceed the specified maximum bandwidths.

IEEE 802.1Q defines the following parameters for queues under the control of the CBS algorithm.

portTransmitRate: Maximum transmission data rate of an external port. The E-MAC determines this parameter.

bandwidthFraction: Maximum fraction of portTransmitRate that can be used for a queue.

idleSlope: Rate of change of credit for a queue when transmission of frames from the queue is not proceeding so the credit value (in bits per second) is increasing. idleSlope is also equal to the maximum fraction of the total bandwidth (portTransmitRate) that is available to the given queue under a specified condition (frames from the queue can be placed in a continuous stream. See Annex L of IEEE 802.Q.
idleSlope = bandwidthFraction * portTransmitRate

sendSlope: Rate of change of credit for a queue while transmission of a frame from the queue is in progress so the credit value is decreasing.
The value of sendSlope is defined as follows:
sendSlope = idleSlope - portTransmitRate

Furthermore, the values below are used to define individual traffic classes (or queues for the classes) under control of the algorithm. See Annex L of IEEE 802.Q.

maxFrameSize: Maximum size of frames (in bits) of the corresponding traffic class that can be transmitted from a port

maxInterferenceSize: Maximum burst size (in bits) by which delays for the corresponding traffic class can be allowed

hiCredit: Maximum credit value (positive number). Can be calculated by using the following equation: hiCredit = maxInterferenceSize * (idleSlope / portTransmitRate)

loCredit: Minimum credit value (negative number). Can be calculated by using the following equation: loCredit = maxFrameSize * (sendSlope / portTransmitRate)

Figure 24.44 shows how the CBS algorithm works and the meaning of the above parameters.

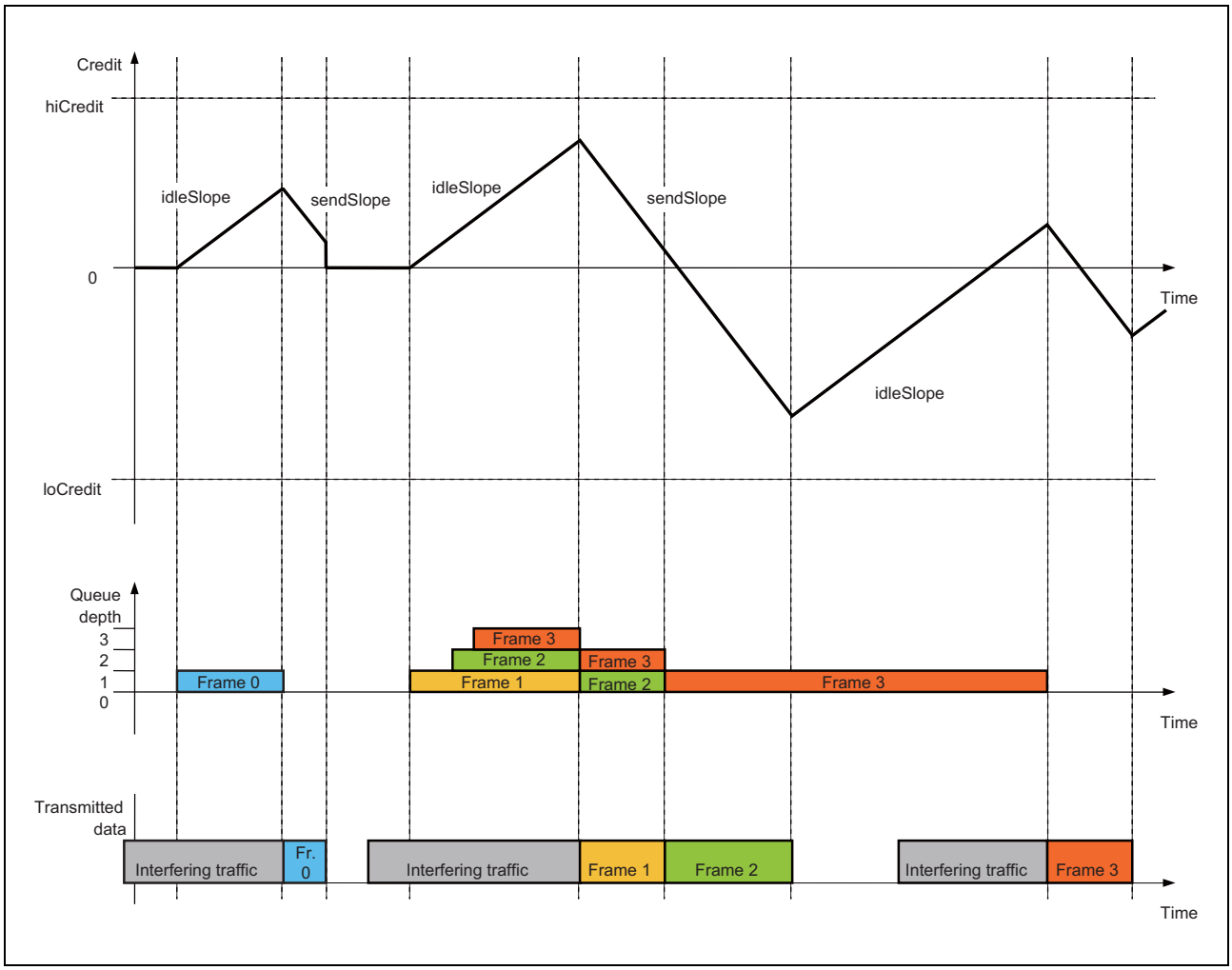


Figure 24.44 CBS (Credit-Based Shaping) Operation

Figure 24.45 shows the implementation of CBS in the AVB-DMAC.

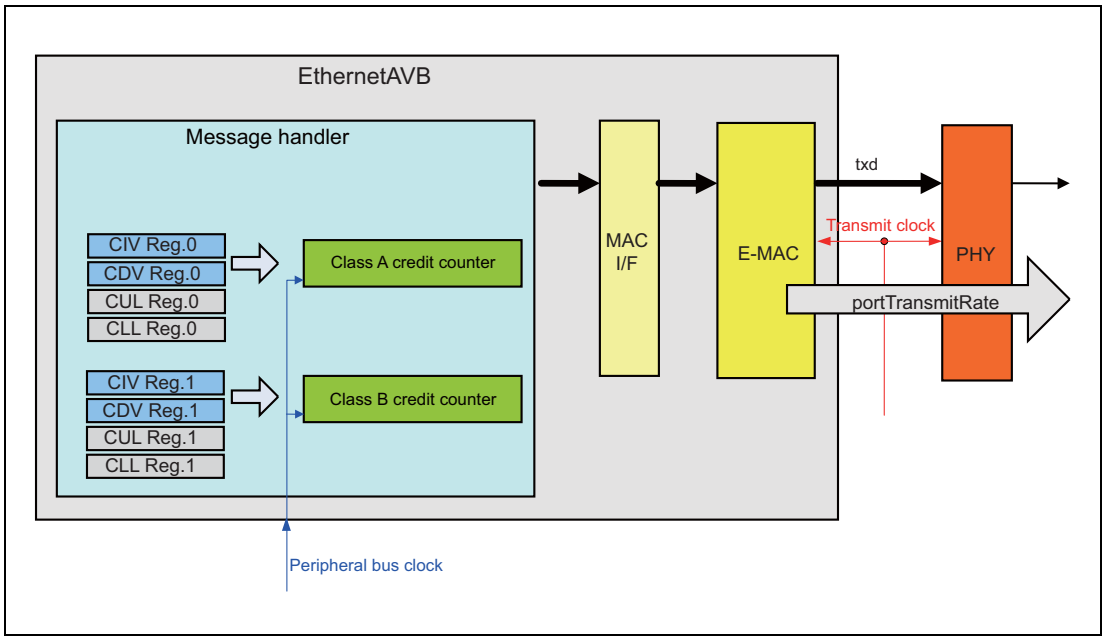


Figure 24.45 CBS (Credit-Based Shaping) Operation in the AVB-DMAC

The above implementation is based on “credit counters” for the respective traffic classes (SR class A and class B). The following parameters apply for these classes.

CBS increment value (CIV): Signed positive number
The credit is incremented by this amount every peripheral bus clock cycle while a frame from the queue is pending but transmission has not started (idleSlope).

CBS decrement value (CDV): Signed negative number
The credit is decremented by this amount every peripheral bus clock cycle while transmission of a frame from the queue is proceeding (sendSlope).

The CBS increment value (CIV) and CBS decrement value (CDV) are defined as follows.

CIV = idleSlope * Mfactor

CDV = sendSlope * Mfactor

Mfactor is a multiplier factor to ensure accuracy for CIV and CDV. CIV and CDV are calculated by using the following equations.

CIV = (portTransmitRate/CHI_freq) * bwFraction * Mfactor

CDV = (portTransmitRate/CHI_freq) * (bwFraction - 1) * Mfactor

CHI_freq is the frequency of the peripheral bus clock. The credit counters are driven by the peripheral bus clock, so calculating the slope parameters for CBS requires (1/CHI_freq).

Use software to prepare Mfactor for the CBS parameters. All queues for the same class must have the same CBS parameters. Mfactor for a specified class c can be changed during operation, unless transmission is pending for that class (i.e. the transmit process status bit in the AVB-DMAC status register (CSR.TPOt) = 0). At that time, the credit counter values for class A and class B are 0. Note that

the credit value will not match a new incrementation or decrementation parameter if Mfactor is changed while the credit counter value is non-zero. Mfactor is not present in the AVB-DMAC registers.

Set the CIV and CDV parameters in the CBS increment value registers (CIVRc) and the CBS decrement value registers (CDVRc). These are treated as dynamic settings since they should be updated when streams are registered and erased in accord with IEEE 802.1Qat.

The AVB-DMAC also has CBS upper limit registers (CULc) (the upper limit registers for classes A and B) and CBS lower limit registers (CLLc) (the lower limit registers for classes A and B). Set Mfactor to match the credit value and set the upper limit (hiCredit) and the lower limit (loCredit) for each class as defined above.

CUL = hiCredit * Mfactor = maxInterferenceSize * bwFraction * Mfactor

CLL = loCredit * Mfactor = maxFrameSize * (bwFraction - 1) * Mfactor

Example:

Assume that portTransmitRate = 100 Mbps, CHI_freq = 130 MHz and bwFraction = 3%. Then idleSlope and sendSlope represented as one bit vs. cycles of the peripheral bus clock are as follows.

idleSlope = (portTransmitRate/CHI_freq) * bwFraction = 100/130 (Mbps/MHz) * 3% = 0.023 of a bit per high-speed peripheral bus clock cycle

sendSlope = idleSlope - (portTransmitRate / CHI_freq) = –0.746 bits per peripheral bus clock cycle

Let Mfactor be 100, then CIV and CDV parameters are determined as follows.

CIV = idleSlope * Mfactor = 2.3

CDV = sendSlope * Mfactor = –74.6

24.4.6.1 Restrictions on CIV, CDV and Mfactor

The maximum value (the minimum value for negative numbers) up to which the credit counter will not overflow determines the maximum values of CIV and CDV that can be set in the CBS registers. This maximum credit value is equivalent to the worst case of the hiCredit value, and the maximum values for class A and class B are calculated as follows.

<Conditions>

- Class A maximum value (hiCredit_max_classA)
  classA bwFraction ≅ 100%
  Maintaining the proper relations in the transmission priority order requires waiting for a period equivalent to the maximum frame size.
  hiCredit_max_classA ≅ maxInterferenceSize for class A = Interference due to one max. sized frame = header + max. size payload + CRC (2000 bytes) + preamble (8 bytes) + IFG (12 bytes) + processing_delay (≅ 80 bytes) ≅ 2100 bytes
- Class B maximum value (hiCredit_max_classB)
  classB bwFraction ≅ 100%
  Maintaining the proper relations in the transmission priority order requires waiting for a period equivalent to the maximum size of frames in the class A transmission queue and other transmission queues.
  hiCredit_max_classB ≅ maxInterferenceSize for class B = Interference due to two max-size frames = 2* hiCredit_max_classA ≅ 4200 bytes

hiCredit_max_classA = 16800

hiCredit_max_classB = 33600

The maximum values that can be selected with Mfactor for the 32-bit signed counter without overflow are:

Mfactor_max_classA = 2^{31}-1 / hiCredit_max_classA ≅ 127826 and
Mfactor_max_classB = 2^{31}-1 / hiCredit_max_classB ≅ 63913.

A high degree of accuracy can be achieved even with a low bandwidth. In class B, bandwidthFraction = 0.05% and the bandwidth error < 0.1%.

The maximum value of CIV is calculated from the following equation.

CIV = idleSlope x Mfactor = (portTransmitRate / CHI_freq) * bandwidthFraction × Mfactor

When Mfactor is the maximum value and bandwidthFraction is the maximum value (up to 100%):

```
CIV_max_classA =  (portTransmitRate / CHI_freq) * Mfactor_max_classA and
CIV_max_classB =  (portTransmitRate / CHI_freq) * Mfactor_max_classB.
```

Table 24.87 shows examples of values for portTransmitRate and peripheral clock frequency. Note that the values in the table are the limits of CIV up to which the 32-bit credit counter will not overflow. The CIV parameters are implemented as 16 bits + a sign bit, so a further limit of CIV ≤ 65535 applies to both class A and class B.

Table 24.87 Example of Maximum Values for Class A and Class B CIV Parameters

portTransmitRate	CHI_freq [MHz]	CIV_max_classA	CIV_max_classB
100Mbps	50	255652	127826
100Mbps	80	159782	79891
100Mbps	100	127826	63913
100Mbps	125	102260	51130
100Mbps	133	96109	48054
100Mbps	150	85217	42608

24.4.6.2 Credit Incrementation during Inter-Frame Gaps (IFGs)

The inter-frame gap (IFG) after a frame is transmitted is not treated as part of frame transmission by the CSB credit counter. During an IFG, the credit is incremented for all SR queues that have pending frames or negative credit. **Figure 24.46** illustrates credit operations during IFGs.

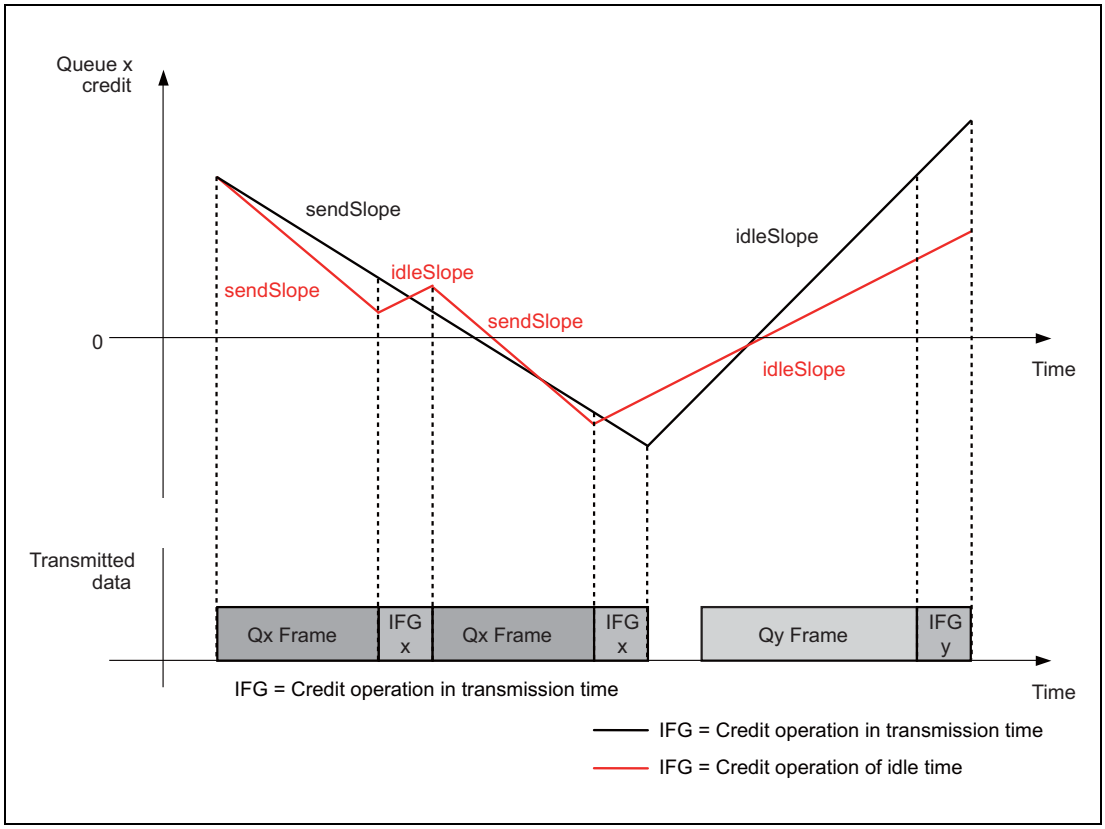


Figure 24.46 Credit Operations during IFGs

Accordingly, the IFG need not be included in calculation of the bandwidth requirements for the specified SR class when deciding the idleSlope, sendSlope, and CIV and CDV parameters. However, IFG must also be included in the calculation in order to confirm that the total bandwidth allocated to all SR classes does not exceed 100% of portTransmitRate. This is described in section 35.2.2.8.4 of IEEE 802.1Q.

24.4.6.3 Example

The case of a class A 48-kHz stereo audio stream among Ethernet frames is described as an example.

After every class A measurement interval (125 us), 80 octets consisting of two sets of six 32-bit samples plus a 32-octet header are stored as audio data within a frame. The IEEE 802.3 also imposes a 42-octet media-specific framing overhead (an 8-octet preamble, 14-octet IEEE 802.3 header, 4-octet IEEE 802.1Q priority/VID Tag, 4-octet CRC, and 12-octet IFG) are also added. Accordingly, the total frame size is 80 + 42 = 122, and one such frame is transmitted after every class measurement interval.

This represents a total bandwidth of about 7.8 Mbits per second (122 octets × 8 bits per octet × 8000 frames per second) for this class.

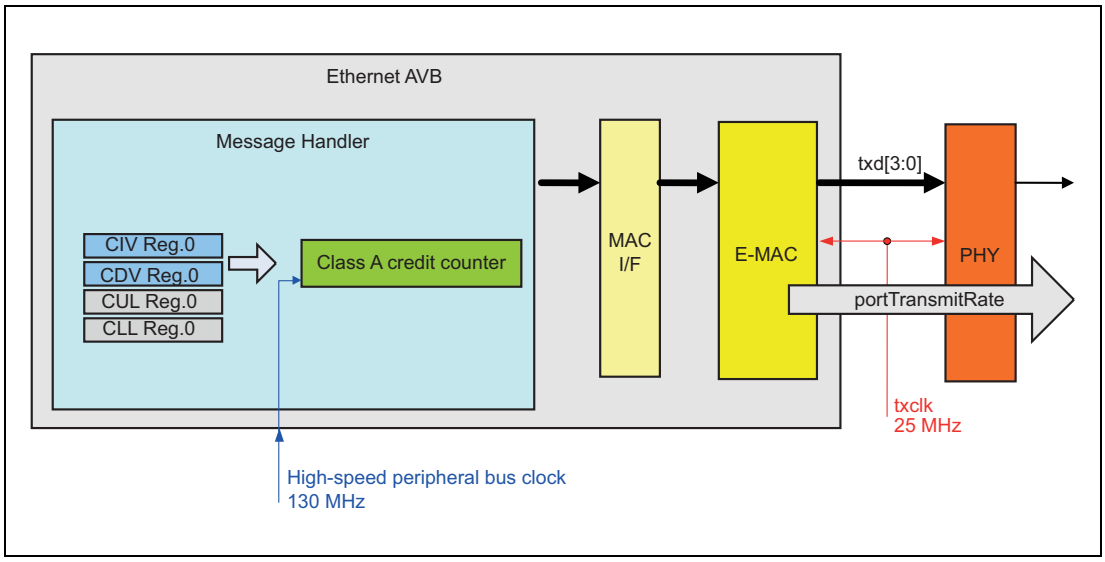


Figure 24.47 Example of CBS Settings

Given that:

- the E-MAC runs at 100 Mbps, so portTransmitRate = 100 Mbps and
- high-speed peripheral bus clock (operating clock for the credit counter) frequency = 130 MHz,

securing a bandwidth of 7.04 Mbits/sec for class A requires configuring the CBS parameters as follows.

- bandwidthFraction = 0.704%
- idleSlope = (portTransmitRate / CHI_freq) * bandwidthFraction ≅ 0.0176 bits per high-speed peripheral bus clock cycle
- sendSlope = idleSlope – (portTransmitRate / CHI_freq) ≅ -2.4824 bits per high-speed peripheral bus clock cycle

When Mfactor = 100000, the parameters are as follows.

- CIV = idleSlope × Mfactor = 1760 bits per high-speed peripheral bus clock cycle
- CDV = sendSlope × Mfactor = 248240 bits per high-speed peripheral bus clock cycle

These are the final values for setting in the CIVR1 and CDVR1 registers.

24.4.7 IEEE802.1: gPTP

24.4.7.1 gPTP Timer

An 84-bit timer is provided to support the gPTP function. **Figure 24.48** shows the definitions of bits for the timer and in related registers.

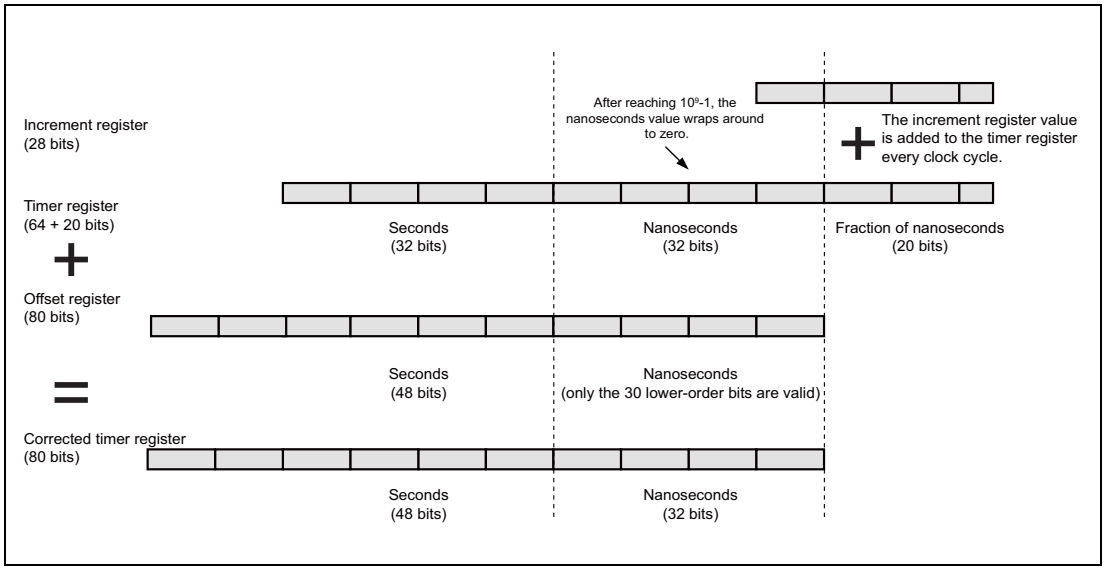


Figure 24.48 Definitions of gPTP Timer Bits and Related Bits

The higher-order 32 bits indicate seconds. For the next 32 bits, counting by one corresponds to the passage of 1 ns. The lower-order 20 bits are a fractional value (less than 1 ns). Software can only read the 32 higher-order bits, indicating seconds, and the subsequent 32-bits, indicating nanoseconds. The 20 lower-order 20 bits, representing less than 1 ns, are not readable. They are only used within the AVB-DMAC to maintain accuracy in time measurement.

The timer can be reset by setting the timer control request bits in the gPTP configuration control register (GCCR.TCR[1:0]) to B'01. These bits are set to B'00 on completion of normal resetting of the timer.

After the timer starts, the value in the gPTP timer increment register (GTI.TIV[27:0]) is added to the value of the gPTP timer every clock cycle.

Before setting a value in the gPTP timer increment register (GTI.TIV[27:0]), set the timer increment value setting request bit in the gPTP configuration control register (GCCR.LTI). If this bit is not set to 1, new values that are written will not be reflected in the register. This bit returns to 0 after the setting is completed.

An offset to the gPTP timer is also available. If this is required, set the value in the gPTP timer offset register (GTO.TOV[31:0]). Before setting a value in this register, set the timer offset value setting request bit in the gPTP configuration control register (GCCR.LTO). If this bit is not set to 1, new values that are written will not be reflected in the register. This bit returns to 0 after the setting is completed. When adding an offset, take care that it does not exceed 80 bits.

The value of the gPTP timer can be read from the gPTP timer capture register (GCTi.CTV[31:0]). Set the timer capture source select bits in the gPTP configuration control register (GCCR.TCSS[1:0]) to select the timer value for capture as the value of the gPTP timer, the corrected value of the gPTP timer (value with the offset added), or the AVTP presentation time. Setting the timer control request bits in the gPTP configuration control register (GCCR.TCR[1:0]) to B'11 initiates the capture. Once normal

capture of the timer is complete, the value of the timer control request bits in the gPTP configuration control register (GCCR.TCR[1:0]) returns to B'00.

The timer for gPTP operates as a free-running timer but can be synchronized with the Grandmaster clock.

24.4.7.2 Free-Running Operation

The IEEE 802.1 AS standard for timing and synchronization does not prescribe the physical adjustment of local clocks to the Grandmaster clock. To avoid negative effects from the correction procedure, we recommend the use of a free-running timer.

As a free-running timer, the timer counts the local time in seconds or nanoseconds. The gPTP timer increment register (GTI.TIV[27:0]) is set to 1 ns (the setting value = H'0010 0000) and the gPTP timer offset register (GTOi.TOV[31:0]) is set to 0. The ratio information captured at the time of the gPTP delay measurement and synchronization procedures is used to correct the frequency ratio to that of the Grandmaster clock. The Grandmaster clock can be calculated from the local clock by using the information collected during the gPTP measurement and synchronization procedures.

24.4.7.3 Synchronization with the Grandmaster Clock

In situations requiring physical synchronization of the local clock with the Grandmaster clock, the fractional nanoseconds value (the 20 lower-order bits of the gPTP timer) is used to make the adjustment. Specifically, the increment value is finely adjusted to correct for deviations of the clock frequency from that of the Grandmaster clock.

Use the timer offset value (in the gPTP timer offset registers, GTOi.TOV[31:0]) to correct for offsets from the theoretical value (at start-up, etc.). The sum of the timer value and the offset register is the "corrected timer" value.

Note that for the nanoseconds part of the Offset register GTOi.TOV[31:0] should be below 10^9.

The following equation gives a method of calculating the increment (GTI.TIV[27:0]) from the frequency of the gPTP clock and its deviation from that of the Grandmaster clock. Variable d is the deviation ($d = 10^{-6}$ for 1 ppm).

$$\mathrm{GTI.TIV[27:0]} = \mathrm{round}\left(\frac{2^{20}\mathrm{GHz}}{f_{GPTP}} \times (1 + d)\right)$$

After adjusting for the current deviation of clock frequency, re-set the gPTP timer increment register (GTI.TIV[27:0]).

After calculating the new offset value, re-set the gPTP timer offset register (GTOi.TOV[31:0]).

24.4.7.4 Support Provided by the gPTP Timer in Transmission and Reception

The timer value described above is used in the time-stamp values captured when start frame delimiters are detected in reception and generated in transmission.

Captured time stamp values for received frames are stored in the corresponding descriptors. Those for transmitted frames are stored with tag information in the time-stamp FIFO. The time stamp values are thus correlated with both transmitted and received frames.

Note that the use of corrected timer values can introduce an error due to the offset correction in the gPTP synchronization procedure.

Errors due to the SDF notification and the asynchronous interface between the timer modules must also be taken into account.

24.4.8 Support for IEEE 1722

For IEEE 1722, the following two functions are supported.

- Output and capture of values in the IEEE 1722 AVTP (Audio/Video Transport Protocol) presentation time format
- Comparison of IEEE 1722 AVTP presentation time stamps

The 32-bit AVTP time stamp field of IEEE 1722 frames holds the AVTP presentation time when the AVTP time-stamp enable bit in the frame is 1. The AVTP time stamp field is generated from the gPTP timer and is given as seconds (gPTP_seconds) and nanoseconds (gPTP_nanoseconds) according to the following equation.

AVTP time stamp = (gPTP_seconds * 10^9 + gPTP_nanoseconds) modulo 2^{32}

The AVTP presentation time can be read from the gPTP timer capture register (GCTi.CTV[31:0]). Set the timer capture source select bits in the gPTP configuration control register (GCCR.TCSS[1:0]) to select the timer value for capture as the AVTP presentation time. Setting the timer control request bits in the gPTP configuration control register (GCCR.TCR[1:0]) to B’11 initiates the capture. The value is obtained by adding the maximum transit time defined in the gPTP maximum transit time register (GMTT.MTTV[31:0]) to the corrected timer value. The AVTP presentation time wraps around approximately every four seconds.

CAUTION

The AVTP presentation time captured in GCTi.CTV[31:0] is only valid when the corrected timer value is in synchronization with the Grandmaster clock. That is, the timer increment and timer offset values for the corrected timer value must be adjusted during the synchronization procedure so that the corrected gPTP clock is physically adjusted to match the time kept by the Grandmaster clock.

24.4.9 Flow Control

The EthernetAVB does not support flow control in the specification. Flow control is only effective when transmitting and receiving normal (non-AVB) packets.

The E-MAC supports flow control for full-duplex operation in compliance with the IEEE 802.3 standards. This flow control is applicable to both reception and transmission. In regard to the transmission of PAUSE frames, flow control operates in the following ways.

(1) PAUSE Frame Transmission

PAUSE frames can also be transmitted in response to software operations. Writing a timer value to the manual PAUSE frame register (MPR) starts the transmission of a PAUSE frame. This only causes the transmission of one PAUSE frame.

(2) PAUSE Frame Reception

After reception of a PAUSE frame, transmission of the next frame does not proceed until the time indicated by the Timer value elapses. However, transmission of a frame currently being transmitted continues. PAUSE frames are only received while the RXF bit in the E-MAC mode register (ECMR) is set to 1. The number of received PAUSE frames is counted.

(3) PAUSE Frames with the Timer Value 0

The setting of the 0-time PAUSE frame enable bit (ECMR.ZPF) enables or disables the reception and transmission of PAUSE frames with the TIME parameter value 0.

- When control of PAUSE frames with the TIME parameter value 0 is enabled
  A PAUSE frame with the TIME parameter value 0 is transmitted when the capacity of the reception FIFO is less than the value of the overflow alert FIFO threshold register (FCFTR) while the time indicated by the TIME parameter value has not elapsed.
  Reception of a PAUSE frame with the TIME parameter value 0 leads to release from the transmission standby state.
- When control of PAUSE frames with the TIME parameter value 0 is disabled
  PAUSE frames with the TIME parameter value 0 are not transmitted. Received PAUSE frames with the TIME parameter value 0 are discarded.

(4) Back Pressure flow control

In half-duplex mode E-MAC uses back pressure for flow control. If receiver function is busy, then E-MAC transmit the imitate of collision frame and E-MAC stop the frame reception.

24.4.10 Magic Packet Detection

The E-MAC has a Magic Packet detection function. This function provides a facility for host devices and other sources to start other peripheral devices connected to a LAN. A peripheral device that handles Magic Packets starts itself in response to receiving a Magic Packet.

When a Magic Packet is detected, data from broadcast packets that were previously being received are stored in the FIFO and the E-MAC is notified of the receiving status. To return to normal operation from the associated interrupt processing, the E-MAC and AVB-DMAC must be initialized by using the operating mode configuration bit in the AVB-DMAC mode register (CCC.OPC[1:0]) to set the operating mode to reset mode.

The procedure for using the Magic Packet detection function with this LSI chip is as follows.

1. Use the various interrupt enabling and masking registers to disable the output of interrupts from interrupt sources.
2. Set the Magic Packet detection enable bit in the E-MAC mode register (ECMR.MPDE).
3. Set the Magic Packet detection interrupt enable bit in the E-MAC interrupt enable register (ECSIPR.MPDIP) to enable the interrupt.
4. Place the CPU in sleep mode and peripheral modules in module standby mode as required.
5. An interrupt is conveyed to the CPU on detection of a Magic Packet.

NOTE

The Magic Packet detection interrupt status can be read in the E-MAC status register (ECSR.MPD). The bit can be cleared by setting the E-MAC status register Magic Packet TM detection bit (ECSR.MPD) to 1.

24.4.11 Interrupts

The EthernetAVB module has three EI level interrupts from the AVB-DMAC and one EI level interrupt from the E-MAC.

Table 24.88 is a list of the interrupts and the relationships between EIINT interrupt channel numbers and source codes.

Table 24.88 EthernetAVB Interrupts

Interrupt Source Name	EIINT Interrupt Channel Number	Source Code
Transmit/receive data management interrupt	244	H'10F4
Error management interrupt	245	H'10F5
Other management (FIFO caution level, etc.) interrupt	246	H'10F6
E-MAC interrupt	247	H'10F7

The AVB-DMAC related interrupts include descriptor interrupts (15 sources), error interrupts (5 sources), reception interrupts (37 sources), transmission interrupts (2 sources), and gPTP interrupts (3 sources). From the CPU's perspective, each appears as one of the above four interrupt sources.

The states of an AVB-DMAC-related interrupt sources can be checked in the following registers.

- Descriptor interrupt status register (DIS)
- Error interrupt status register (EIS)
- Receive interrupt status register (RISi)
- Transmit interrupt status register (TIS)
- gPTP interrupt status register (GIS)

The interrupts are controlled by the corresponding interrupt enable bits. However, the status flags operate independently of the settings of the enable bits.

The states of grouped interrupts can only be checked by reading the interrupt summary status register (ISS) and the queue full error interrupt status bit in the error interrupt status register (EIS.QFS). This reduces the load on the CPU.

NOTE

Target of Descriptor interrupt or Receive interrupt is before address is shown by Current descriptor address register CDARq. There is a possibility CPU start operation before writing to memory for Descriptor interrupt or Receive interrupt.

24.4.11.1 Transmit/Receive Data Management Interrupt

The management interrupt for transmission and reception is conveyed when the interrupt conditions corresponding to the following sources are satisfied.

- Receive frame interrupt in the receive interrupt status register 0 (RIS0.FRFr)
- Descriptor interrupt in the descriptor interrupt status register (DIS.DPFi)

The general error interrupt state can be checked by reading the descriptor interrupt status bits in the interrupt summary status register (ISS.DPSi) or the receive FIFO warning interrupt summary bit (ISS.RFWS).

24.4.11.2 Error Management Interrupt

The error management interrupt is conveyed when interrupt conditions corresponding to the following sources are satisfied.

- Time stamp FIFO full interrupt in the error interrupt status register (EIS.TFFF)
- CBS limitation interrupts in the error interrupt status register (EIS.CULF1, EIS.CULF0, EIS.CLLF1, EIS.CLLF0)
- Receive FIFO full interrupt in the receive interrupt status register 2 (RIS2.RFFF)
- Receive queue full interrupt in the receive interrupt status register 2 (RIS2.QFFr)

The general error interrupt state can be checked by reading the error interrupt summary bit in the interrupt summary status register (ISS.ES).

24.4.11.3 Other Management (FIFO Warning, etc.) Interrupts

The other management (FIFO warning, etc.) interrupt is conveyed when interrupt conditions corresponding to the following sources are satisfied.

(1) Reception related interrupt

Receive FIFO warning interrupt in the receive interrupt status register 1 (RIS1.RFWF)

(2) Transmission related interrupts

Time stamp FIFO warning interrupt in the transmit interrupt status register (TIS.TSWF)

Time stamp FIFO update interrupt in the transmit interrupt status register (TIS.TSUF)

(3) gPTP related interrupts

AVTP presentation target match interrupt in the gPTP interrupt status register (GIS.PTMF)

The general error interrupt state can be checked by reading the receive FIFO warning error interrupt status bit in the interrupt summary status register (ISS.RFWS), the time stamp FIFO warning interrupt status bit (ISS.TFWS), and the time stamp FIFO update interrupt status bit (ISS.TFUS).

24.4.11.4 E-MAC Interrupt

The E-MAC interrupt is conveyed when the E-MAC interrupt source is generated.

The general error interrupt state can be checked by reading the E-MAC interrupt summary bit in the interrupt summary status register (ISS.MS).

24.4.12 Flows of Operations

24.4.12.1 Flow of E-MAC Initialization

Figure 24.49 shows the flow of E-MAC initialization (for AVB mode and full-duplex operation).

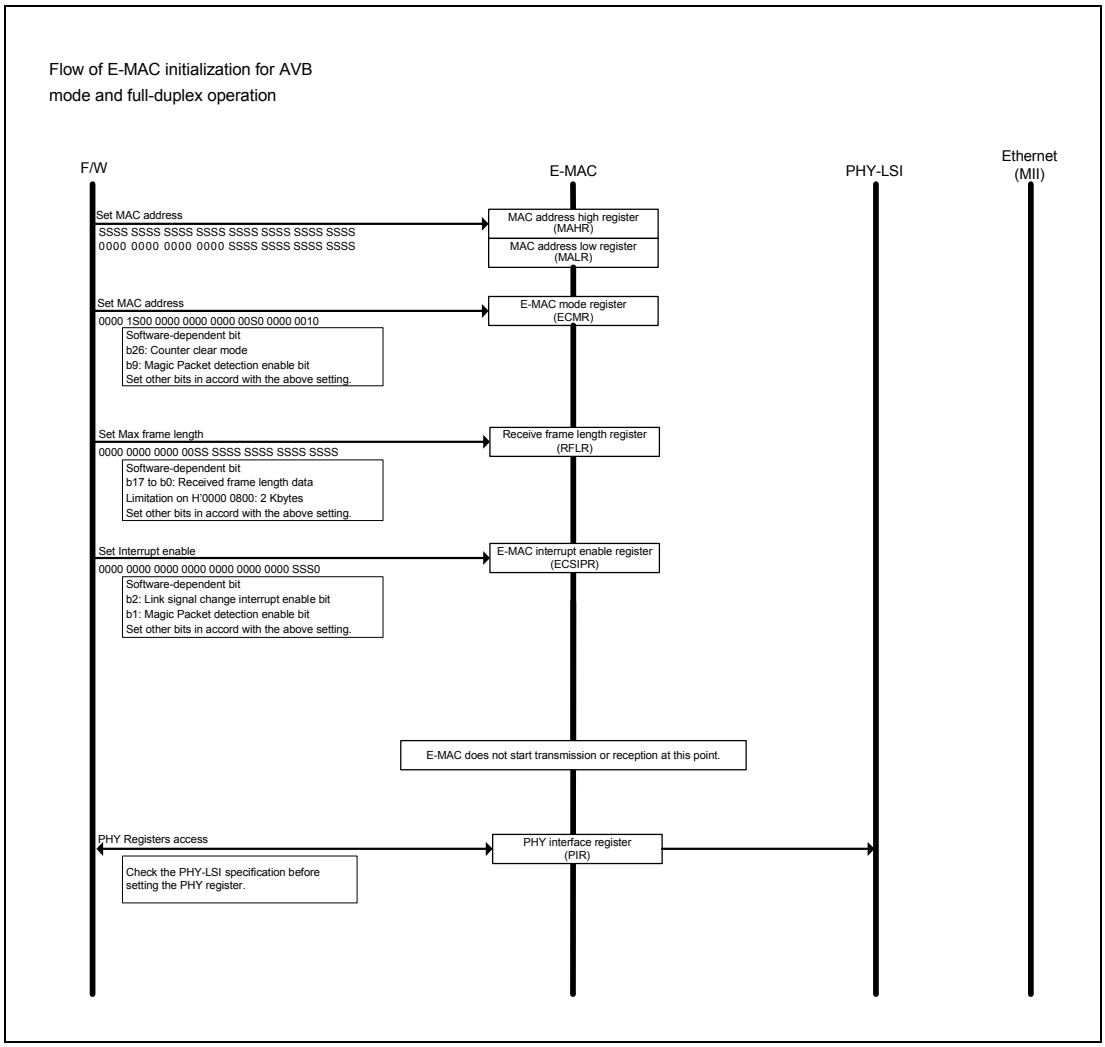


Figure 24.49 Flow of E-MAC Initialization (for AVB Mode and Full-Duplex Operation)

24.4.12.2 Flow of AVB-DMAC Initialization

Figure 24.50 shows the flow of AVB-DMAC initialization (for AVB mode and full-duplex operation).

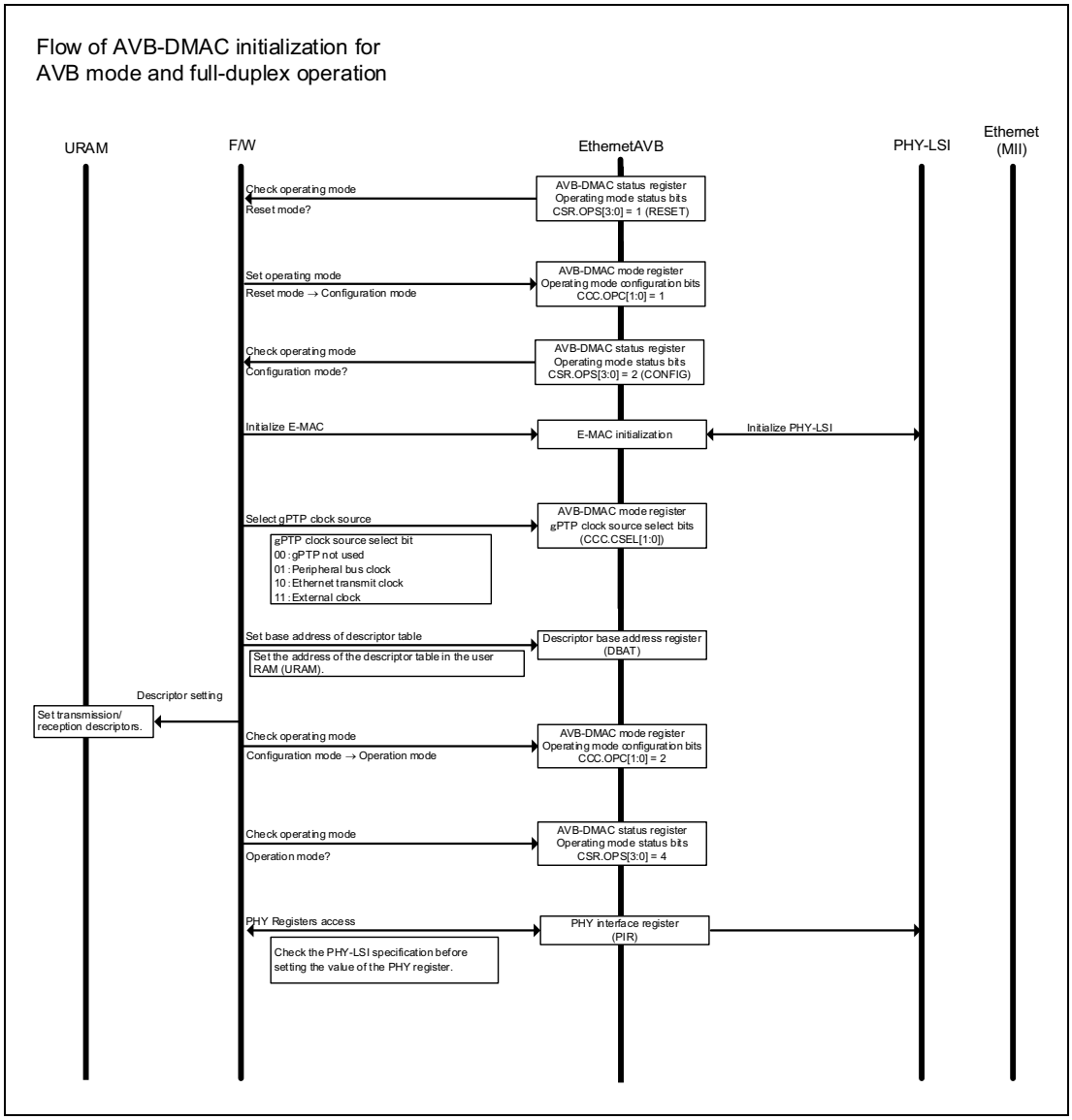


Figure 24.50 Flow of AVB-DMAC Initialization (for AVB Mode and Full-Duplex Operation)

24.4.12.3 Flow for the AVB-DMAC in Reception

Figure 24.51 shows the flow for the AVB-DMAC in reception (in AVB mode and full-duplex operation).

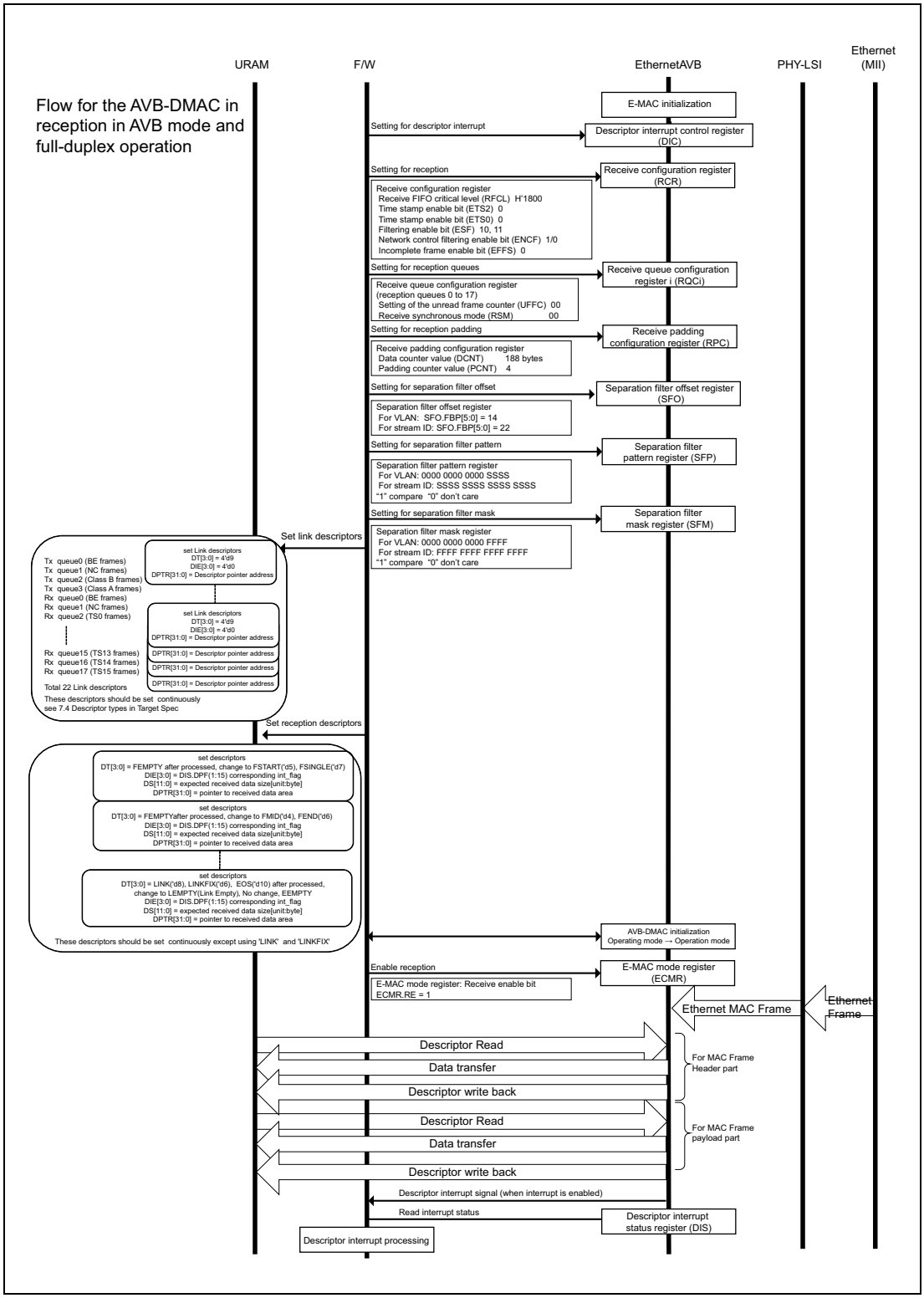


Figure 24.51 Flow for the AVB-DMAC in Reception (in AVB Mode and Full-Duplex Operation)

24.4.12.4 Flow for the AVB-DMAC in Transmission

Figure 24.52 shows the flow for the AVB-DMAC in transmission (in AVB mode and full-duplex operation).

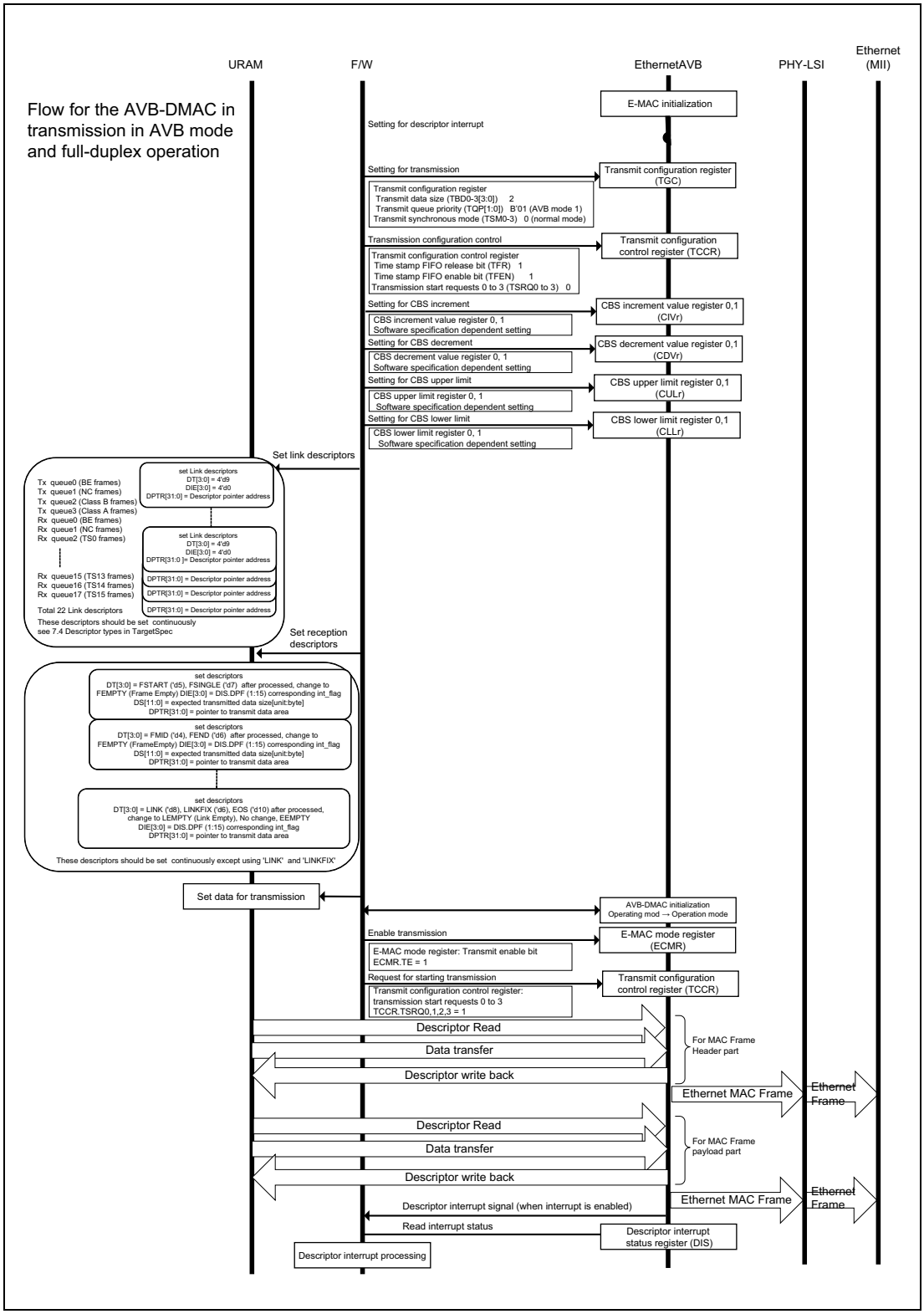


Figure 24.52 Flow for the AVB-DMAC in Transmission (in AVB Mode and Full-Duplex Operation)

24.4.12.5 Flow for Stopping AVB-DMAC Operation in Reception

Figure 24.53 shows the flow for stopping AVB-DMAC operation in reception (normal, common to all modes).

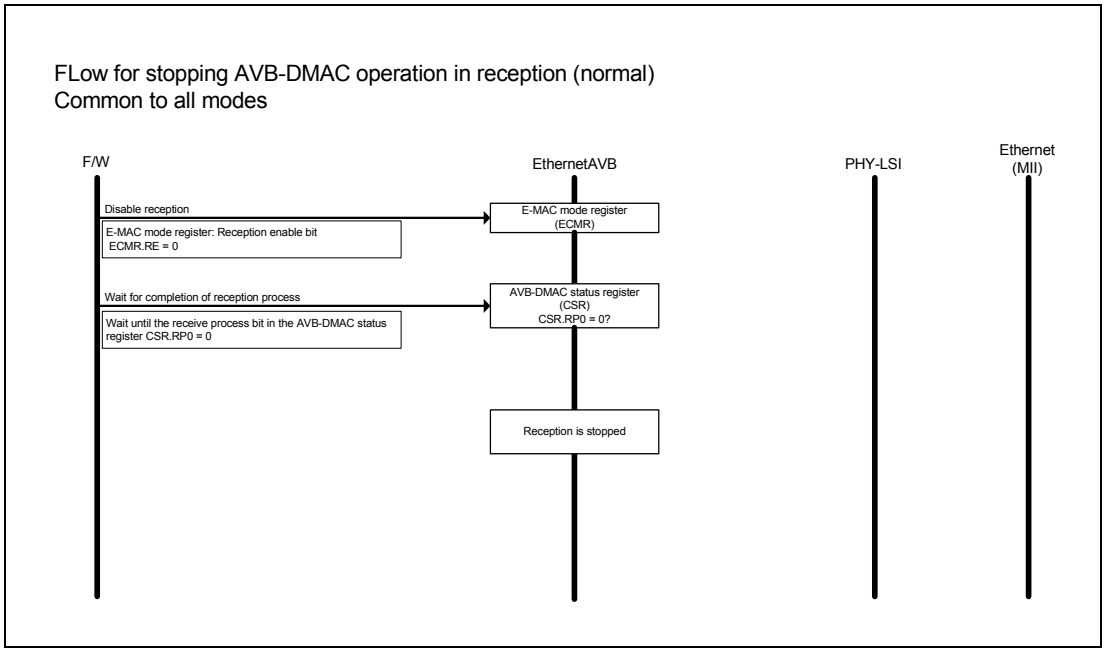


Figure 24.53 Flow for Stopping AVB-DMAC Operation in Reception (Normal, Common to All Modes)

24.4.12.6 Flow for Stopping AVB-DMAC Operation in Transmission

Figure 24.54 shows the flow for stopping AVB-DMAC operation in transmission (normal, common to all modes).

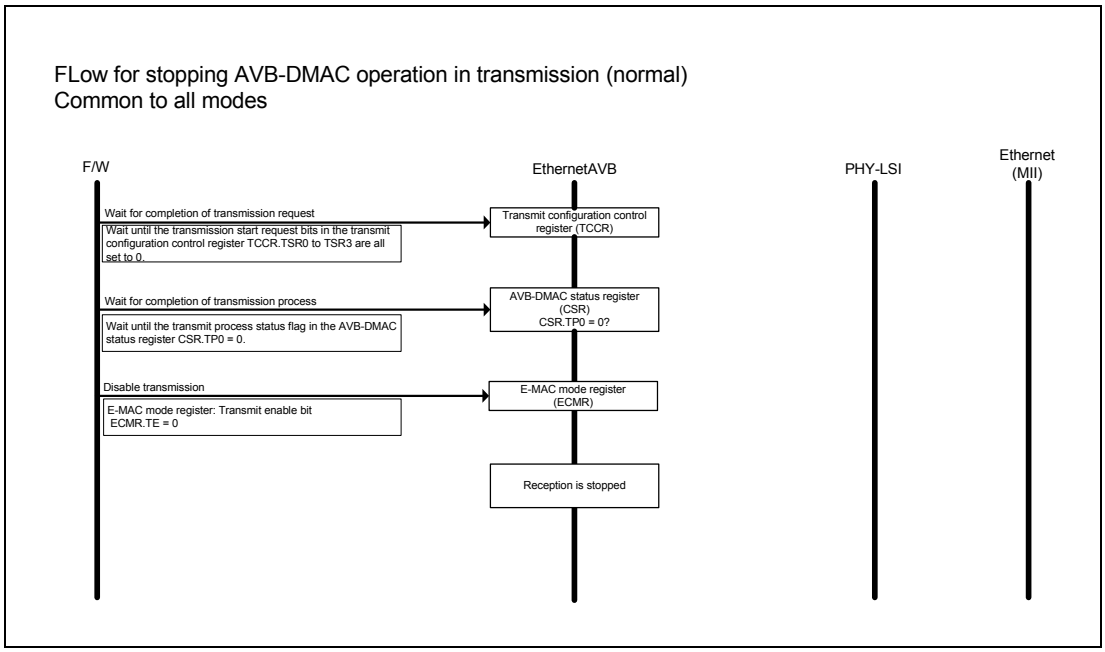


Figure 24.54 Flow for Stopping AVB-DMAC Operation in Transmission (Normal, Common to All Modes)

24.4.12.7 Flow for Stopping and Resetting the AVB-DMAC

Figure 24.55 shows the flow for stopping and resetting the AVB-DMAC (normal, common to all modes).

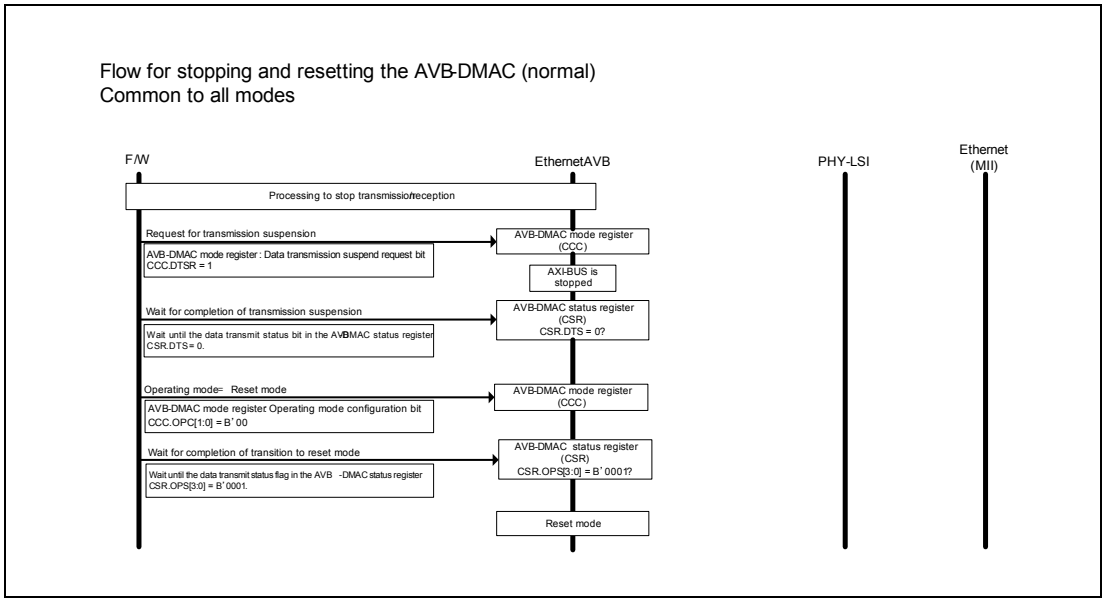


Figure 24.55 Flow for Stopping and Resetting the AVB-DMAC (Normal, Common to All Modes)

24.4.12.8 Flow for Emergency Stopping the AVB-DMAC

Figure 24.56 shows the flow for emergency stopping the AVB-DMAC (normal, common to all modes).

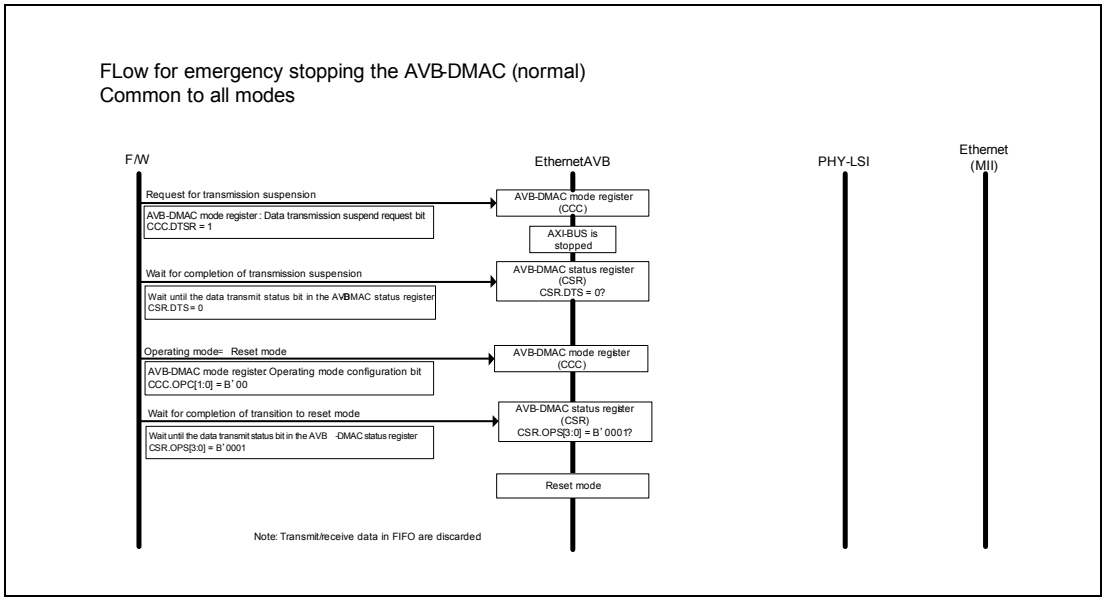


Figure 24.56 Flow for Emergency Stopping the AVB-DMAC (Normal, Common to All Modes)

24.4.12.9 Flow of gPTP Initialization

Figure 24.57 shows the flow of gPTP initialization (normal, common to all modes).

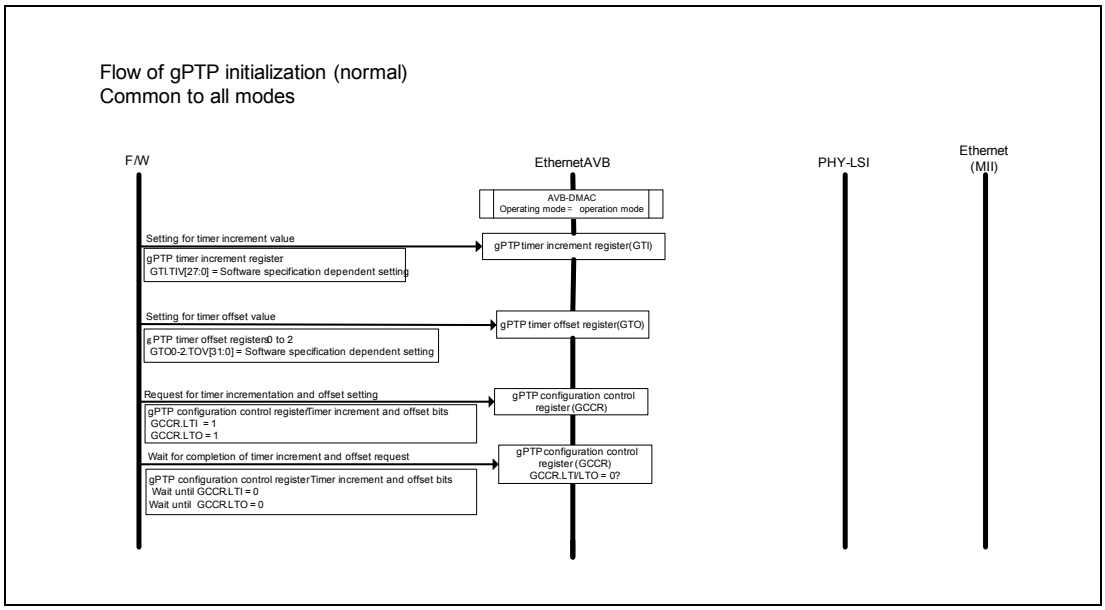


Figure 24.57 Flow of gPTP Initialization (Normal, Common to All Modes)

24.4.12.10 Flow of gPTP Time Stamping in Transmission

Figure 24.58 shows the flow of gPTP time stamping in transmission (normal, common to all modes).

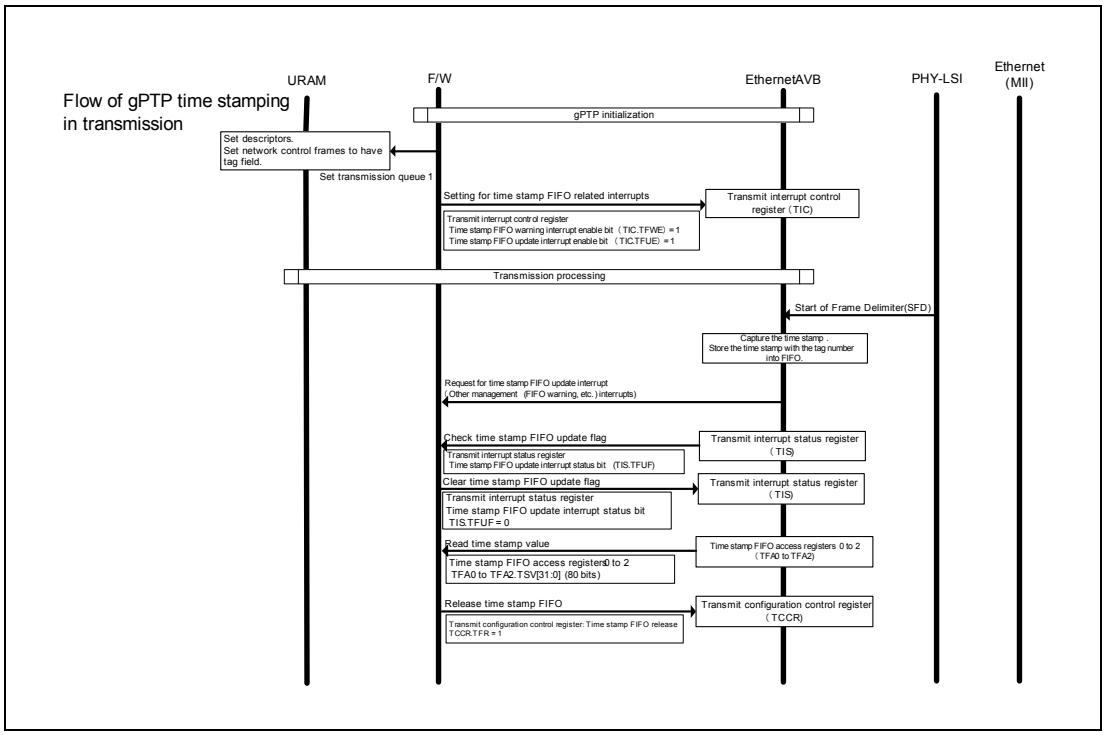


Figure 24.58 Flow of gPTP Time Stamping in Transmission (Normal, Common to All Modes)

24.4.12.11 Flow of gPTP Time Stamping and Synchronization in Reception

Figure 24.59 shows the flow of gPTP time stamping and synchronization in reception (normal, common to all modes).

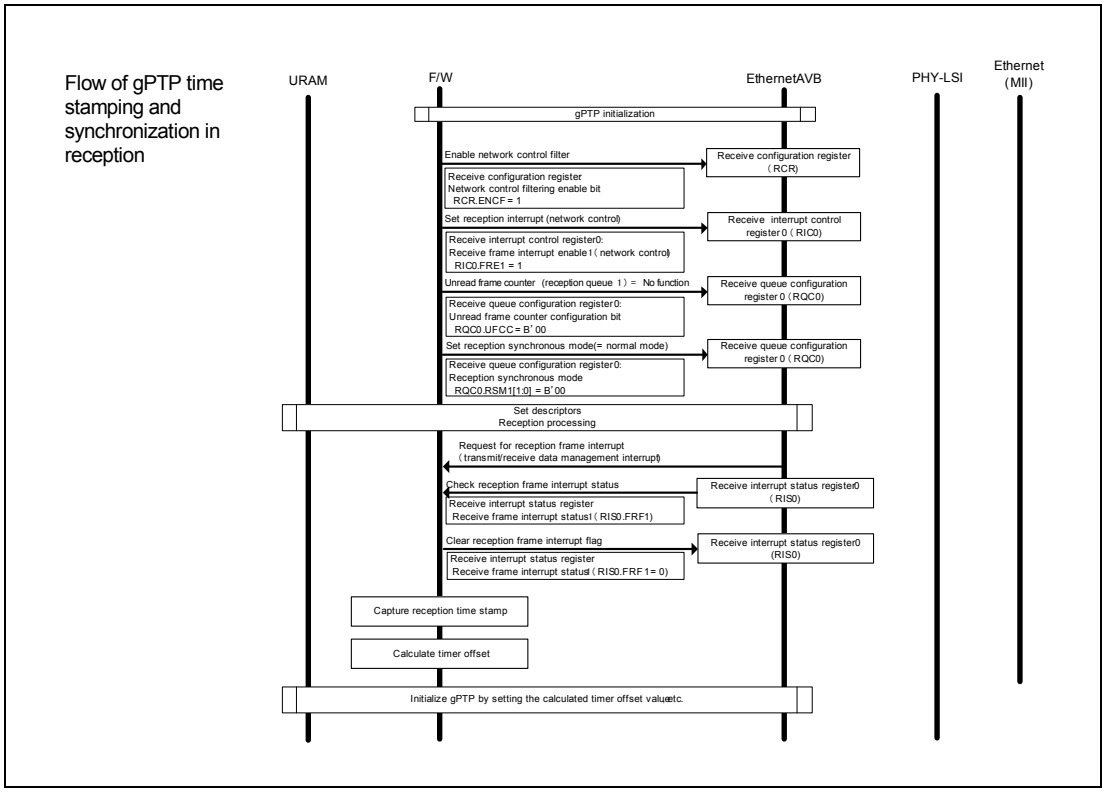


Figure 24.59 Flow of gPTP Time Stamping and Synchronization in Reception (Normal, Common to All Modes)

24.4.12.12 Flow of Capturing gPTP Presentation Times

Figure 24.60 shows the flow of capturing gPTP presentation times (common to all modes).

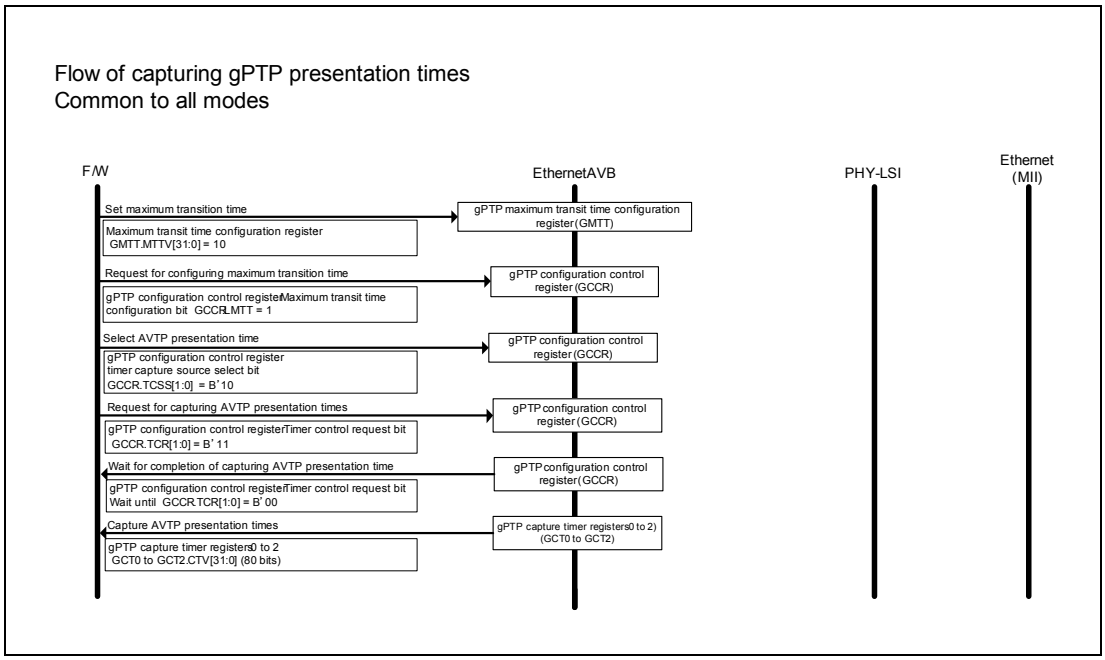


Figure 24.60 Flow of Capturing gPTP Presentation Times (Common to All Modes)

24.4.12.13 Flow of AVTP Presentation Time Comparison

Figure 24.61 shows the flow of AVTP presentation time comparison (common to all modes).

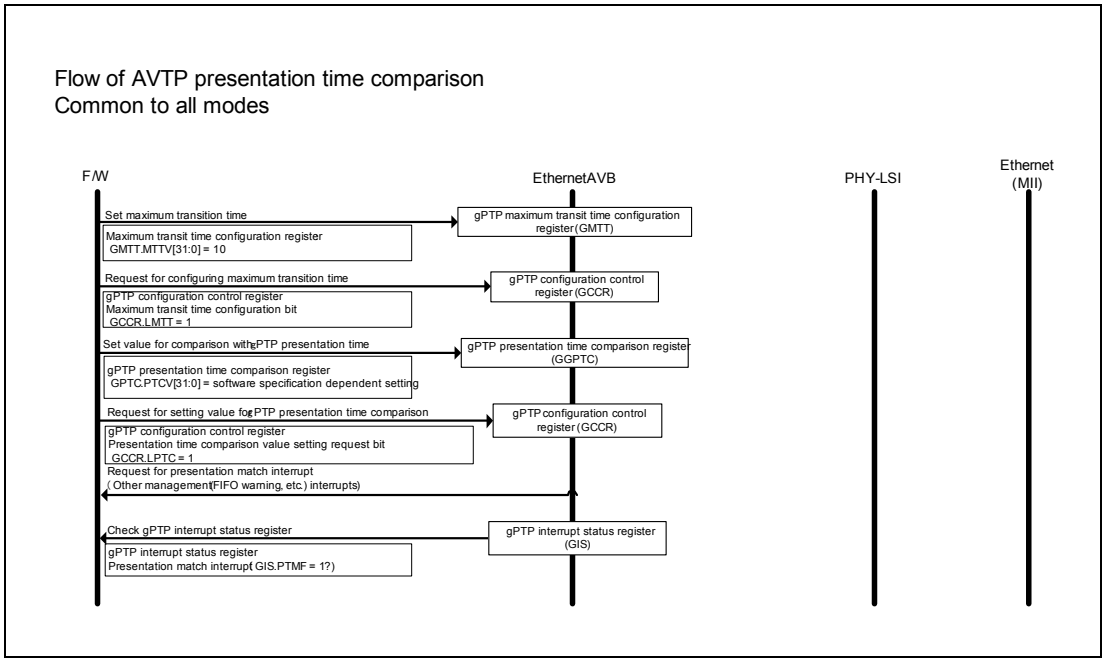


Figure 24.61 Flow of AVTP Presentation Time Comparison (Common to All Modes)

24.4.12.14 Flow of Loopback Mode Operation

Figure 24.62 shows the flow of loopback mode operation.

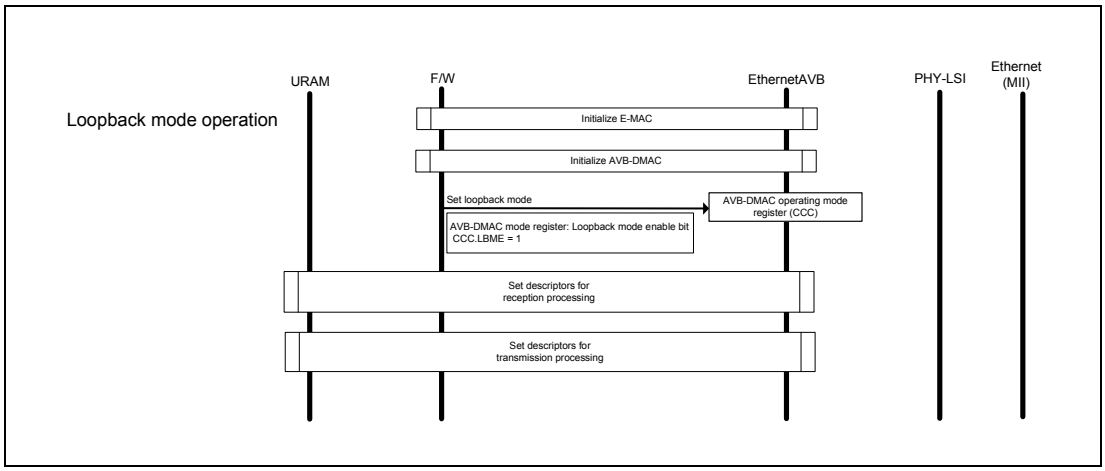


Figure 24.62 Flow of Loopback Mode Operation

24.4.13 Connection to PHY-LSI

24.4.13.1 MII Frame Transmission/Reception Timing

Each MII frame transmission/reception timing is shown in **Figure 24.63** to **Figure 24.68**.

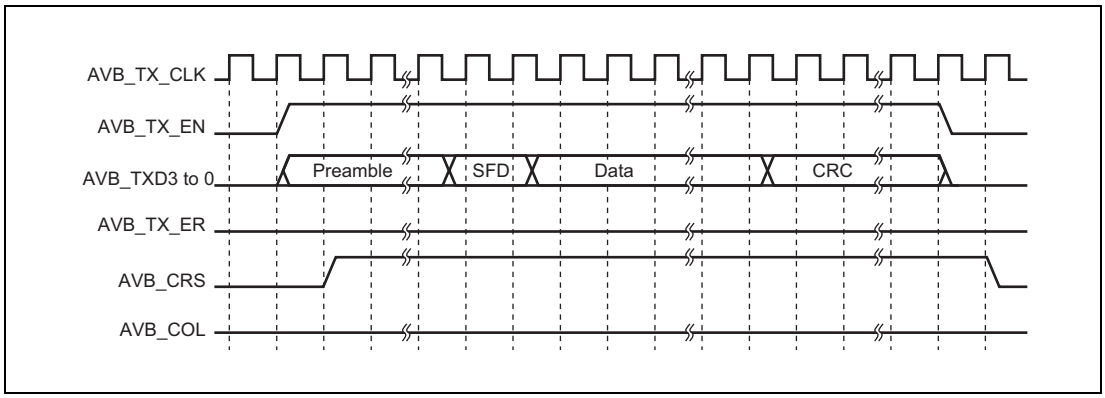


Figure 24.63 MII Frame Transmit Timing (Normal Transmission)

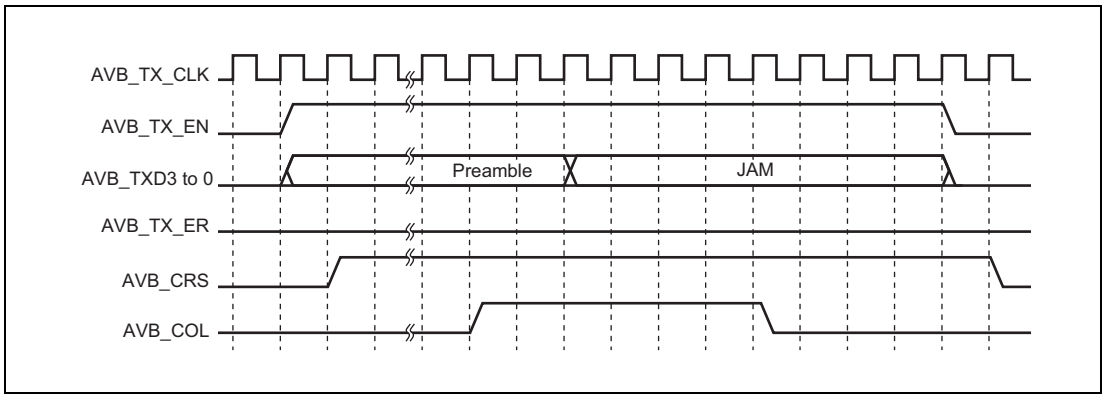


Figure 24.64 MII Frame Transmit Timing (Collision)

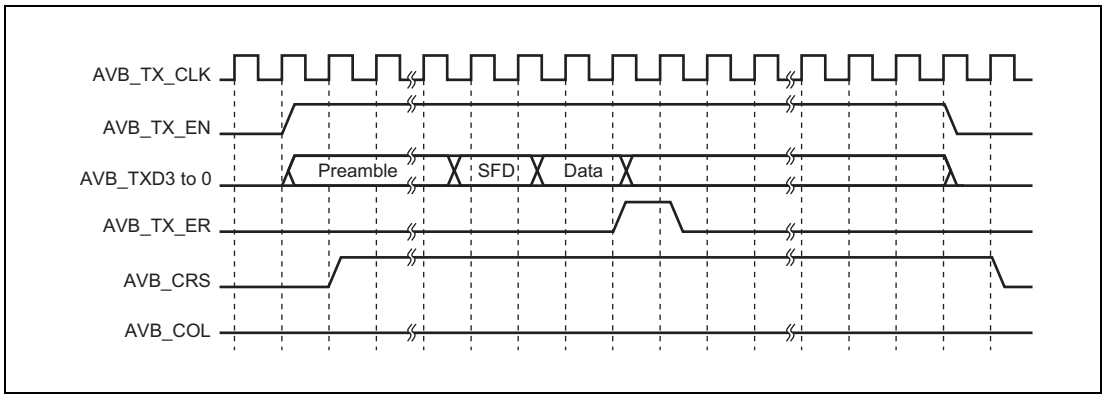


Figure 24.65 MII Frame Transmit Timing (Transmit Error)

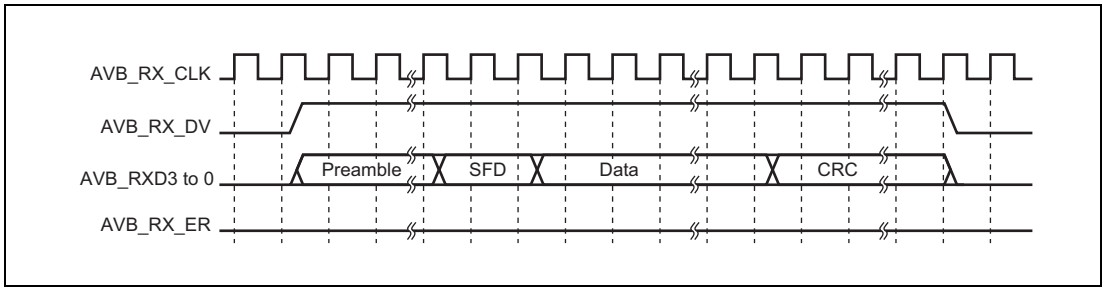


Figure 24.66 MII Frame Receive Timing (Normal Reception)

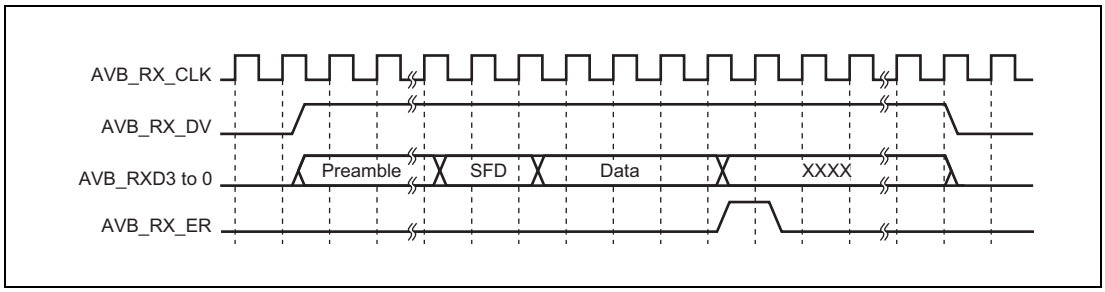


Figure 24.67 MII Frame Receive Timing (Reception Error (1))

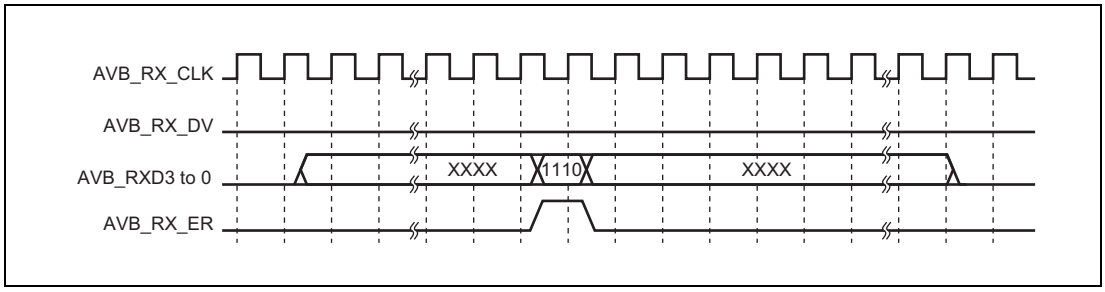


Figure 24.68 MII Fame Receive Timing (Reception Error (2))

24.4.13.2 MII Frame Reception Timing

Each MII frame reception timing is shown in **Figure 24.69** to **Figure 24.74**.

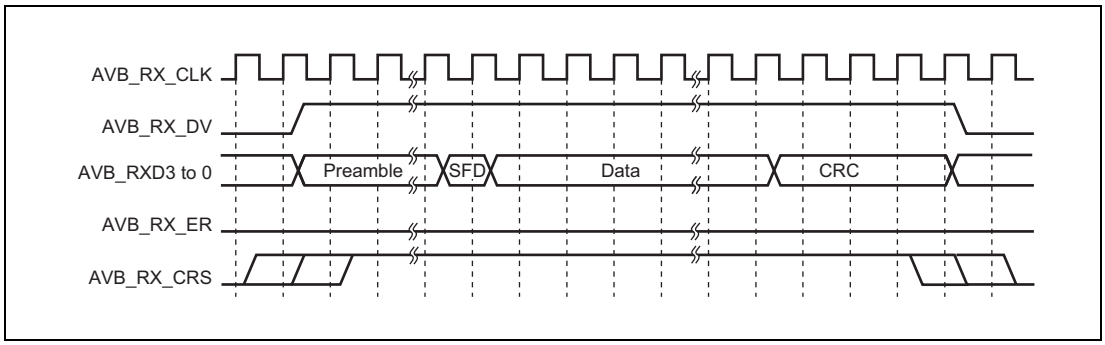


Figure 24.69 MII Fame Receive Timing (Normal Reception)

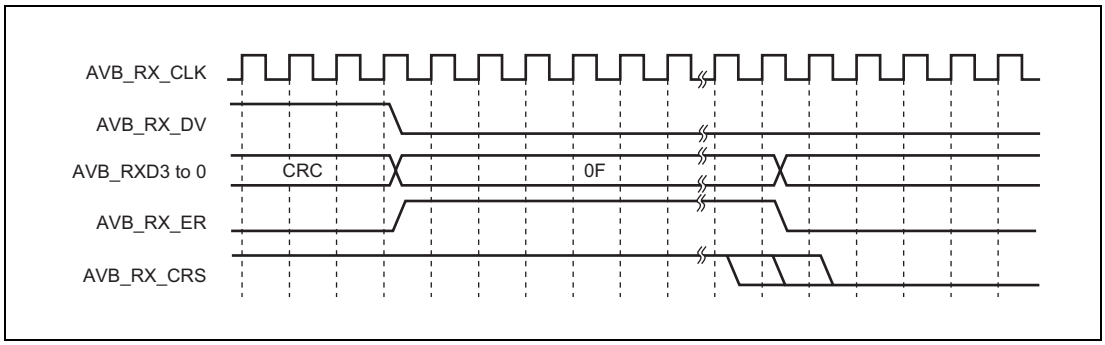


Figure 24.70 MII Fame Receive Timing (with Carrier Extension)

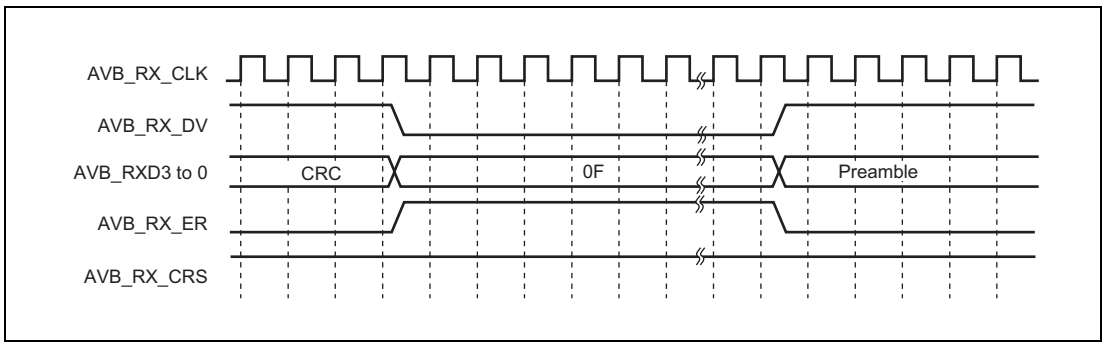


Figure 24.71 MII Fame Receive Timing (Burst Reception)

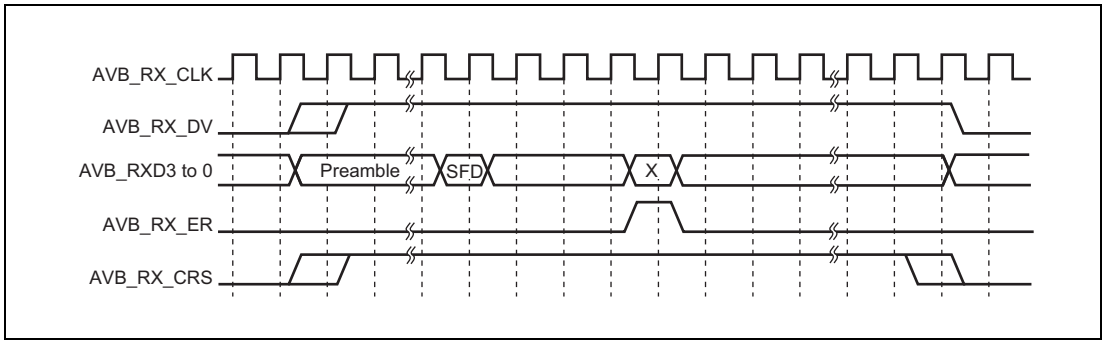


Figure 24.72 MII Fame Receive Timing (Reception Error)

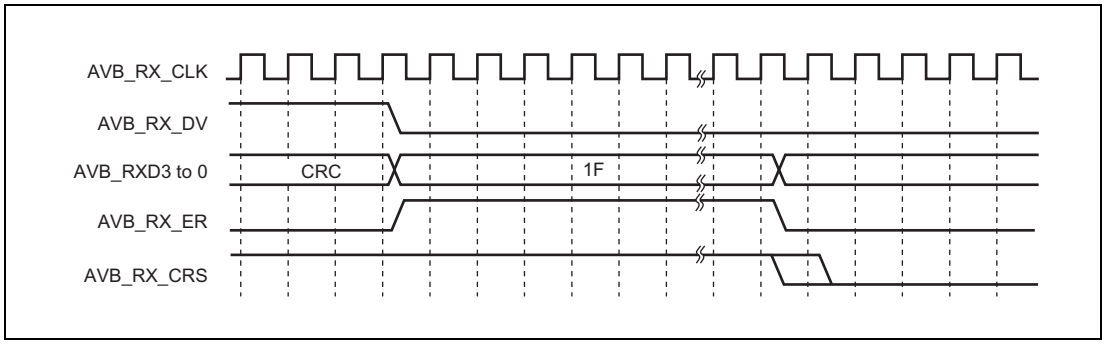


Figure 24.73 MII Fame Receive Timing (Error with Carrier Extension)

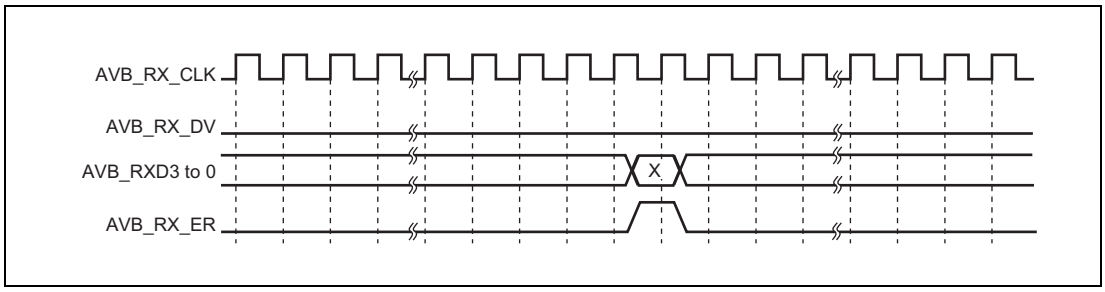


Figure 24.74 MII Fame Receive Timing (False Carrier Indication)

24.4.13.3 Accessing MII Registers

MII registers in the PHY-LSI are accessed via PIR in this LSI. PIR is used as a serial interface conforming to the MII frame format specified in IEEE802.3u.

(1) MII Management Frame Format

Figure 24.75 shows the format of an MII management frame. To access an MII register, a management frame is implemented by the program in accordance with the procedures shown in MII Register Access Procedure.

Access Type	MII Management Frame							
Item	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
Number of bits	32	2	2	5	5	2	16	
Read	1..1	01	10	00001	RRRRR	Z0	D..D	
Write	1..1	01	01	00001	RRRRR	10	D..D	X

[Legend]

PRE: 32 consecutive 1s
ST: Write of 01 indicating start of frame
OP: Write of code indicating access type
PHYAD: Write of 0001 if the PHY-LSI address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI address.
REGAD: Write of 000q if the register address is 1 (sequential write starting with the MSB).
This bit changes depending on the PHY-LSI register address.
TA: Time for switching data transmission source on MII interface
(a) Write: 10 written
(b) Read: Bus release (notation: Z0) performed
DATA: 16-bit data. Sequential write or read from MSB
(a) Write: 16-bit data write
(b) Read: 16-bit data read
IDLE: Wait time until next MII management format input
(a) Write: Independent bus release (notation: X) performed
(d) Read: Bus already released in TA: control unnecessary

Figure 24.75 MII Management Frame Format

(2) MII Register Access Procedure

The program accesses MII registers via PIR. Access is implemented by a combination of 1-bit-unit data write, 1-bit-unit data read, bus release, and independent bus release. **Figure 24.76** to **Figure 24.79** show the MII register access timing. The timing will differ depending on the PHY-LSI type.

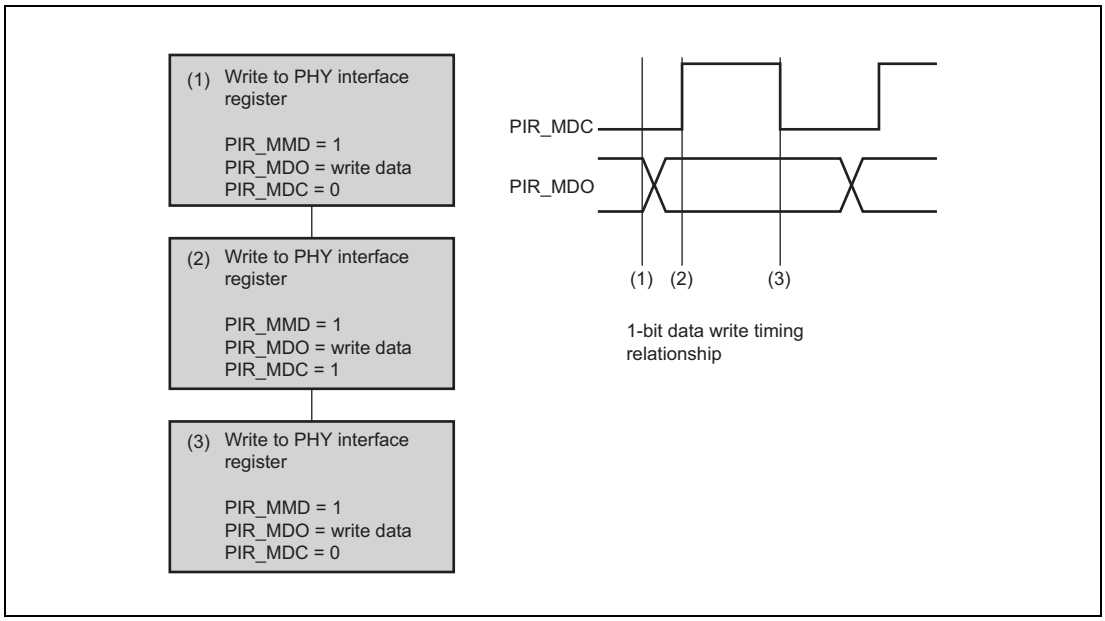


Figure 24.76 1-Bit Data Write Flowchart

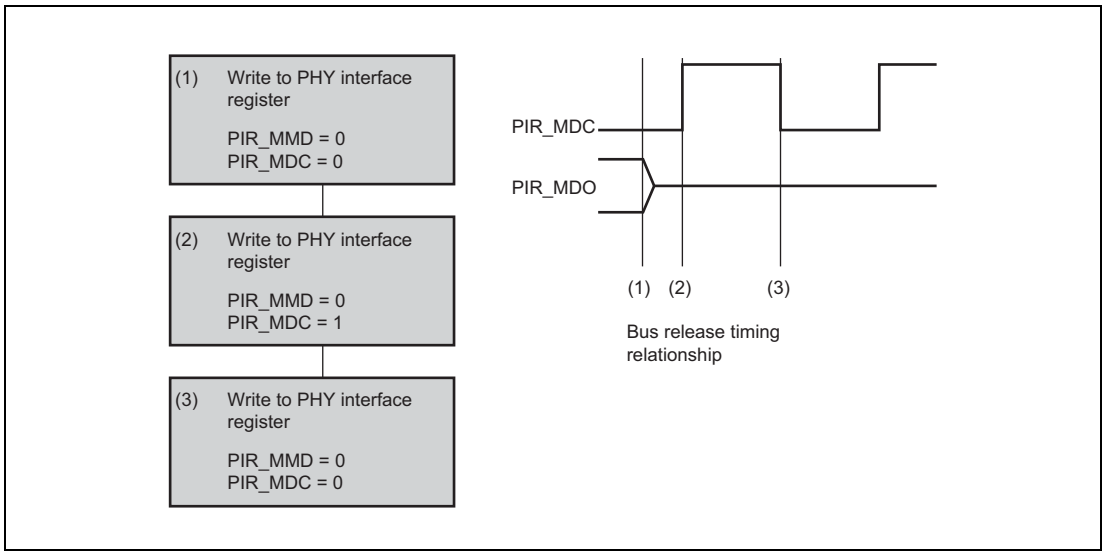


Figure 24.77 Bus Release Flowchart (TA in Read in Figure 24.75)

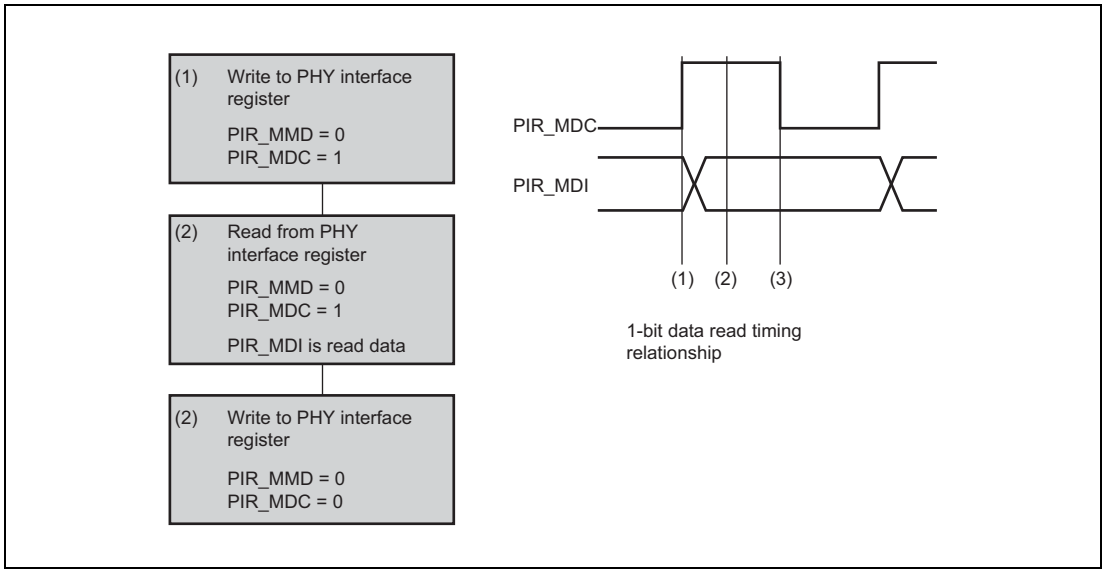


Figure 24.78 1-Bit Data Read Flowchart

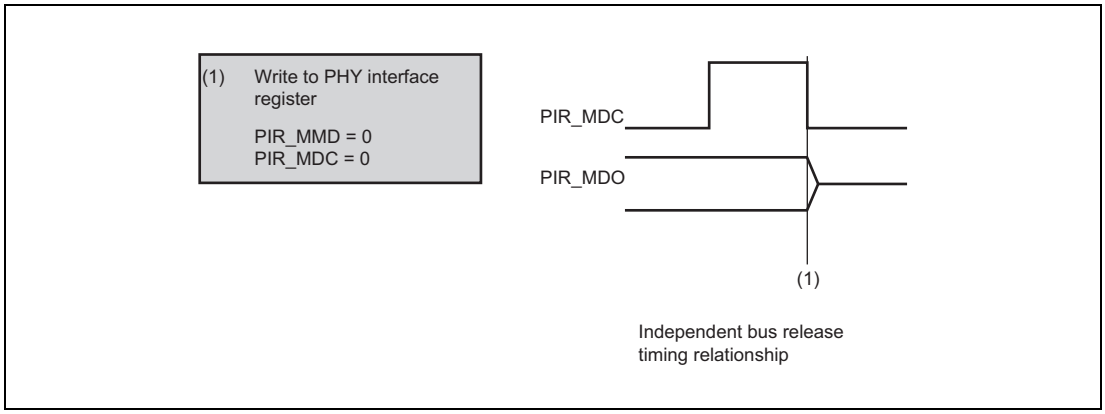


Figure 24.79 Independent Bus Release Flowchart (IDLE in Write in Figure 24.75)

24.4.14 Usage Notes

24.4.14.1 Checksum Calculation of Ethernet Frames

This LSI is capable of calculating the checksum data of the received frames. Only the data fields of the Ethernet frames are subject to checksum calculation. Specifically, a data field follows the length/type field and is followed by the CRC field. **Figure 24.80** shows schematics indicating which parts of the Ethernet frames are calculated. Calculation involves 16-bit addition only; it does not involve bit inversion. Note that when the checksum data is valid, the CRC data (4 bytes) is not transferred as a receive frame, and the checksum data (sum data) is added automatically. **Figure 24.81** shows schematics of Ethernet frames to which the checksum data has been added.

CAUTION

Also for the frames with VLANtag inserted, the 15th byte from the top and the following bytes before the CRC field are subject to calculation.

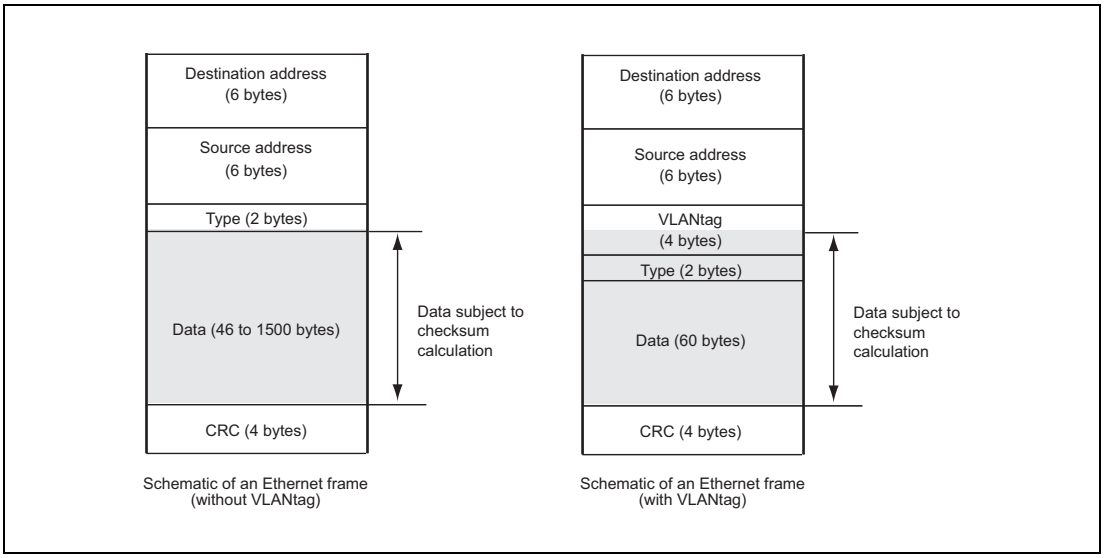


Figure 24.80 Data Subject to Checksum Calculation

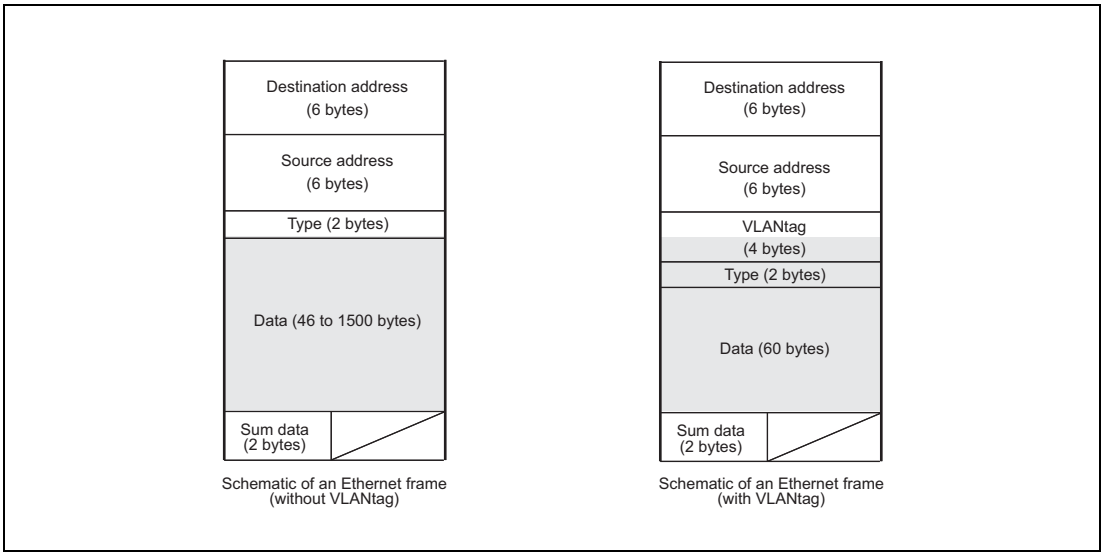


Figure 24.81 Data after Checksum Data Addition

24.4.14.2 Notes on Using the Intelligent Checksum Function

Checksum calculation using the intelligent checksum function is not affected by padding insertion specified by the receive data padding insert register (RPADIR). This is because checksum calculation is performed when transferring receive data from E-MAC to AVB-DMAC, while padding of receive data is performed when transferring receive data from AVB-DMAC to the receive buffer in memory.

24.4.14.3 Rx-FIFO read error may not be flagged when using FEMPTY_ND descriptor

After reading last byte targeted to FEMPTY_ND descriptor, there can be up to 3 bytes read from Rx-FIFO targeting next descriptor of same frame. Rx-FIFO read errors of not stored data due to FEMPTY_ND are neither flagged in DESCR.EI nor by EIS.QEF as per target specification.

When external ECC/parity logic flags Rx-FIFO read error related to last byte targeted to FEMPTY_ND descriptor, data in next descriptor may be corrupted without notification.

The issue is limited to implementations where Rx-FIFO error information is provided to ETNB and to applications using FEMPTY_ND descriptor.

The issue only occurs when the first byte saved in descriptor following FEMPTY_ND is not a multiple of 4 bytes inside the received frame.

Use a FEMPTY descriptor to store unwanted data if consistent error flagging is required.

24.4.14.4 When trying to release non-existing timestamp FIFO entry, new FIFO update flag may be lost

When SW releases an entry of timestamp FIFO by writing TCCR.TFR while this FIFO is empty (TSR.TFFL[2:0] is 000_B), flagging of next FIFO update may be inconsistent. The next timestamp is correctly stored in FIFO and fill level is incremented to 1 but TIS.TFUF is not set to 1.

The issue is limited to applications releasing FIFO entries without checking if there are entries available.

Do not release not existing FIFO entries.

24.4.14.5 gPTP compare may fail for range of compare values

When the comparison value (GPTC.PTCV[31:0]) is in range of [x-1 to x+1] (x is the configured increment value in GTI.TIV[27:0]), it may happen that a comparison match is not detected when Timer wraps around.

The issue is limited to applications using AVPT comparison function.

Do not configure comparison values inside the critical range.

24.4.14.6 UFC stop level triggers RIS2.QFFr even no received frame is lost

When a received frame is dropped due to non-availability of empty descriptor, RIS2.QFFr is correctly set to 1.

Additionally, when UFCVi.CVr reached the configured stop level (UFCS.SLj), the queue full flag (RIS2.QFFr) is set to 1 even before any further received frame is dropped.

The issue is limited to applications using unread frame counter with stop level function.

Such application gets information about lost received frames which in fact might not have lost.

24.4.14.7 RIS0.FRFr may lost when data processing stops close below configured warning level

When SW decrements UFC counter value (CV) to WL-1, resulting CV can still be WL due to recent new storage completion, RIS0.FRFr is set to 0.

Any further reception will set RIS0.FRFr as normal.

The issue is limited to applications using a single frame interrupt triggered by RIS0.FRFr and not be able to process each interrupt in time.

The missing interrupt due to this effect is automatically recovered by next frame reception as usual.

24.4.14.8 Receive frame interrupt and descriptor interrupt may be issued before completion of writing data

When receive frame interrupt is issued, the software should check that the descriptor type is updated correctly before processing the frame data.

The descriptor type is shown in the descriptor field DESCR.DT (the address of the current descriptor is shown by the register CDARq). If DESCR.DT is not yet updated by the DMA hardware (e.g. from FEMPTY to FEND), then the write of data to the memory is not completed. In that case the software should repeat checking the descriptor type until an update has happened.

Alternatively the software can use the Unread Frame Counter (UFC) value to compare with its processed descriptors to confirm there is no missing descriptor from the last received interrupt.

Section 25 Media Local Bus Interface (MLBB)

This section contains the block diagram and a list of all MLBB registers.
For detailed information about the functionality and the registers, refer to the document:

"OS62420 Media LB Device Interface Macro Advanced Product Data Sheet"

Document number OS62420AP2

OASIS SiliconSystems

The RH850/D1L/D1M implements the V1.4 of the Media Local Bus module.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.

25.1 Overview of RH850/D1L/D1M Media Local Bus (MLBB)

25.1.1 Units

This microcontroller has the following number of units of the Media Local Bus (MLBB).

Table 25.1 Units

Media Local Bus (MLBB)	D1L1	D1L2(H)	D1M1(H) D1M1-V2 D1M1A	D1M2	D1M2H
Units	0	0	0	0	1
Names	–	–	–	–	MLBB0

Units index n

Throughout this section, the individual units of a Medial Local Bus are identified by the index "n" (n = 0), for example, MLBBnMLBC0 for the MLBBn control 0 register.

25.1.2 Buffer RAM

The Media Local Bus Interface is equipped with following RAMs:

- channel table RAM: 128 x 144 bit
- MLBB data buffer RAM: 8 x 16 Kbit

25.1.3 Register addresses

All Media Local Bus register addresses are given as address offsets from the individual base addresses <MLBBn_base>.
The <MLBBn_base> addresses of each MLBBn are listed in the following table:

Table 25.2 Register base addresses <MLBBn_base>

MLBBn unit	<MLBBn_base> address
MLBB0	FFDE 1000_H

25.1.4 Clock supply

All Media Local Bus Interfaces provide two clock inputs.

Table 25.3 Clock supply

MLBBn unit	MLBBn clock	Connected to
MLBB0	sys_clk	• Clock Controller C_ISO_MLBB
	HCLK	• Clock Controller MLBBXCCLK
	PCLK	• Clock Controller MLBBPCLK

25.1.5 Interrupts

The Media Local Bus Interface can generate the following interrupt requests:

Table 25.4 MLBBn interrupt requests

MLBBn signals	Function	Connected to
MLBB0:		
mlb_int	MediaLB system interrupt	Interrupt Controller INTMLBB0SYS
ahb_int[0]	HBUS error interrupts	Interrupt Controller INTMLBB0STA0
ahb_int[1]		Interrupt Controller INTMLBB0STA1

25.1.6 Reset sources

The Media Local Bus Interfaces and their registers are initialized by the following reset signal:

Table 25.5 Reset sources

MLBBn unit	Reset signal
MLBB0	• Reset Controller SYSRES • Reset Controller MLBB0RES • Reset upon wake-up from DEEPSTOP mode

CAUTION

By default the MLBB0RES reset is active.
Thus before accessing this module this reset must be released via the MRSTC register of the Reset Controller. Refer to Section 9, Reset Controller for details.

25.1.7 I/O signals

The following table shows the I/O signals of the Media Local Bus Interface.

Table 25.6 MLBBn I/O signals

MLBBn signals	Function	Connected to
MLBB0		
MLBCLK_IN	Clock input	Port MLBB0CLK
MLBDAT_IO	Data input/output	Port MLBB0DAT
MLBSIG_IO	Signal input/output	Port MLBB0SIG

25.1.8 Bus master ID

The Media Local Bus I/F bus master interface is connected to the XC1 cross-connect system. The master interface has the following master ID:

MLBB0: MSTID7

25.2 Functional Overview

In the following only the block diagram and a list of registers is given.

For detailed information about the functionality and the registers, refer to the document:

"OS62420 Media LB Device Interface Macro Advanced Product Data Sheet"

Document number OS62420AP2

OASIS SiliconSystems

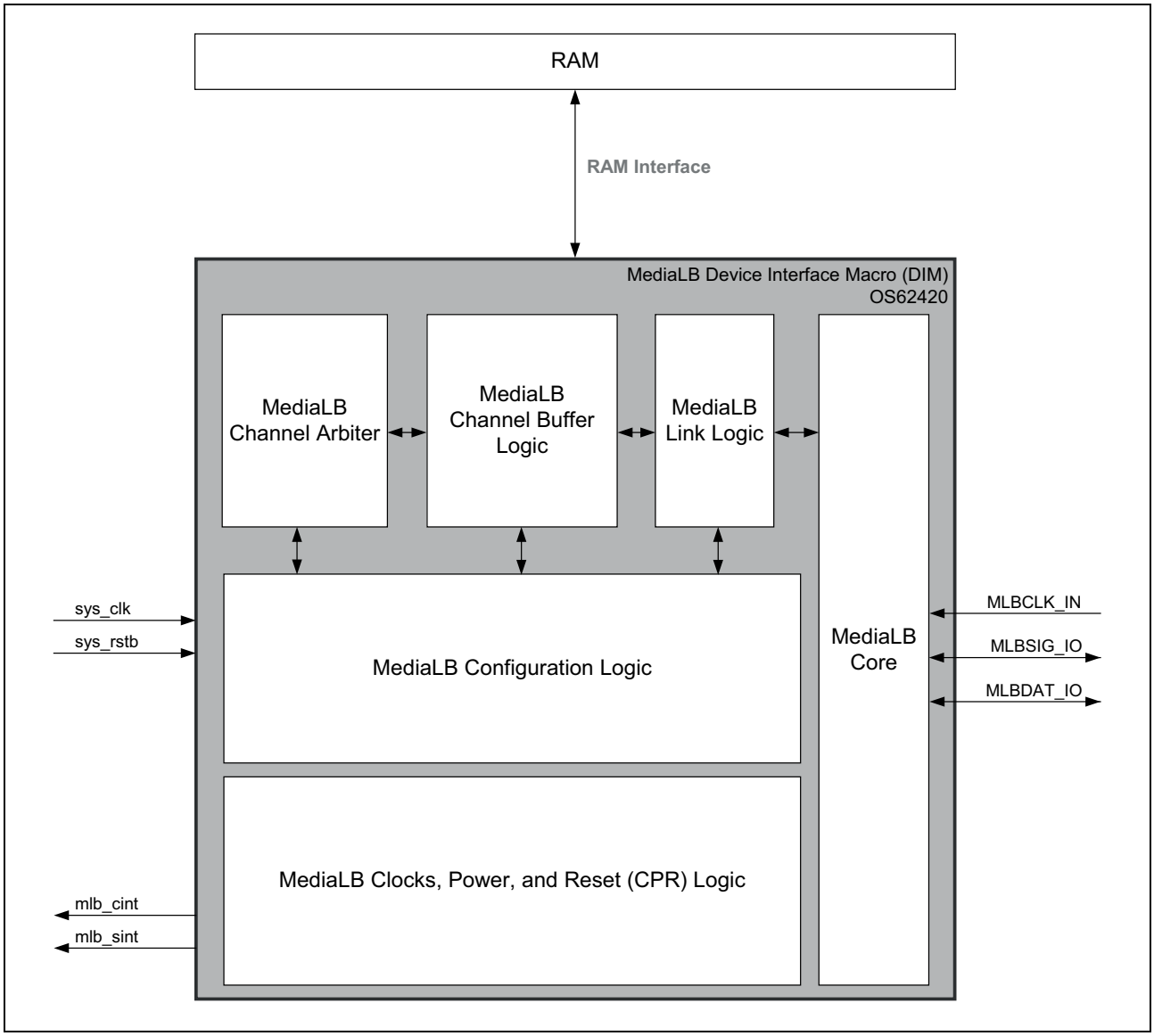


Figure 25.1 Block diagram of the Media Local Bus Interface

25.3 MLBn registers overview

This section contains a list of all registers of the Media Local Bus.

The Media Local Bus is controlled and operated by the following registers:

Table 25.7 MLBBn register overview

Module Name	Register name	Symbol	Address
MLBBn	MLBC0 MediaLB Control 0 Register	MLBC0	<MLBBn_base> + 000
MLBBn	MLBPC0 MediaLB 6-pin Control 0 Register	MLBPC0	<MLBBn_base> + 008
MLBBn	MS0 MediaLB Channel Status 0 Register	MS0	<MLBBn_base> + 00C
MLBBn	MS1 MediaLB Channel Status 1 Register	MS1	<MLBBn_base> + 014
MLBBn	MSS MediaLB System Status Register	MSS	<MLBBn_base> + 020
MLBBn	MSD MediaLB System Data Register (Read-	MSD	<MLBBn_base> + 024
MLBBn	MIEN MediaLB Interrupt Enable Register	MIEN	<MLBBn_base> + 02C
MLBBn	MLBPC2 MediaLB 6-pin Control 2 Register	MLBPC2	<MLBBn_base> + 034
MLBBn	MLBPC1 MediaLB 6-pin Control 1 Register	MLBPC1	<MLBBn_base> + 038
MLBBn	MLBC1 MediaLB Control 1 Register	MLBC1	<MLBBn_base> + 03C
MLBBn	HCTL HBI Control Register	HCTL	<MLBBn_base> + 080
MLBBn	HCMR0 HBI Channel Mask 0 Register	HCMR0	<MLBBn_base> + 088
MLBBn	HCMR1 HBI Channel Mask 1 Register	HCMR1	<MLBBn_base> + 08C
MLBBn	HCER0 HBI Channel Error 0 Register	HCER0	<MLBBn_base> + 090
MLBBn	HCER1 HBI Channel Error 1 Register	HCER1	<MLBBn_base> + 094
MLBBn	HCBR0 HBI Channel Busy 0 Register	HCBR0	<MLBBn_base> + 098
MLBBn	HCBR1 HBI Channel Busy 1 Register	HCBR1	<MLBBn_base> + 09C
MLBBn	MDAT0 MIF Data 0 Register	MDAT0	<MLBBn_base> + 0C0
MLBBn	MDAT1 MIF Data 1 Register	MDAT1	<MLBBn_base> + 0C4
MLBBn	MDAT2 MIF Data 2 Register	MDAT2	<MLBBn_base> + 0C8
MLBBn	MDAT3 MIF Data 3 Register	MDAT3	<MLBBn_base> + 0CC
MLBBn	MDWE0 MIF Data Write Enable 0 Register	MDWE0	<MLBBn_base> + 0D0
MLBBn	MDWE1 MIF Data Write Enable 1 Register	MDWE1	<MLBBn_base> + 0D4
MLBBn	MDWE2 MIF Data Write Enable 2 Register	MDWE2	<MLBBn_base> + 0D8
MLBBn	MDWE3 MIF Data Write Enable 3 Register	MDWE3	<MLBBn_base> + 0DC
MLBBn	MCTL MIF Control Register	MCTL	<MLBBn_base> + 0E0
MLBBn	MADR MIF Address Register	MADR	<MLBBn_base> + 0E4
MLBBn	ACTL AHB Control Register	ACTL	<MLBBn_base> + 3C0
MLBBn	ACSR0 AHB Channel Status 0 Register	ACSR0	<MLBBn_base> + 3D0
MLBBn	ACSR1 AHB Channel Status 1 Register	ACSR1	<MLBBn_base> + 3D4
MLBBn	ACMR0 AHB Channel Mask 0 Register	ACMR0	<MLBBn_base> + 3D8
MLBBn	ACMR1 AHB Channel Mask 1 Register	ACMR1	<MLBBn_base> + 3DC

<MLBBn_base>

The base addresses <MLBBn_base> of the MLBBn is defined in the first section of this chapter under the key word “Register addresses”.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

Section 26 Window Watchdog Timer (WDTA)

This section contains a generic description of the Window Watchdog Timer (WDTA).

The first part of this section describes all RH850/D1L/D1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of WDTA.

26.1 Features of RH850/D1L/D1M WDTA

26.1.1 Number of Units and Channels

This microcontroller has the following number of WDTA units.

Table 26.1 Number of Units

Product Name	All products
Number of Units	2
Name	WDTAn (n = 0, 1)

Table 26.2 Index

Index	Meaning
n	Throughout this section, the individual window watchdog timer units are identified by the index “n”: for example, WDTAnWDTE (n = 0, 1) is the WDTAn enable register.

26.1.2 Register Base Addresses

WDTAn base addresses are listed in the following table.

WDTAn register addresses are given as offsets from the base addresses in general.

Table 26.3 Register Base Addresses

Base Address Name	Base Address
<WDTA0_base>	FFED 0000_H
<WDTA1_base>	FFED 1000_H

26.1.3 Clock Supply

The WDTAn clock supply is shown in the following table.

Table 26.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
WDTA0	PCLK	Clock Controller • C_ISO_PCLK in RUN mode • EMCLK in DEEPSTOP mode
	WDTATCKI	Clock Controller C_AWO_WDTA0
WDTA1	PCLK	Clock Controller C_ISO_PCLK
	WDTATCKI	Clock Controller $f_{RL}/4$

26.1.4 Interrupt Request and Error Signals

WDTAn interrupt requests and error signals are listed in the following table.

Table 26.5 WDTAn interrupt requests and error signals

WDTAn signals	Function	Connected to
WDTA0:		
WDTA0TRES	WDTA0 error reset	Reset Controller WDTA0RES
WDTA0TNMI	WDTA0 error NMI	Error Control Module INTWDTA0NMI
INTWDTA0	WDTA0 75% interrupt	Interrupt Controller INTWDTA0
WDTA1:		
WDTA1TRES	WDTA1 error reset	Reset Controller WDTA1RES
WDTA1TNMI	WDTA1 error NMI	Error Control Module INTWDTA1NMI
INTWDTA1	WDTA1 75% interrupt	Interrupt Controller INTWDTA1

26.1.5 Reset Sources

WDTAn reset sources are listed in the following table. WDTAn is initialized by these reset sources.

Table 26.6 Reset Sources

Unit Name	Reset Source
WDTA0	• Reset Controller SYSRES
WDTA1	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

26.1.6 WDTATRTYP

The connection of the WDTA reset type input signals are listed in the table below.

Table 26.7 WDTARTYPE connections

WDTAn unit	WDTAn signals	Connected to
WDTA0	WDTATRTYP	Reset Controller WDTATRTYP
WDTA1		

CAUTION

Confliction of resets

A reset conflict may occur if a reset is asserted before another already active reset is released.

Table 26.8 Conflict factors

		Second asserted			
		PURES, POC1RES		Any except PURES, POC1RES	
		WDTATRTYP	Start mode	WDTATRTYP	Start mode
First asserted	PURES, POC1RES	0	Software trigger	0	Software trigger
	Any except PURES, POC1RES	0	Software trigger*1	1	Default (except ISORES)

Note 1. – While $\overline{\text{RESET}}$, $\overline{\text{CLMA0RES}}$, WDTA0RES, WDTA1RES, SWRES, ECMRES, ISORES is asserted, POC0RES is asserted and released within 15 µs:
WDTATRTYP = 1, start mode is default
– While $\overline{\text{RESET}}$, $\overline{\text{CLMA0RES}}$, WDTA0RES, WDTA1RES, SWRES, ECMRES, ISORES asserted, POC0RES is asserted and released after 15 µs:
WDTATRTYP = 0, start mode is Software trigger

26.2 Overview

26.2.1 Functional Overview

WDTA has the following functions:

- Selection of the operation mode after reset, by using the option bytes

Enabling/disabling of WDTA, starting/stopping of the counter after reset, setting of the counter overflow time, and enabling/disabling of the VAC function can be selected. WDTA startup options to be set by the option bytes are described in **Table 26.9**.

- WDTA trigger function

Writing an activation code to the WDTA trigger register starts WDTA and restarts the counter. Activation codes include fixed activation codes and variable activation codes (VAC function). In a variable activation code, a different value from the previous time (variable value) is written to the WDTA trigger register, which causes the counter to be restarted.

- Interrupt request generation at 75% of the counter overflow value

An interrupt request signal can be generated when the WDTA counter reaches 75% of the overflow interval time (this function can be enabled or disabled by the setting of WDTAnMD.WDTAnWIE).

- Window function

The period during which writing to the WDTA trigger register is valid (window-open period) can be set. Writing to the WDTA trigger register at a time outside the window-open period causes an error.

- WDTA error detection function

When an error is detected, a non-maskable interrupt request or an internal reset is generated.

For details about the error sources, see Section 26.5.3, WDTA Error Detection.

NOTE

The WDTA0 (operating on the AWO area) continues its operation also in the DEEPSTOP mode.
There is no configuration option to change this behaviour.

Table 26.9 WDTA start-up options

Start-up option	Function	Description	Connected to
OPWDEN	WDTA enable/disable	Enables/disables the WDTA: 0: WDTA is disabled 1: WDTA is enabled	• WDTA0: flash option OPBT0.WDT0OPWDEN • WDTA1: flash option OPBT0.WDT1OPWDEN
OPWDOVF[2:0]	Overflow interval time reset value setting	Specifies the reset value of the overflow interval time control bits WDTAnMD.WDTAnOVF[2:0].	• WDTA0: flash option OPBT0.WDT0OPWDOVF[2:0] • WDTA1: flash option OPBT0.WDT1OPWDOVF[2:0]
OPWDTPR	Start mode signal selector	Specifies the signal that sets the start mode: 0: OPWDRUN start-up option 1: WDTATRTYP input signal If WDTATRTYP is selected (OPWDTPR = 1), the start mode depends on the reset type. Refer to Section 26.5.1, WDTA after Reset Release for details.	• WDTA0: flash option OPBT0.WDT0OPWDTPR • WDTA1: flash option OPBT0.WDT1OPWDTPR
OPWDRUN	Start mode setting	Specifies the start mode: 0: Software trigger start mode 1: Default start mode Refer to Section 26.5.1, WDTA after Reset Release for details.	• WDTA0: flash option OPBT0.WDT0OPWDRUN • WDTA1: flash option OPBT0.WDT1OPWDRUN
OPWDVAC	Variable Activation Code (VAC) selection	Specifies the trigger register for the generation of counter re-start triggers to keep the counter from overflowing. 0: WDTAnWDTE (fixed) 1: WDTAnEVAC (variable) When WDTAnWDTE is selected, the value to be written to the register (activation code) is fixed (AC_H). When WDTAnEVAC is selected, the activation code to be written to the register is variable. For details, refer to 26.5.2.1, Calculating an Activation Code when the VAC Function is Used.	• WDTA0: flash option OPBT0.WDT0OPWDVAC • WDTA1: flash option OPBT0.WDT1OPWDVAC

26.2.2 Block Diagram

Figure 26.1 shows the main components of the WDTA.

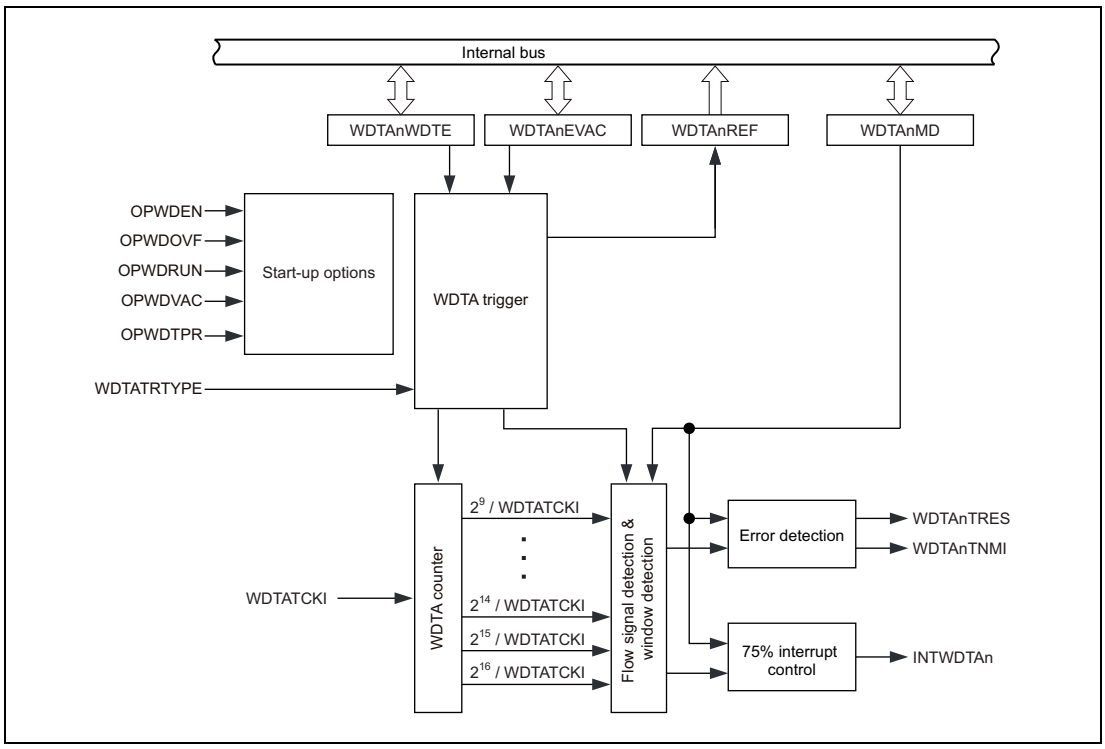


Figure 26.1 Block Diagram of the Window Watchdog Timer A

26.3 Registers

26.3.1 List of Registers

WDTAn registers are listed in the following table.

For details about <WDTAn_base>, see Section 26.1.2, Register Base Addresses.

Table 26.10 Registers

Module	Register	Symbol	Address
WDTAn	WDTA enable register	WDTAnWDTE	<WDTAn_base> + 0000_H
WDTAn	WDTA enable VAC register	WDTAnEVAC	<WDTAn_base> + 0004_H
WDTAn	WDTA reference value register	WDTAnREF	<WDTAn_base> + 0008_H
WDTAn	WDTA mode register	WDTAnMD	<WDTAn_base> + $000C_H$

26.3.2 WDTAnWDTE — WDTA Enable Register

This register is the WDTA trigger register when the VAC function is not used (start-up option OPWDVAC = 0).

Writing AC_H to this register generates a WDTA trigger and starts or restarts the WDTA counter. See Section 26.5.2, WDTA Trigger, for details.

The behavior of this register depends on the setting of the start-up option OPWDVAC, see Table 26.13, WDTAnWDTE Behavior.

Access: This register can be read/written in 8-bit units.

Address: <WDTAn_base> + 0000_H

Value after reset: The initial value depends on the start-up options OPWDEN, OPWDVAC, OPWDTPR, OPWDRUN, and the WDTATRTYP signal. Refer to Table 26.12, Values of WDTAnRUN7 after Reset.
WDTA0 is initialized by SYSRES.
WDTA1 is initialized by SYSRES and wake-up from DEEPSTOP mode.

Bit	7	6	5	4	3	2	1	0
	WDTAnRUN[7:0]							
Value after reset	0/1	0	1	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.11 WDTAnWDTE Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTAnRUN [7:0]	Writing the fixed activation code (AC_H) generates the WDTA trigger and starts/restarts the WDTAn counting. Writing the value other than AC_H generates an error. The WDTAn cannot be stopped once it was started. See **Table 26.13, WDTAnWDTE Behavior**, when reading from or writing to these bits.

The WDTAnRUN7 bit is only valid if WDTA is enabled (OPWDEN = 1) and the VAC function is disabled (OPWDVAC = 0). **Table 26.12** lists the values of the WDTAnRUN7 bit after reset according to the start-up options.

Table 26.12 Values of WDTAnRUN7 after Reset

Start-up options				Reset type signal WDTATRTYP	Start Mode	Value of WDTAnRUN7 after Reset
OPWDEN	OPWDVAC	OPWDTPR	OPWDRUN			
0	Ignored	Ignored	Ignored	Ignored	WDTA is disabled	0
1	1	Ignored	Ignored	Ignored	Not valid (VAC mode)	0
	0	0	0	Ignored	Software trigger start	0
			1		Default start	1
		1	0	Ignored	Software trigger start	0
			1	0	Software trigger start	0
				1	Default start	1

The behavior of WDTAnWDTE during read/write accesses depends on the OPWDVAC setting, as shown in Table 26.13, WDTAnWDTE Behavior.

Table 26.13 WDTAnWDTE Behavior

		WDTAnWDTE	
OPWDVAC	**Description**	**Read**	**Write**
0	The VAC function is disabled. WDTAnWDTE is enabled.	$2C_H$ is read (in software trigger start mode, before the activation of WDTAn). AC_H is read (after the activation of WDTAn).	WDTA trigger Write AC_H *1.
1	The VAC function is enabled. WDTAnWDTE is disabled.	$2C_H$ is read.	Writing is ignored.

Note 1. Any other write value will cause an error.

26.3.3 WDTAnEVAC — WDTA Enable VAC Register

This register is the WDTA trigger register when the VAC function is used (start-up option OPWDVAC = 1).

Writing a correct activation code to this register generates a WDTA trigger and starts or restarts the WDTA counter. For details, see Section 26.5.2, WDTA Trigger. For details about the activation codes when the VAC function is used, see Section 26.5.2.1, Calculating an Activation Code when the VAC Function is Used.

The behavior of this register depends on the setting of the start-up option OPWDVAC. See Table 26.16, WDTAnEVAC Behavior.

Access: This register can be read/written in 8-bit units.

Address: <WDTAn_base> + 0004_H

Value after reset: The initial value depends on the start-up options OPWDEN, OPWDVAC, OPWDTPR, OPWDRUN, and the WDTATRTYP signal. Refer to Table 26.15, Values of WDTAnEVAC7 after Reset.
WDTA0 is initialized by SYSRES.
WDTA1 is initialized by SYSRES and wake-up from DEEPSTOP mode.

Bit	7	6	5	4	3	2	1	0
	WDTAnEVAC[7:0]							
Value after reset	0/1	0	1	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 26.14 WDTAnEVAC Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTAnEVAC [7:0]	Writing an variable activation code generates the WDTA trigger and starts/restarts the WDTAn counting. Writing a wrong activation code generates an error. The WDTAn cannot be stopped once it was started. See Table 26.16, WDTAnEVAC Behavior, when reading from or writing to these bits.

The WDTAnEVAC7 bit is only valid if WDTA is enabled (OPWDEN = 1) and the VAC function is enabled (OPWDVAC = 1). **Table 26.15** lists the values of the WDTAnEVAC7 bit after reset according to the start-up options.

Table 26.15 Values of WDTAnEVAC7 after Reset

Start-up options				Reset type signal		Value of WDTAnEVAC7 after Reset
OPWDEN	OPWDVAC	OPWDTPR	OPWDRUN	WDTATRTYP	Start Mode	
0	Ignored	Ignored	Ignored	Ignored	WDTA is disabled	0
1	0	Ignored	Ignored	Ignored	Not valid (non VAC mode)	0
	1	0	0	Ignored	Software trigger start	0
			1		Default start	1
		1	0	Ignored	Software trigger start	0
			1	0	Software trigger start	0
				1	Default start	1

The behavior of WDTAnEVAC during read/write accesses depends on the OPWDVAC setting, as shown in **Table 26.16**.

Table 26.16 WDTAnEVAC Behavior

		WDTAnEVAC	
OPWDVAC	**Description**	**Read**	**Write**
0	The VAC function is disabled. WDTAnEVAC is disabled.	$2C_H$ is read.	Writing is ignored.
1	The VAC function is enabled. WDTAnEVAC is enabled.	$2C_H$ is read (in software trigger start mode, before the activation of WDTAn). The variable activation code written last is read (after the activation of WDTAn).	Write the variable activation code*1 For details, see **Section 26.5.2.1, Calculating an Activation Code when the VAC Function is Used**.

Note 1. Any other write value will cause an error.

26.3.4 WDTAnREF — WDTA Reference Value Register

This register contains the reference value for calculating the activation code of the VAC function. It is automatically updated after every trigger operation. See Section 26.5.2.1, Calculating an Activation Code when the VAC Function is Used.

If the VAC function is disabled (OPWDVAC = 0), reading this register returns 00_H.

Access: This register can only be read in 8-bit units.

Address: <WDTAn_base> + 0008_H

Value after reset: 00_H
WDTA0 is initialized by SYSRES.
WDTA1 is initialized by SYSRES and wake-up from DEEPSTOP mode.

Bit	7	6	5	4	3	2	1	0
	WDTAnREF[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 26.17 WDTAnREF Register Contents

Bit Position	Bit Name	Function
7 to 0	WDTAnREF[7:0]	Reference value for activation code calculation for the VAC function

26.3.5 WDTAnMD — WDTA Mode Register

This register specifies the overflow interval time, the 75% interrupt enable/disable, the error mode, and the window-open period.

The value of this register can be updated only once after reset release and before the first trigger is generated. The updated value will be effective after a maxium of 3 × Timer count clock cycles since the WDTA trigger register is written to.

Updating this register after the first WDTA trigger is generated causes an error, but an error does not occur if the same value has been written to it.

Access: This register can be read/written in 8-bit units.

Address: <WDTAn_base> + $000C_H$

Value after reset: The initial value depends on the start-up options OPWDOVF[2:0]. See Table 26.9, WDTA start-up options.
WDTA0 is initialized by SYSRES.
WDTA1 is initialized by SYSRES and wake-up from DEEPSTOP mode.

Bit	7	6	5	4	3	2	1	0
	—	WDTAnOVF[2:0]			WDTAnWIE	WDTAnERM	WDTAnWS[1:0]	
Value after reset	0	*1	*1	*1	1	1	1	1
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The WDTAnOVF[2:0] value after reset can be set by the start-up options OPWDOVF[2:0].

Table 26.18 WDTAnMD Register Contents (1/2)

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
6 to 4	WDTAnOVF[2:0]	Selects the overflow interval time:
3	WDTAnWIE	Enables/disables the 75% interrupt request INTWDTAn. 0: INTWDTAn disabled 1: INTWDTAn enabled
2	WDTAnERM	Specifies the error mode. 0: NMI request mode 1: Reset mode

WDTAn OVF2	WDTAn OVF1	WDTAn OVF0	Overflow Interval Time
0	0	0	2^9 / WDTATCKI
0	0	1	2^{10} / WDTATCKI
0	1	0	2^{11} / WDTATCKI
0	1	1	2^{12} / WDTATCKI
1	0	0	2^{13} / WDTATCKI
1	0	1	2^{14} / WDTATCKI
1	1	0	2^{15} / WDTATCKI
1	1	1	2^{16} / WDTATCKI

Table 26.18 WDTAnMD Register Contents (2/2)

Bit Position	Bit Name	Function
1, 0	WDTAnWS[1:0]	Selects the window-open period.

WDTAn WS1	WDTAn WS0	Window-Open Period
0	0	25%
0	1	50%
1	0	75%
1	1	100%

26.4 Interrupt Sources

WDTA detects the status of the WDTA counter value or illegal accesses to the WDTA-related registers, and generates an interrupt request. The following are WDTA interrupt requests:

(1) INTWDTAn (WDTA timer count 75% interrupt request)BodyBracketNum
An interrupt request signal is generated at 75% of the counter overflow time of the WDTA timer. An interrupt request signal can be set to be enabled or disabled by using the WDTA mode register (WDTAnMD).

(2) WDTAnTNMI (WDTA error detection interrupt)
Detection of a WDTA error to generation of an NMI interrupt request. The WDTA mode register (WDTAnMD) can be used to switch an NMI interrupt and a reset. For details about WDTA errors, see Section 26.5.3, WDTA Error Detection.

26.5 Functions

26.5.1 WDTA after Reset Release

26.5.1.1 Start Modes

There are two start modes (software start mode and default start mode) when WDTAn starts after reset release. The start mode can be selected by the start-up option OPWDRUN.

The start mode selection is listed in **Table 26.19**.

Table 26.19 Start Mode Selection

Start-Up Options OPWDRUN	Start Mode	Description
0	Software trigger	• The WDTA counter stops (0000_H) after reset release. • Writing an activation code to the WDTA trigger register starts WDTA.
1	Default	The WDTA counter starts after reset release.

26.5.1.2 Start mode selection

The start mode can be selected as follows:

- By start-up options
- By the WDTATRTYP input signal
  This signal indicates the reset type. Thus, the selected start mode after reset release depends on the

reset type.

The start mode selection is listed in the following table.

Table 26.20 Start mode selection

Start-up options		Input signal		
OPWDTPR	**OPWDRUN**	**WDTATRTYP**	**Reset type**	**Start mode**
0	0	Ignored	Ignored	Software trigger
	1			Default
1	0	Ignored	Ignored	Software trigger
	1	0	PURES*1*2 POC1RES ISORES(WDTA1)*3	Software trigger
		1	Any except PURES, POC1RES, ISORES(WDTA1)*2	Default

Note 1. PURES is asserted by power-up or debugger reset.

Note 2. Refer to Section 26.1.6, WDTATRTYP for reset conflict cases.

Note 3. ISORES does not reset WDTA0.
In case default start mode needed after DEEPSTOP, insert trigger to start WDTA1 operation.

26.5.1.3 WDTA Settings after Reset Release

Table 26.21 shows the WDTA settings after reset release.

Table 26.21 WDTA Settings after Reset Release

Function	Setting	Remark
WDTA enable/disable	Specified by start-up options	
Start mode		
VAC function		
WDTA overflow interval time	Specified by start-up options	Modification is possible only once by the setting of the WDTA mode register (WDTAnMD).
75% interrupt mode	75% interrupt disabled	
Behavior on error detection	Reset generation	
Window-open period	100%	

The setting of the WDTA mode register (WDTAnMD) is enabled when the first WDTA trigger is generated (writing an activation code to WDTAnWDTE and WDTAnEVAC). Perform the WDTAnMD register setting before a WDTA trigger is generated.

Setting of WDTA by using WDTAnMD is possible only once. If the value set for WDTAnMD is changed after a WDTA trigger is generated, an error occurs. However, an error does not occur if the same value has been set.

26.5.1.4 Default Start Mode Timing

The default start mode timing and the changes to the WDTA settings are illustrated in **Figure 26.2**.

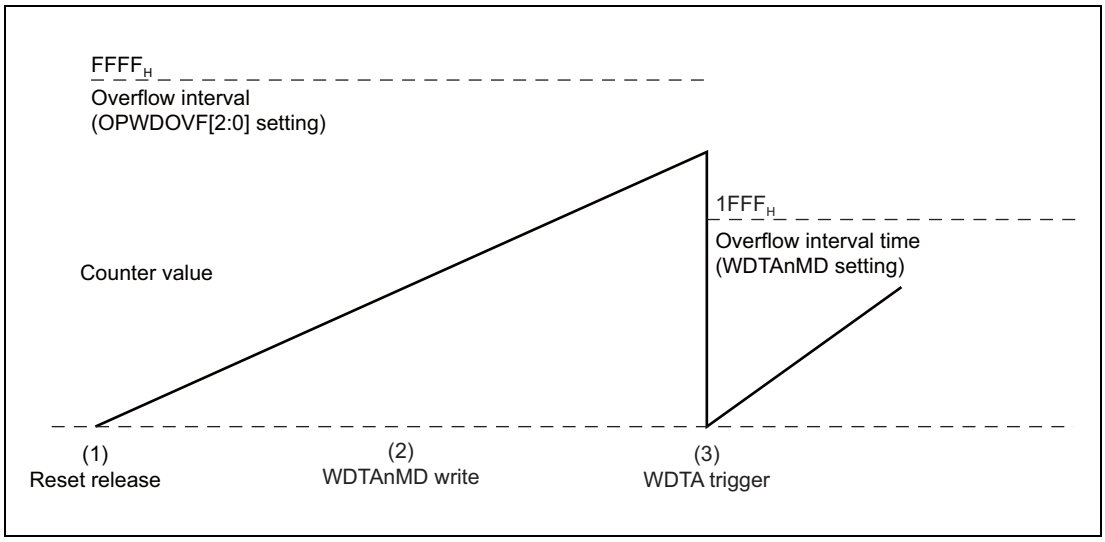


Figure 26.2 Timing Diagram of WDTA Start in Default Start Mode

The timing diagram shown in **Figure 26.2** shows the following behaviors:

(1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

Example: Overflow interval time after reset release
$= 2^{16}$/WDTATCKI (OPWDOVF[2:0] = 111_B)

(2) WDTAnMD is set before a WDTA trigger is generated. Note, however, that the setting is not applied immediately.

(3) Write to the WDTA trigger register before the WDTA counter overflows.
The WDTAnMD setting is applied due to the WDTA trigger.

Example: Overflow interval time after a WDTA trigger is generated
$= 2^{13}$/WDTATCKI

26.5.1.5 Software Trigger Start Mode Timing

The software trigger start mode timing and the changes to the WDTA settings are illustrated in **Figure 26.3**.

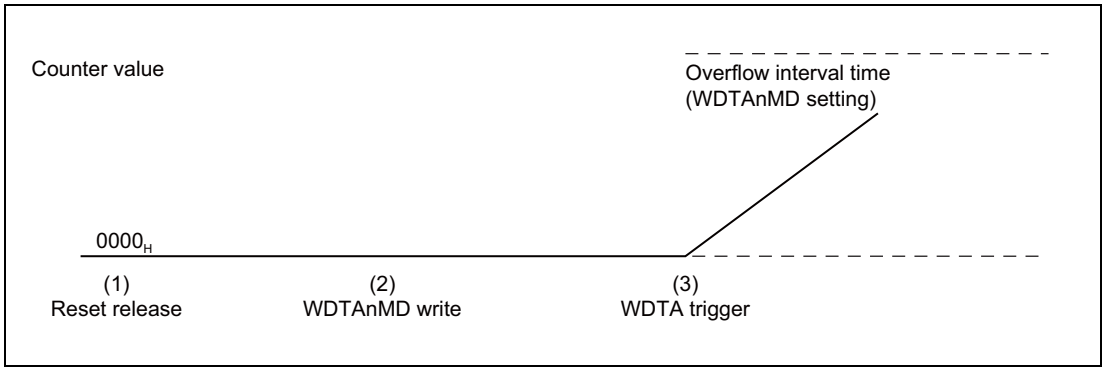


Figure 26.3 Timing Diagram of WDTA Start in Software Trigger Start Mode

The timing diagram shown in **Figure 26.3** shows the following behaviors:

(1) The WDTA counter value remains 0000_H until the first trigger is generated after reset release. The overflow interval time is set by start-up options, but it does not affect because the counter operation has not been performed.

(2) WDTAnMD is set before a WDTA trigger is generated. Note, however, that the setting is not applied immediately.

(3) The WDTA counter starts due a WDTA trigger.
The overflow interval time specified in WDTAnMD and other settings are applied.

26.5.2 WDTA Trigger

Writing a special value called an activation code to the WDTA enable register (WDTAnWDTE) and the WDTA enable VAC register (WDTAnEVAC) leads to generation of a WDTA trigger.

The WDTA trigger has the following functions:

- Starting the WDTA counter in software trigger start mode
- Restarting the WDTA counter
- WDTA mode setting by the WDTAnMD register (only for the first WDTA trigger after reset release)

The WDTA trigger register, which generates a WDTA trigger, is specified by the start-up option OPWDVAC.

Table 26.22 lists the WDTA trigger registers and activation codes.

Table 26.22 WDTA Trigger and Activation Code

Type of Activation Code	Trigger Register	Activation Code
Fixed (OPWDVAC = 0)	WDTAnWDTE	AC_H
Variable (OPWDVAC = 1)	WDTAnEVAC	For details, see 26.5.2.1, Calculating an Activation Code when the VAC Function is Used.

When successively writing the processing to clear WDTA, the clear processing below is not acknowledged for the following period:

12 × CPU clock[*1] cycles + 6 × WDT clock[*2] cycles

After writing the processing to clear WDTA and then changing to standby mode during the above mentioned period, the clear processing below is not acknowledged for the following period after return from stand-by mode:

6 × CPU clock[*1] cycles + 3 × WDT clock[*2] cycles

NOTES

1. CPU clock: C_ISO_CPUCLK
2. WDT clock: Clock domain connected to WDTATCKI

26.5.2.1 Calculating an Activation Code when the VAC Function is Used

Use the following expression to calculate the variable activation code (ExpectWDTE) to be set to the WDTA trigger register (WDTAnEVAC) when the VAC function is used, by using the WDTA reference value register (WDTAnREF):

$$\text{ExpectWDTE} = AC_H - \text{WDTAnREF (previous)}$$

Note that the value in the WDTAnREF register is updated every time a start-code is written to the trigger register WDTAnEVAC. Use the following expression to calculate the updated value of the WDTAnREF register:

WDTAnREF (following) = (rotate the value of ExpectWDTE to the left by 1 bit)

Table 26.23 lists the variable activation codes according to the number of WDTA triggers.

Table 26.23 Expected Variable Activation Code Development

No[*1]	WDTAnREF (Previous)		ExpectWDTE (AC_H - WDTAnREF)		WDTAnREF (Following)	
0	0000 0000	00_H	1010 1100	AC_H	0101 1001	59_H
1	0101 1001	59_H	0101 0011	53_H	1010 0110	$A6_H$
2	1010 0110	$A6_H$	0000 0110	06_H	0000 1100	$0C_H$
…	…	…	…	…	…	…

Note 1. Number of triggers after reset

NOTE

Writing a wrong activation code generates an error.

26.5.3 WDTA Error Detection

WDTA detects an error, including generation of the WDTA count overflow or illegal operations.

The following shows when a WDTA error is detected:

- WDTA counter overflow
- Wrong activation code is written to the WDTA trigger register
- Writing to the trigger register at a time outside the window-open period.
- When the setting value in the WDTA mode register (WDTAnMD) is changed after the first WDTA trigger is generated
- When the setting value in the WDTA mode register (WDTAnMD) is changed twice before the WDTA trigger is generated

26.5.3.1 WDTA Error Mode

When a WDTA error is detected, either an NMI interrupt or a reset is generated according to the setting of the WDTA error mode bit (WDTAnMD.WDTAnERM). The error mode bit after reset release is set to the reset mode.

- WDTAnMD.WDTAnERM = 0: NMI mode
- WDTAnMD.WDTAnERM = 1: reset mode

Figure 26.4 shows the reset or NMI request generation when the counter overflows and default start mode is selected.

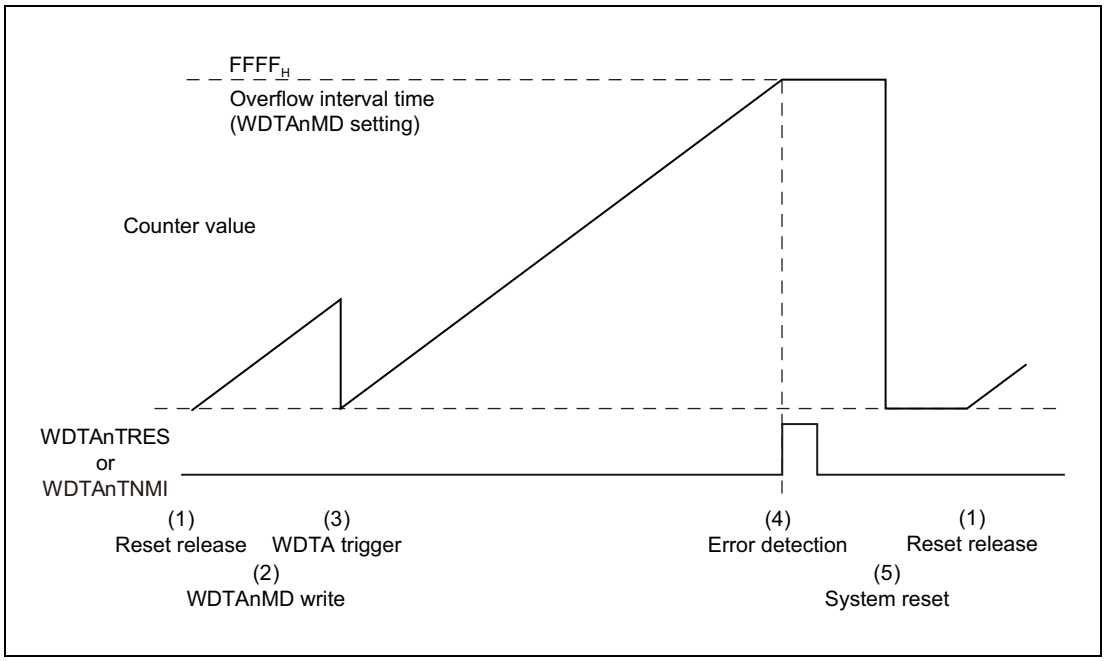


Figure 26.4 Timing Diagram of WDTA NMI Request or Reset Generation

The timing diagram shown in **Figure 26.4** shows the following behaviors:

(1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

(2) WDTAnMD is set before a WDTA trigger is generated.
In this case, 2^{16}/WDTATCKI is set for the overflow interval time.

(3) A WDTA trigger is generated, and the WDTAnMD setting is applied.

(4) When the counter overflows, an error is detected. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated.
The counter value remains until a system reset is performed.

(5) When the system is reset, the counter is cleared and stopped until reset release.

26.5.4 75% Interrupt Request Signals

When the WDTA counter reaches 75% of the time set for the overflow interval, the interrupt request INTWDTAn is generated.

By use of the WDTAnMD.WDTAnWIE register, this function can be enabled or disabled afterwards.

Figure 26.5 shows the 75% interrupt request generation under following conditions:

- Default start mode selected
- 75% interrupt request is enabled after the first WDTA trigger is generated
- WDTA overflow interval time: 2^{16}/WDTATCKI

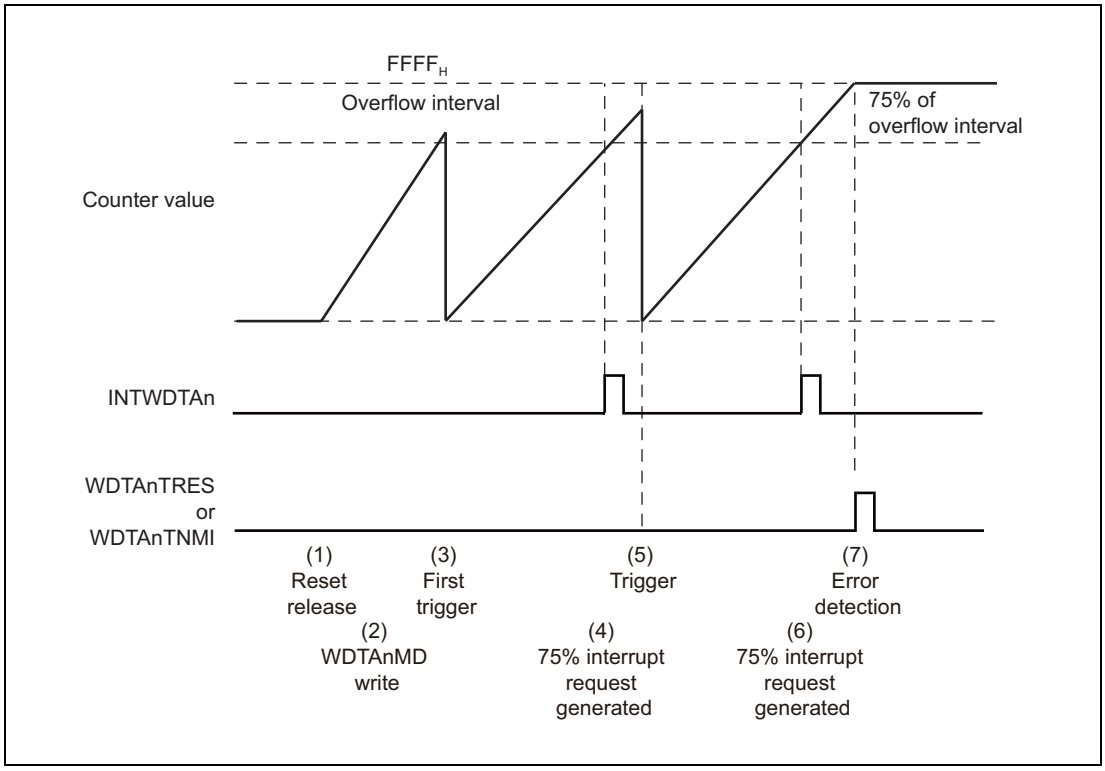


Figure 26.5 Timing Diagram of WDTA 75% Interrupt Request Signals

(1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

(2) WDTAnMD is set before a WDTA trigger is generated. In this case, 2^{16}/WDTATCKI is set for the overflow interval time.

(3) A WDTA trigger is generated, and the WDTAnMD setting is applied.

(4) When the WDTA counter reaches 75% of the overflow interval timer, interrupt request INTWDTAn is generated.

(5) The WDTA trigger restarts the counting.

(6) When the WDTA counter reaches 75% of the overflow interval timer, interrupt request INTWDTAn is generated.

(7) When the counter overflows, an error is detected. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated. The counter value remains until a system reset is performed.

26.5.5 Window Function

The period when a WDTA trigger is valid (window-open period) can be set. If the window-open period is set to the value less than 100%, an error occurs by the WDTA trigger generated not in the window-open period. The window-open period after reset release is 100%. The period is set to the value by the WDTAnMD.WDTAnWS[1:0] setting after the first WDTA trigger is generated.

Figure 26.6 shows the behavior of the window function under the following conditions.

- Default start mode selected
- 25% window-open period is enabled after the first WDTA trigger is generated (WDTAnWS[1:0] = 00_B)
- WDTA overflow interval time: 2^{16}/WDTATCKI

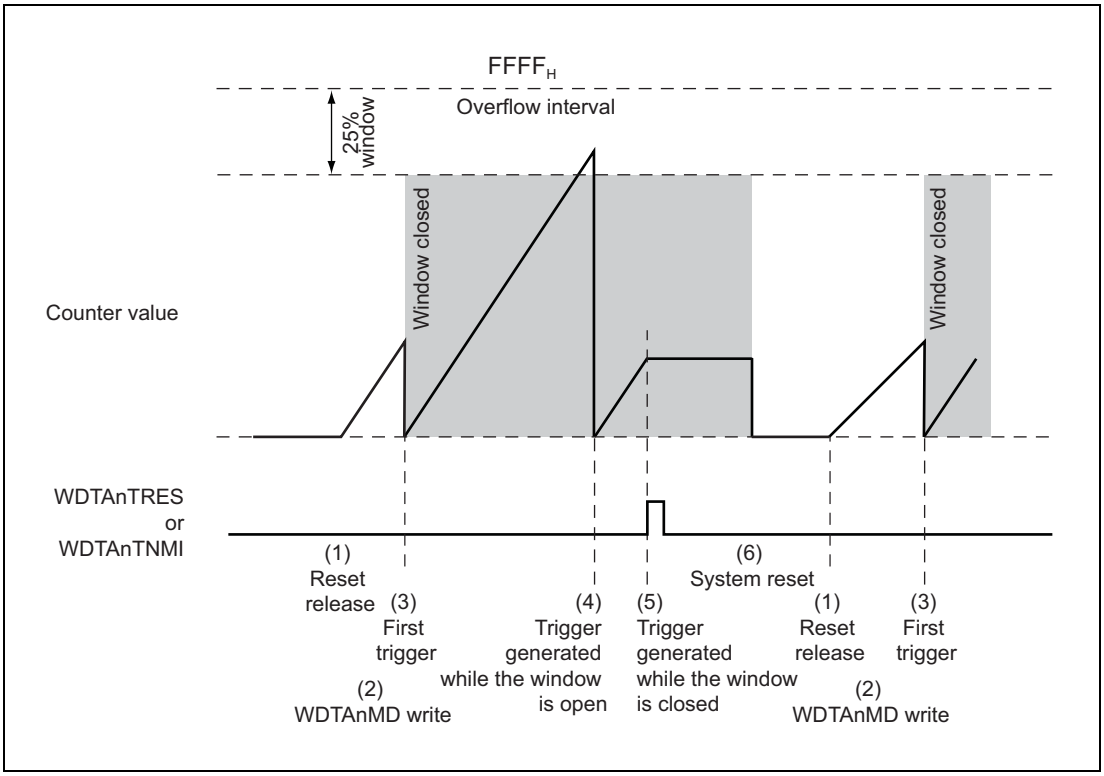


Figure 26.6 Timing Diagram of WDTA Window Function

(1) In default start mode, the WDTA counter starts after reset release. The overflow interval time after reset release is set by start-up options.

(2) WDTAnMD is set before a WDTA trigger is generated. In this case, 2^{16}/WDTATCKI is set for the overflow interval time.

(3) A WDTA trigger is generated, and the WDTAnMD setting is applied.

(4) During the window-open period, the WDTA trigger restarts the counting.

(5) During the window-closed period, an error is detected by the WDTA trigger. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated. The counter value remains until a system reset is performed.

(6) When the system is reset, the counter is cleared and stopped until reset release.

Section 27 OS Timer (OSTM)

This section contains a generic description of the OS Timer (OSTM).

The first part of this section describes all RH850/D1L/D1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the OSTM.

27.1 Features of RH850/D1L/D1M OSTM

27.1.1 Number of Units

This microcontroller has the following number of units of the OSTM.

Table 27.1 Number of Units

Product Name	All products
Number of Units	2
Name	OSTMn (n = 0, 1)

Table 27.2 Index

Index	Meaning
n	Throughout this section, the individual OSTM units are identified by the index “n” (n = 0, 1); for example, OSTMnCNT is the OSTM counter register.

27.1.2 Register Base Address

OSTM base addresses are listed in the following table.

OSTM register addresses are given as offsets from the base addresses in general.

Table 27.3 Register Base Address

Base Address Name	Base Address
<OSTM0_base>	FFEC 0000_H
<OSTM1_base>	FFEC 1000_H

27.1.3 Clock Supply

The OSTM clock supply is shown in the following table.

Table 27.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
OSTMn	PCLK	C_ISO_OSTM

27.1.4 Interrupt Request

OSTM interrupt requests are listed in the following table.

Table 27.5 Interrupt Requests

OSTMn signals	Function	Connected to
OSTM0TINT	OSTM0 interrupt	Interrupt Controller INTOSTM0
OSTM1TINT	OSTM1 interrupt	Interrupt Controller INTOSTM1 Error Control Module input

27.1.5 Reset Sources

OSTM reset sources are listed in the following table. OSTM is initialized by these reset sources.

Table 27.6 Reset Sources

Unit Name	Reset Source
OSTM0	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode
OSTM1	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

27.1.6 External Input/Output Signals

External input/output signals of OSTM are listed below.

Table 27.7 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
OSTM0		
OSTM0TTOUT	OSTM output signal	Port OSTM0TTOUT Error Control Module timer input
OSTM1		
OSTM1TTOUT	OSTM output signal	Port OSTM1TTOUT

27.2 Overview

OSTM is a 32-bit timer/counter.

It can be used in interval timer mode or in free-run compare mode. The settings for operating mode specify the direction of counting (up or down) to control the generation of interrupt requests.

27.2.1 Functional Overview

OSTM has the following features.

- Two operating modes
  - Interval timer mode
  - Free-run compare mode
- OSTMnTINT interrupt
- Timer output signal with two modes
  - Software trigger mode
  - Toggle mode

27.2.2 Block Diagram

The following block diagram shows the main components of OSTM.

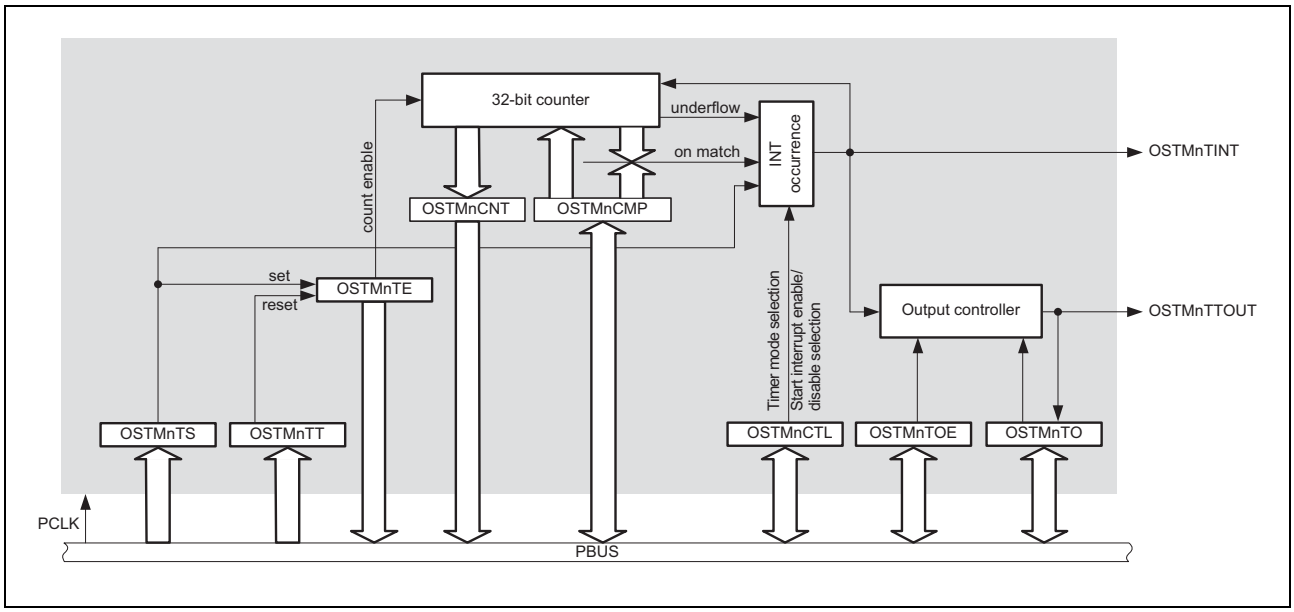


Figure 27.1 Block diagram of OSTM

27.2.3 Count Clock

The count clock used by OSTM is PCLK.

27.2.4 Output Modes

OSTM has the following output modes:

- Software control mode:
  The level set to OSTMnTO.OSTMnTO is output to OSTMnTTOUT.
- Toggle mode:
  OSTMnTTOUT toggles when the OSTMnTINT request is generated.

The output mode is selected by bit OSTMnTOE.OSTMnTOE.

Both output modes are illustrated in the following figure.

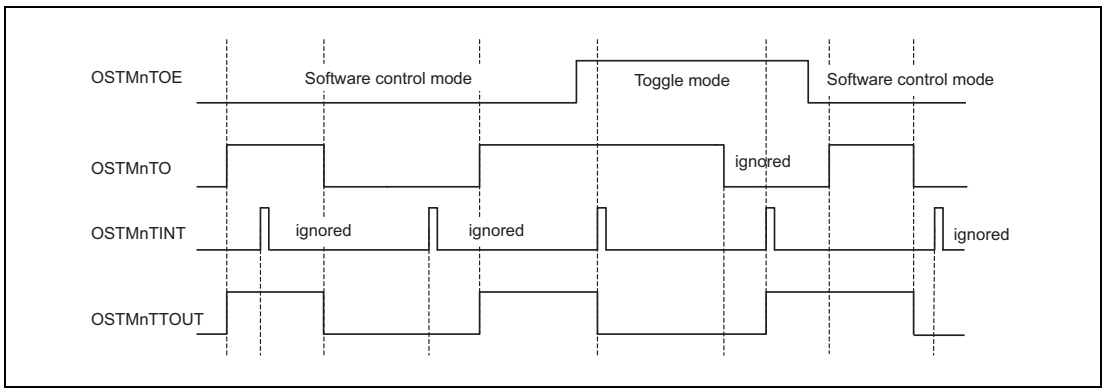


Figure 27.2 Timing diagram of output modes

The timing diagram above shows the following:

- In software control mode, OSTMnTTOUT changes to the value that was set in OSTMnTO.OSTMnTO.
- In toggle mode, OSTMnTTOUT toggles when OSTMnTINT interrupt request is generated.

27.2.5 Interrupt Sources (OSTMnTINT)

By default, an OSTMnTINT interrupt request is generated on counter underflow (interval timer mode) or when the counter matches the compare value (free-run compare mode).

An interrupt request can also be generated on starting and restarting of the counter. This is controlled by the OSTMnCTL.OSTMnMD0 bit.

Since OSTMnTINT triggers the toggling of OSTMnTTOUT in toggle mode (OSTMnTOE.OSTMnTOE = 1), the setting of the OSTMnCTL.OSTMnMD0 bit also influences the output of OSTMnTTOUT.

This is illustrated in the following figure.

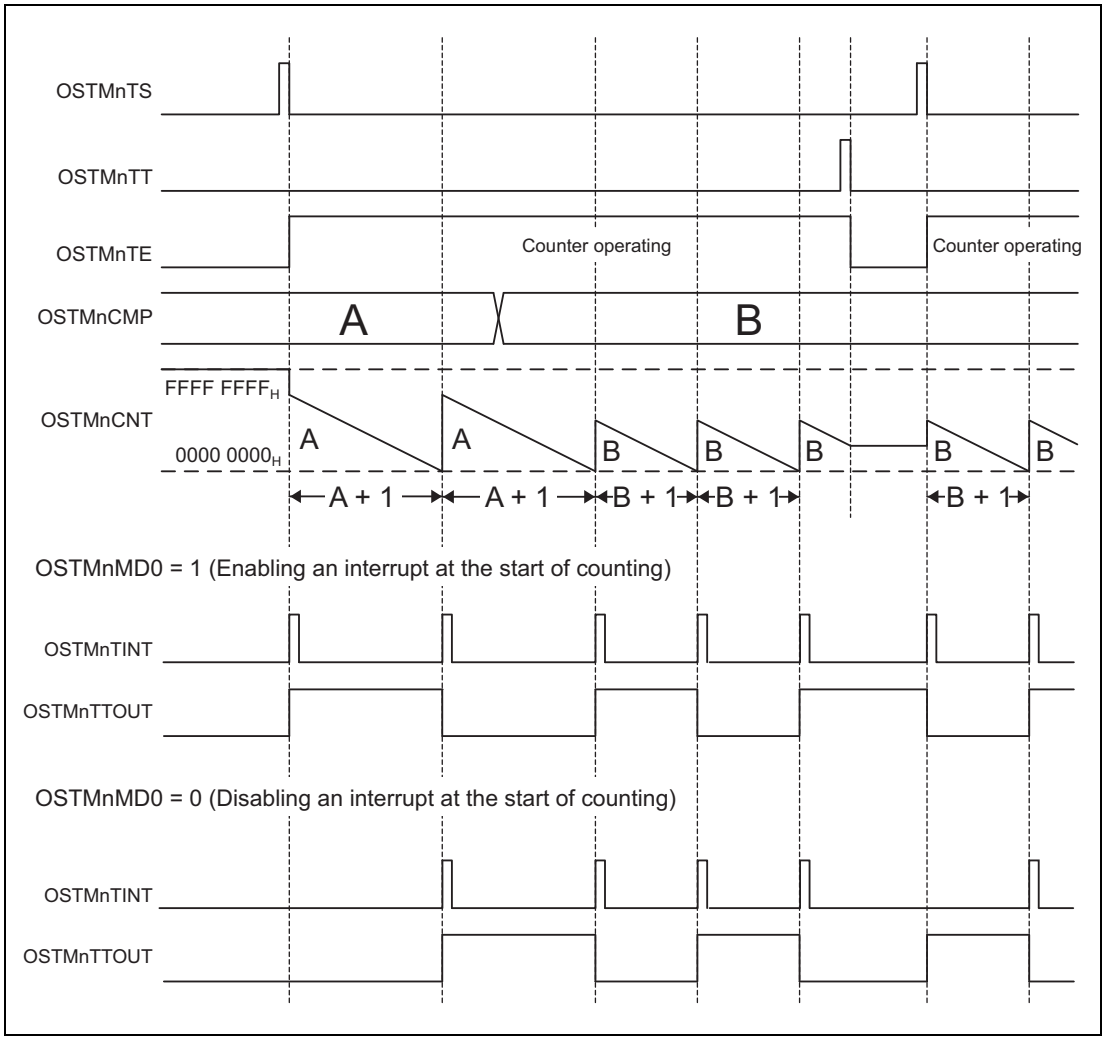


Figure 27.3 Generating an Interrupt when Counting Starts (in Interval Timer Mode)

27.3 Registers

27.3.1 List of Registers

OSTMn registers are listed in the following table.

For details about <OSTMn_base>, see Section 27.1.2, Register Base Address.

Table 27.8 Registers

Module	Register	Symbol	Address
OSTMn	OSTMn compare register	OSTMnCMP	<OSTMn_base> + 00_H
OSTMn	OSTMn counter register	OSTMnCNT	<OSTMn_base> + 04_H
OSTMn	OSTMn output register	OSTMnTO	<OSTMn_base> + 8_H
OSTMn	OSTMn output enable register	OSTMnTOE	<OSTMn_base> + C_H
OSTMn	OSTMn count enable status register	OSTMnTE	<OSTMn_base> + 10_H
OSTMn	OSTMn count start trigger register	OSTMnTS	<OSTMn_base> + 14_H
OSTMn	OSTMn count stop trigger register	OSTMnTT	<OSTMn_base> + 18_H
OSTMn	OSTMn control register	OSTMnCTL	<OSTMn_base> + 20_H
OSTMn	OSTMn emulation register	OSTMnEMU	<OSTMn_base> + 24_H

27.3.2 OSTMnCMP - OSTMn Compare Register

This register stores the start value of the down-counter or the value with which the counter is compared, depending on the operation mode.

Access: This register can be read/written in 32-bit units.

Address: <OSTMn_base> + 00_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OSTMnCMP[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTMnCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27.9 OSTMnCMP Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCMP [31:0]	• In interval timer mode: start value of the down-counter • In free-run compare mode: compare value

27.3.3 OSTMnCNT — OSTMn Counter Register

This register indicates the counter value of the timer.

Access: This register can only be read in 32-bit units.

Address: <OSTMn_base> + 04_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OSTMnCNT[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OSTMnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 27.10 OSTMnCNT Register Contents

Bit Position	Bit Name	Function
31 to 0	OSTMnCNT [31:0]	Timer counter value

Table 27.11 lists the correspondence among the operating mode, counting direction, and start value. The start value indicates the value to be read after the operating mode is changed.

Table 27.11 Correspondence among Operating Mode, Counting Direction, and Start Value

Timer Operating Mode	OSTMnCTL.OSTMnMD1	Counting Direction	Start Value
Interval timer mode	0*1	Down	FFFF $FFFF_H$
Free-run compare mode	1	Up	0000 0000_H

Note 1. Value after reset.

27.3.4 OSTMnTO - OSTMn Output Register

This register specifies and reads the level of OSTMnTTOUT.

Access: This register can be read/written in 8-bit units.
It can only be written when the software control mode is enabled (OSTMnTOE.OSTMnTOE = 0).

Address: <OSTMn_base> + 8_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTO
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 27.12 OSTMnTO Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	OSTMn TO	Specifies/reads the level of OSTMnTTOUT: 0: Low level 1: High level

27.3.5 OSTMnTOE - OSTMn Output Enable Register

This register specifies OSTMnTTOUT output mode.

Access: This register can be read/written in 8-bit units.

Address: <OSTMn_base> + C_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTOE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 27.13 OSTMnTOE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	OSTMn TOE	Specifies the OSTMnTTOUT output mode: 0: Software control mode: The level set to OSTMnTO.OSTMnTO is output to OSTMnTTOUT. 1: Toggle mode: OSTMnTTOUT toggles when the interrupt request OSTMnTINT is generated.

27.3.6 OSTMnTE — OSTMn Count Enable Status Register

This register indicates whether the counter is enabled or disabled.

Access: This register can only be read in 8-bit units.

Address: <OSTMn_base> + 10_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 27.14 OSTMnTE Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	OSTMnTE	Indicates whether the counter is enabled or disabled: 0: Counter disabled 1: Counter enabled This bit is set to 1 in response to OSTMnTS.OSTMnTS being set to 1. Setting OSTMnTT.OSTMnTT to 1 resets this bit to 0.

NOTE

If the counter is disabled, the counter value retains its value.

If the counter is restarted, it

- restarts counting down from the value in the OSTMnCMP register if it is in interval timer mode or
- restarts counting up from the counter value 0000 0000_H if it is in free-run compare mode.

27.3.7 OSTMnTS — OSTMn Count Start Trigger Register

This register starts the counter.

Access: This register can only be written in 8-bit units.

Address: <OSTMn_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 27.15 OSTMnTS Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	OSTMnTS	Starts the counter: 0: This setting is invalid. 1: Starts the counter and sets OSTMnTE.OSTMnTE = 1. • In interval timer mode, a forced restart is executed if this bit is set while OSTMnTE.OSTMnTE = 1. • In free-run compare mode, setting this bit is ignored as long as OSTMnTE.OSTMnTE = 1.

27.3.8 OSTMnTT — OSTMn Count Stop Trigger Register

This register stops the counter.

Access: This register can only be written in 8-bit units.

Address: <OSTMn_base> + 18_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	OSTMnTT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 27.16 OSTMnTT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	OSTMnTT	Stops the counter: 0: This setting is invalid. 1: Stops the counter and clears the OSTMnTE.OSTMnTE bit.

27.3.9 OSTMnCTL — OSTMn Control Register

This register specifies the operating mode for the counter and controls the generation of OSTMnTINT interrupt requests when counting starts.

Although this register is readable and writable, writing to it is only possible when OSTMnTE.OSTMnTE = 0; that is, the register becomes read only when OSTMnTE.OSTMnTE = 1.

Access: This register can be read/written in 8-bit units.

Address: <OSTMn_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	OSTMnMD1	OSTMnMD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 27.17 OSTMnCTL Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1	OSTMnMD1	Specifies the operating mode for the counter: 0: Interval timer mode 1: Free-run compare mode
0	OSTMnMD0	Controls the generation of OSTMnTINT interrupt requests at the start of counting: 0: Interrupts when counting starts are disabled. 1: Interrupts when counting starts are enabled.

27.3.10 OSTMnEMU — OSTMn Emulation Register

This register controls operation in combination with SVSTOP.

Access: This register can be read/written in 8-bit units.
Only proceed with writing while the counter is stopped (OSTMnTE.OSTMnTE = 0 and EPC.SVSTOP = 0).

Address: <OSTMn_base> + 24_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	OSTMnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 27.18 OSTMnEMU Register Contents

Bit Position	Bit Name	Function
7	OSTMnSVSDIS	When EPC.SVSTOP = 0 Supply of the count clock is continued when the debugger acquires control of the microcontroller (at breakpoints and so on) regardless of the value of this bit (0/1). When EPC.SVSTOP = 1 0: The count clock is stopped when the debugger acquires control of the microcontroller (at breakpoints and so on). 1: Supply of the count clock is continued when the debugger acquires control of the microcontroller (at breakpoints and so on).
6 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

27.4 Operation

27.4.1 Starting and stopping OSTM

OSTM is started and stopped as follows:

Starting the timer

OSTM is started by the following setting.

- Setting the OSTMnTS.OSTMnTS bit to 1

The OSTMnTE.OSTMnTE status bit is set to 1.

The counter starts to count up or down in accordance with the settings for operating mode. For details, see Section 27.4.2, Interval timer mode and Section 27.4.3, Free-Run Compare Mode.

Stopping the timer

Setting the OSTMnTT.OSTMnTT bit to 1 stops OSTM.

This also clears the OSTMnTE.OSTMnTE status bit.

27.4.2 Interval timer mode

In interval timer mode, OSTM can be used as a reference timer generating interrupt requests at fixed intervals.

27.4.2.1 Basic operation in interval timer mode

In interval timer mode, the timer counts down from the value specified in the OSTMnCMP register. An OSTMnTINT interrupt request is generated when the counter underflows (reaches 0000 0000_H).

To select interval timer mode, set OSTMnCTL.OSTMnMD1 = 0.

New values can be written to the OSTMnCMP register at any time. If it is rewritten during count operation, the counter loads the new OSTMnCMP value when the next 0000 0000_H is reached. Then the counter continues with the new value.

OSTMnTINT and OSTMnTTOUT periods

The period of OSTMnTINT and OSTMnTTOUT are:

- OSTMnTINT generation period = counter clock period × (OSTMnCMP + 1)
- OSTMnTTOUT period = OSTMnTINT occurrence period × 2

The following figure shows the basic operation of OSTM when counter-start interrupts is enabled in interval timer mode and OSTMnTTOUT in toggle mode.

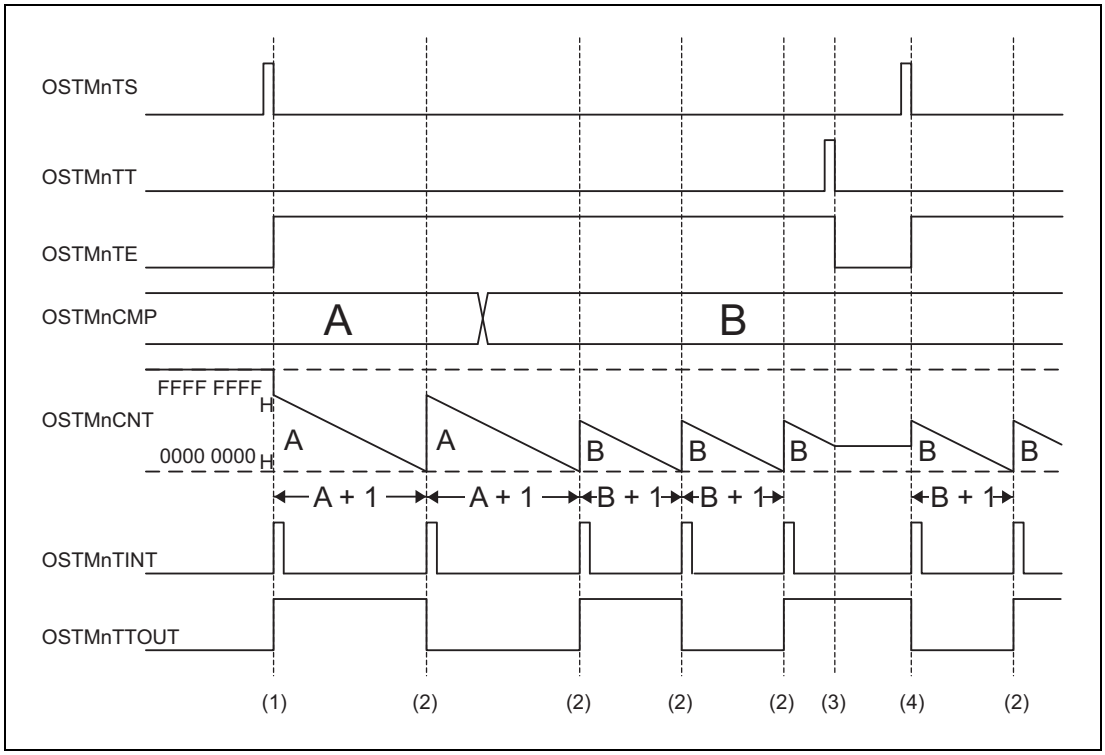


Figure 27.4 Timing Diagram of OSTM in Interval Timer Mode

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter.
The counter starts counting down from the value of OSTMnCMP.
If OSTMnCTL.OSTMnMD0 is 1, OSTMnTINT interrupt requests are generated at the start of counting and the OSTMnTTOUT output toggles. The OSTMnCNT register contains the current value as the counter.

(2) When the counter reaches 0000 0000_H, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT output toggles. The counter loads the new start value from OSTMnCMP and continues counting down.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.

(4) When counting is restarted (OSTMnTS.OSTMnTS = 1), the counter loads the new start value from OSTMnCMP and starts counting down.

Forced restart

The counter is forcibly restarted by setting OSTMnTS.OSTMnTS = 1 during counting.

The counter loads the start value from the OSTMnCMP register and continues to count down.

The following figure shows the forced restart of OSTM in interval timer mode, with counter-start interrupts enabled (OSTMnCTL.OSTMnMD0 = 1) and OSTMnTTOUT in toggle mode (OSTMnTOE.OSTMnTOE = 1).

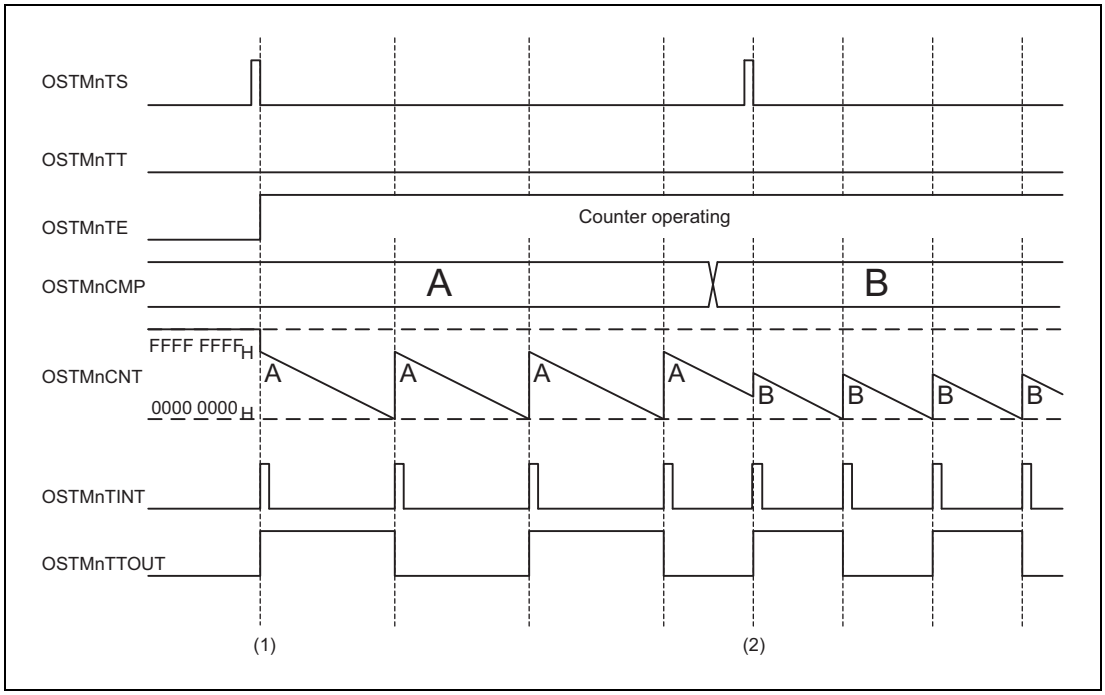


Figure 27.5 Timing Diagram of Forced Restart in Interval Timer Mode

Operations shown in the above timing diagram are as follows.

(1) The counter is started and stopped as described under Figure 27.4, Timing Diagram of OSTM in Interval Timer Mode.

(2) Setting OSTMnTS.OSTMnTS = 1 restarts the counter while counting is in progress (i.e. while OSTMnTE.OSTMnTE = 1).
The counter immediately restarts counting down, starting with the current value of OSTMnCMP. When OSTMnCTL.OSTMnMD0 = 1, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT output toggles when counting starts.

27.4.2.2 Operation when OSTMnCMP = 0000 0000$_H$

When OSTMnCMP = 0000 0000$_H$, OSTM behaves as follows.

- When the counter is enabled, the OSTMnTINT interrupt request is always set to 1.
- If OSTMnTTOUT output is in toggle mode, OSTMnTTOUT toggles with the PCLK period.

The following figure shows operations of OSTM when OSTMnCMP = 0000 0000$_H$, counter-start interrupts are enabled and OSTMnTTOUT is in toggle mode.

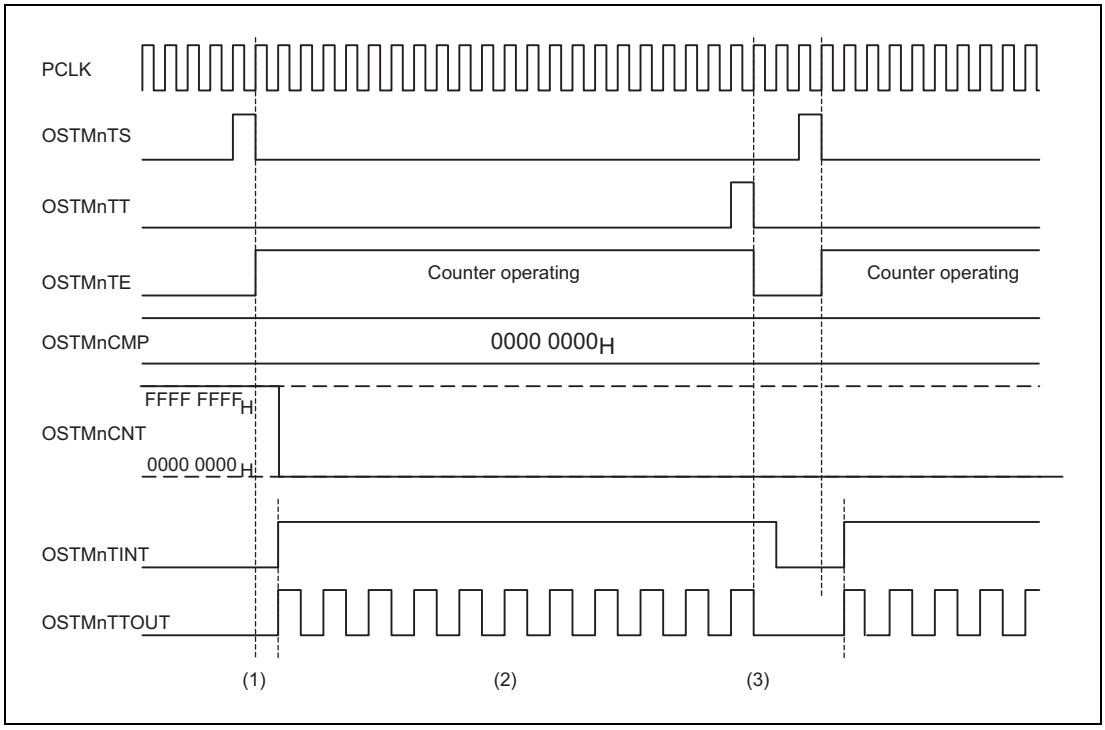


Figure 27.6 Timing Diagram when OSTMnCMP = 0000 0000$_H$ in Interval Timer Mode

The timing diagram above shows the following operations:

(1) The counter is reloaded with the value in OSTMnCMP as soon as it starts counting, so the value 0000 0000$_H$ is retained in OSTMnCMP.

(2) The OSTMnTINT interrupt request is continuously asserted and OSTMnTTOUT starts toggling with the PCLK clock period.

(3) After the counter stops, the OSTMnTINT interrupt request signal is deasserted and OSTMnTTOUT stops toggling.

(4) When interrupts on starting of the counter are disabled, OSTMnTINT and OSTMnTTOUT are not generated for one clock cycle of the count clock when counting starts.

27.4.2.3 Setting Procedure for Interval Timer Mode

The setting procedure in interval timer mode after reset release is described below:

Setting procedure

(1) Set the start value of the counter in the OSTMnCMP register.

(2) Select interval timer mode by clearing the OSTMnCTL.OSTMnMD1 bit.

(3) Enable or disable interrupts when counting starts (OSTMnCTL.OSTMnMD0).

(4) To use the OSTMnTTOUT output pin:
- Select the output mode (OSTMnTOE.OSTMnTOE).
- In software control mode, initialize OSTMnTO.

27.4.3 Free-Run Compare Mode

27.4.3.1 Basic operation in free-run compare mode

In free-run compare mode, the counter counts up from 0000 0000_H to FFFF $FFFF_H$. When the value of the OSTMnCMP register matches the current counter value, an OSTMnTINT interrupt request is output.

In free-run compare mode, set OSTMnCTL.OSTMnMD1 = 1.

New values can be written to the OSTMnCMP register at any time.

The following figure shows the basic operation of OSTM in free-run compare mode with the start of counting enabled (OSTMnCTL.OSTMnMD0 = 1) and OSTMnTTOUT in toggle mode (OSTMnTOE.OSTMnTOE = 1).

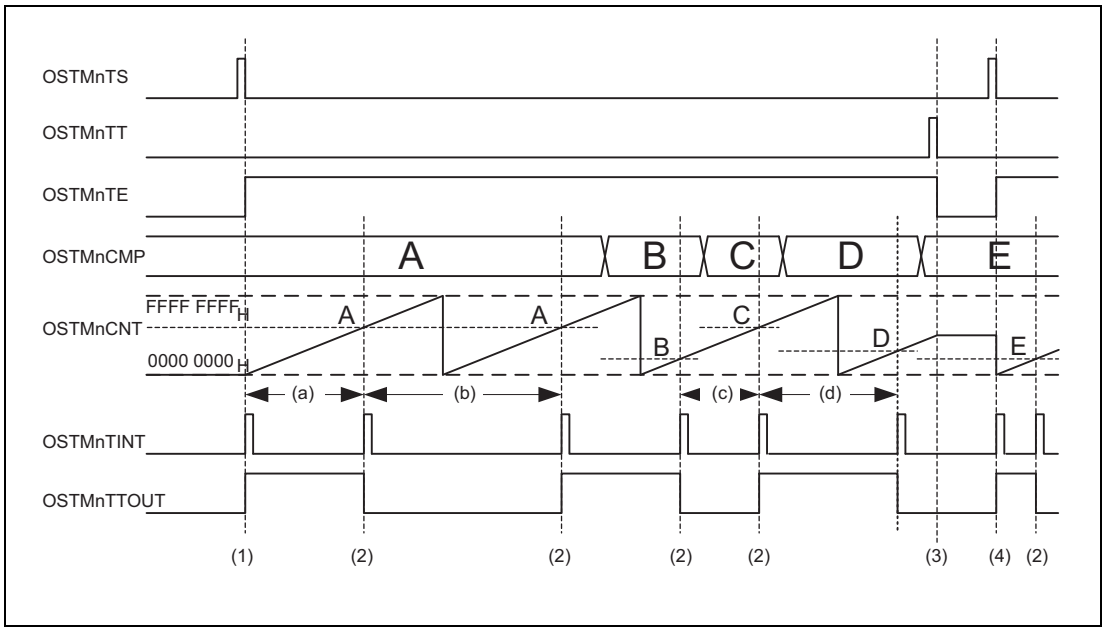


Figure 27.7 Timing Diagram of OSTM in Free-Run Compare Mode

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1.
The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter counts up

from 0000 0000_H to FFFF $FFFF_H$. The OSTMnCNT register is the counter, so it contains the current value.
When OSTMnCTL.OSTMnMD0 = 1, an OSTMnTINT interrupt request is generated at the start of counting.

(2) When the current counter value matches the value in the OSTMnCMP register, an OSTMnTINT interrupt request is generated and the OSTMnTTOUT output toggles.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter.
The counter retains its current value until it is restarted.

(4) Counting by the counter restarts from 0000 0000_H when OSTMnTS.OSTMnTS = 1.

OSTMnTINT period

The OSTMnTINT generation period is different depending on the starting time. If OSTMnCMP is rewritten during operation, the period is changed according to the size of the new and old compare values.

Table 27.19 OSTMnTINT Generation Timing

Old Value for Comparison	New Value for Comparison	Counter Value at Time of Rewriting	Period of OSTMnTINT Generation	Label in Timing Diagram
Counter starts			(A + 1) × counter clock period	(a)
A	A	No rewriting	(FFFF $FFFF_H$ + 1) × counter clock period	(b)
B	C > B	B < counter value < C	(C – B) × counter clock period	(c)
C	D < C	Counter value > D,C	(FFFF $FFFF_H$ – C + D + 1) × counter clock period	(d)

Forced restart

Forced restarting does not proceed during counting even if the OSTMnTS.OSTMnTS bit is set. The counter ignores the attempted setting and continues counting.

27.4.3.2 Operation when OSTMnCMP = 0000 0000$_H$

The following figure shows the operation of OSTM when OSTMnCMP = 0000 0000$_H$, and counter-start interrupts are enabled (OSTMnCTL.OSTMnMD0 = 1) and OSTMnTTOUT is in toggle mode (OSTMnTOE.OSTMnTOE = 1).

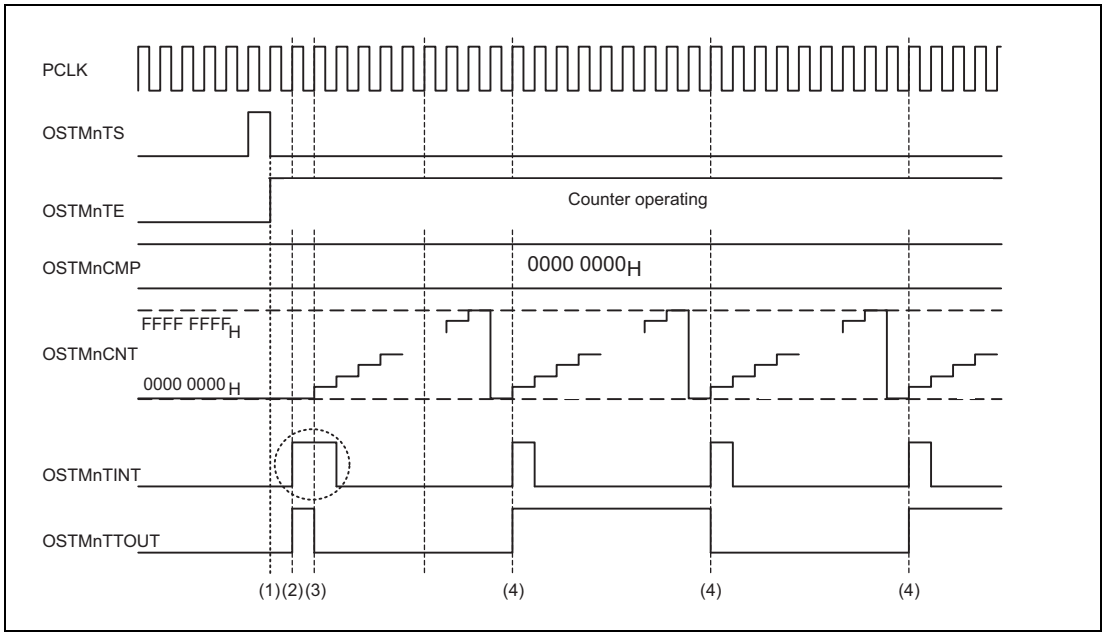


Figure 27.8 Timing Diagram when OSTMnCMP = 0000 0000$_H$ in Free-Run Compare Mode

The timing diagram above shows the following operations.

(1) Once the counter starts, it counts up from 0000 0000$_H$ to FFFF FFFF$_H$.

(2) An OSTMnTINT interrupt request is generated when counting starts.

(3) If the current counter value matches OSTMnCMP, an OSTMnTINT interrupt request is generated. If OSTMnCMP = 0000 0000$_H$ in the above case, OSTMnTINT is generated over two clock cycles.

(4) An OSTMTINT interrupt request is generated and OSTMnTTOUT toggles for each clock cycle (FFFF FFFF$_H$ + 1).

When interrupts on starting of the counter are disabled, OSTMnTINT and OSTMnTTOUT are not generated for one clock cycle of the count clock when counting starts.

27.4.3.3 Setting Procedure for Free-Run Compare Mode

The setting procedure in free-run compare mode after reset release is described below:

Setting procedure

(1) Set the compare value in the OSTMnCMP register.

(2) Select free-run compare mode by setting the OSTMnCTL.OSTMnMD1 bit.

(3) Enable or disable interrupts when counting starts by the OSTMnCTL.OSTMnMD0 bit.

(4) To use the OSTMnTTOUT output pin:
- Select the output mode (OSTMnTOE.OSTMnTOE).
- In software control mode, initialize OSTMnTO.

Section 28 Always-On-Area Timer (AWOT)

This section contains a generic description of the Always-On-Area Timer (AWOT).

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

28.1 Overview of RH850/D1L/D1M Always-On-Area Timer (AWOT)

28.1.1 Units

This microcontroller has the following number of units of the Always-On-Area Timers AWOTn.

Table 28.1 Units

Always-On-Area Timer (AWOT)	
Units	1
Names	AWOT0

Unit index n

Throughout this section, the individual units of the Always-On-Area Timers are identified by the index "n" (n = 0), for example AWOTnCTL for the Always-On-Area Timer n control register.

28.1.2 Register addresses

All Always-On-Area Timers register addresses are given as address offsets from the individual base addresses <AWOTn_base>.
The <AWOTn_base> addresses of each AWOTn are listed in the following table:

Table 28.2 Register base addresses <AWOTn_base>

AWOTn unit	<AWOTn_base> address
AWOT0	FFE4 0000_H

28.1.3 Clock supply

All Always-On-Area Timers provide one clock input.

Table 28.3 Clock supply

AWOTn unit	AWOTn clock	Connected to
AWOT0	PCLK	Clock Controller • C_ISO_PCLK in RUN mode • EMCLK in DEEPSTOP mode
	AWOTCLK	Clock Controller C_AWO_AWOT

CAUTION

The PCLK frequency must be at least the AWOTCLK frequency.

Pay attention to set a correct C_AWO_AWOT clock frequency in DEEPSTOP mode, since the PCLK frequency is $f_{RH}/2$ (i.e. nominal 4 MHz) in DEEPSTOP mode.

28.1.4 Interrupts

The Always-On-Area Timers can generate the following interrupt requests:

Table 28.4 AWOTn interrupt requests

AWOTn signals	Function	Connected to
AWOTINT	AWOT0 interrupt	Interrupt Controller INTAWOT0

28.1.5 Reset sources

The Always-On-Area Timers and their registers are initialized by the following reset signal:

Table 28.5 Reset sources

AWOTn unit	Reset signal
AWOT0	• Reset Controller SYSRES

28.1.6 Internal signal connections

Following AWOTn signals are connected to other modules internally:

Table 28.6 Internal signals connections

AWOTn unit	Signal	Connected to
AWOT0	AWOTIN	Real-Time Clock RTCA0 output RTCA01HZ

28.1.7 I/O signals

The following table shows the I/O signals of the Always-On-Area Timers.

Table 28.7 I/O signals connections

AWOTn unit	Signal	Connected to
AWOT0	AWOTOUT	Port AWOT0TOUT

28.2 Real-Time Clock correction

The AWOT0 capture input signal AWOTIN is connected to the Real-Time Clock 1 second output signal RTCAT1HZ.

In AWOT0 capture mode the AWOT0 can be used to measure the time between two valid edges of the RTCA01s signals in order to calculate a correction value for the Real-Time Clock.

28.3 Functions Overview

Features summary

The Always-On-Area Timer (AWOT) has the following features:

- Three operation modes
  - Interval timer mode with 32-bit resolution
  - Capture mode with 32-bit resolution
  - PWM mode with 16-bit resolution for PWM cycle duration and duty cycle
- Valid edge of capture signal selectable

Block Diagram

The block diagram shows the main components of the AWOT.

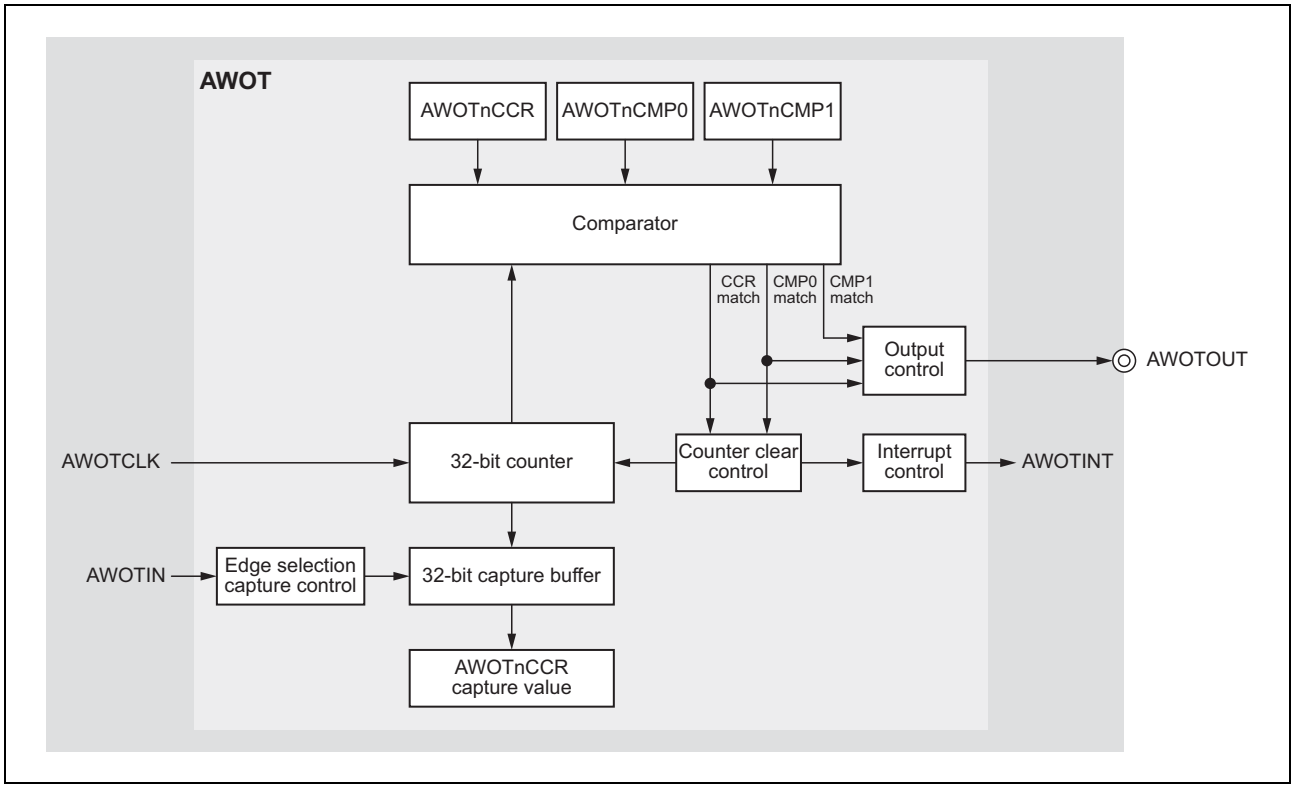


Figure 28.1 Block diagram of the AWOT

28.4 Functional Description

28.4.1 AWOTn Operation Modes

28.4.1.1 Interval timer mode (AWOTnCTL.AWOTnMD[1:0] = 00_B)

In interval counter mode the AWOTINT interrupt is generated periodically.

The period is set up via the 32-bit capture/compare register AWOTnCCR, which acts as a compare register in this mode.
If the counter value equals AWOTnCCR the counter is cleared to 0000 0000_H and the AWOTINT is generated.

If the output signal AWOTOUT is enabled (AWOTnTOE = 1), it toggles with each AWOTINT interrupt. Thus AWOTOUT has a cycle duration of the half of the AWOTINT frequency.

AWOTOUT cycle duration = AWOTCLK clock periods × (AWOT0CCR+1) × 2

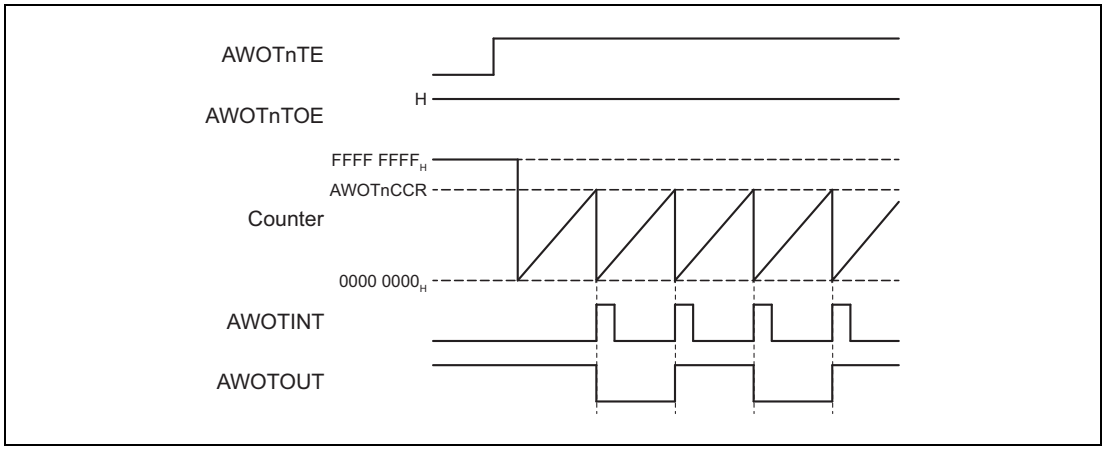


Figure 28.2 Interval timer mode

NOTE

As long as AWOTnTOE = 1, AWOTOUT holds its value while AWOTnTE=0. Again AWOTnTE sets to 1, AWOTOUT starts toggling from its hold value.

Interval timer mode with AWOTnCCR = 0000 0000_H

If the capture/compare register is set to 0000 0000_H, the AWOTINT is asserted and stays active until the AWOT is disabled by AWOTnCTL.AWOTnTE = 0.

28.4.1.2 Capture mode (AWOTnCTL.AWOTnMD[1:0] = 01_B)

In capture mode the number of AWOTCLK cycles between two rising or falling edges of the input signal AWOTIN are counted.

The internal counter starts counting the PCLK clocks from 0000 0000_H. Upon occurrence of a valid AWOTIN edge the counter state is taken over to the AWOTnCCR register and the AWOTINT interrupt is asserted.

Edge selection

The valid AWOTIN edge is selectable:

- AWOTnCTL.AWOTnTIS = 0: falling edge
- AWOTnCTL.AWOTnTIS = 1: rising edge

Capture result

The capture result at each valid AWOTIN edge can be read via the capture/compare register AWOTnCCR, which acts as a capture register in this mode.

In order to avoid conflicts with concurrent counting of the AWOTIN edge and reading of the AWOTnCCR register, AWOTnCCR must not be read while AWOnFLG.AWOTnRPF = 0.
Thus confirm that AWOTnRPF = 1 before reading AWOTnCCR.
After AWOTnCCR reading the AWOTnRPF flag must be cleared by setting AWOTnSTC.AWOTnCLRP = 1.

Count overflow

In case the 32-bit counter overflows the overflow flag AWOTnFLG.AWOTnOVF is set to 1.
This happens at a valid AWOTIN edge, when the counter has reached its maximum value FFFF $FFFF_H$ already before.

With the next valid AWOTIN edge AWOTnOVF is cleared to 0, provided no further overflow did appear in the meantime.

Capture period = AWOTCLK clock periods × ((1 0000 0000_H × AWOT0OVF) + (AWOT0CCR+1))

CAUTIONS

1. **If a second valid AWOTIN edge occurs while the AWOTnRPF flag is 1 (AWOTnCCR read operation not completed) the new capture result will be stored in the internal buffer.**
   The new value will be copied to the AWOTnCCR after the AWOTnRPF was cleared by setting AWOTnSTC.AWOTnCLRP = 1.
2. **As there is only one internal buffer, any succeeding valid AWOTIN edge (while AWOTnRPF flag is still 1) will overwrite the value stored in the internal buffer.**
3. **It takes at least five AWOTCLK clock cycles to store the counter value in the internal buffer and copy it to the AWOTnCCR register (after detecting a valid AWOTIN edge). Thus, in order to ensure proper operation it is necessary that the AWOTCLK clock is at least five times faster than the cycle time of the measured signal. Otherwise the contents of the AWOTnCCR register will become undefined.**

The following diagram illustrates the AWOTn operation in capture mode (valid AWOTIN edge: falling edge).

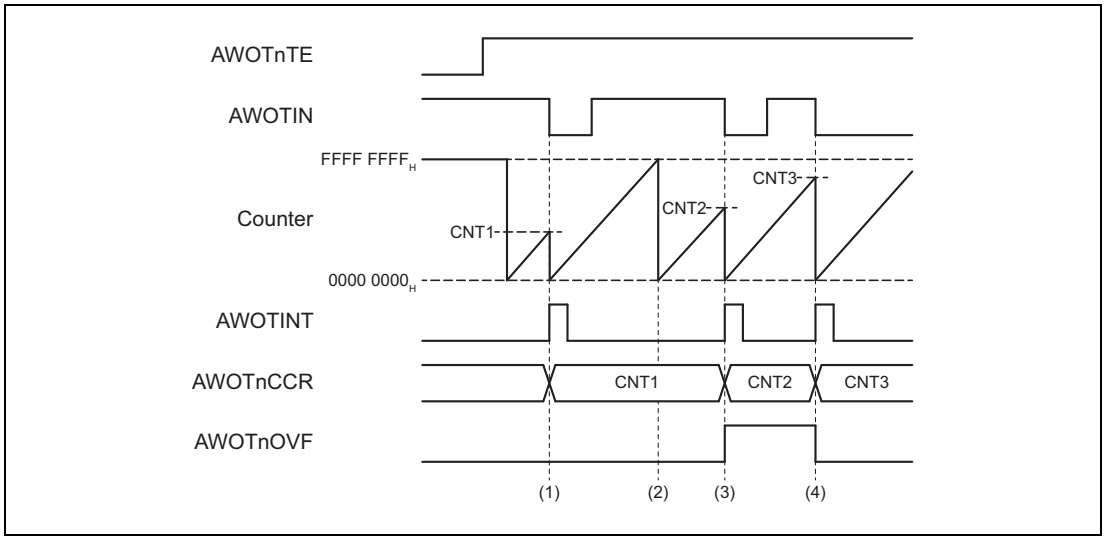


Figure 28.3 Capture mode

(1) Falling AWOTIN edge asserts AWOTINT interrupt and copies counter state CNT1 to AWOTnCCR register, and resets counter to 0000 0000_H.

(2) Counter overflows before next falling AWOTIN edge.

(3) Falling AWOTIN edge copies counter state CNT2 to AWOTnCCR, sets overflow flag AWOTnFLG.AWOTnOVF, and resets counter to 0000 0000_H.

(4) Falling AWOTIN edge copies counter state CNT3 to AWOTnCCR and clears overflow flag AWOTnOVF.

28.4.1.3 PWM mode (AWOTnCTL.AWOTnMD[1:0] = $1x_B$)

In PWM mode the AWOTn can generate a PWM signal at AWOTOUT with 16-bit resolution for the cycle duration and duty cycle.

NOTE

In this section the duty cycle DC is defined by the relation of the output high phase to the PWM cycle duration. Thus a 100% DC means output constant high and 0% DC means output constant low.

The PWM parameters are set via two registers:

- PWM cycle duration = (AWOTnCMP0 + 1) AWOTCLK clock periods
- PWM duty cycle = AWOTnCMP1 AWOTCLK clock periods

The duty cycle DC in % is calculated as follows:

$$DC = \left(1 - \frac{AWOTnCMP1}{AWOTnCMP0 + 1}\right) \times 100$$

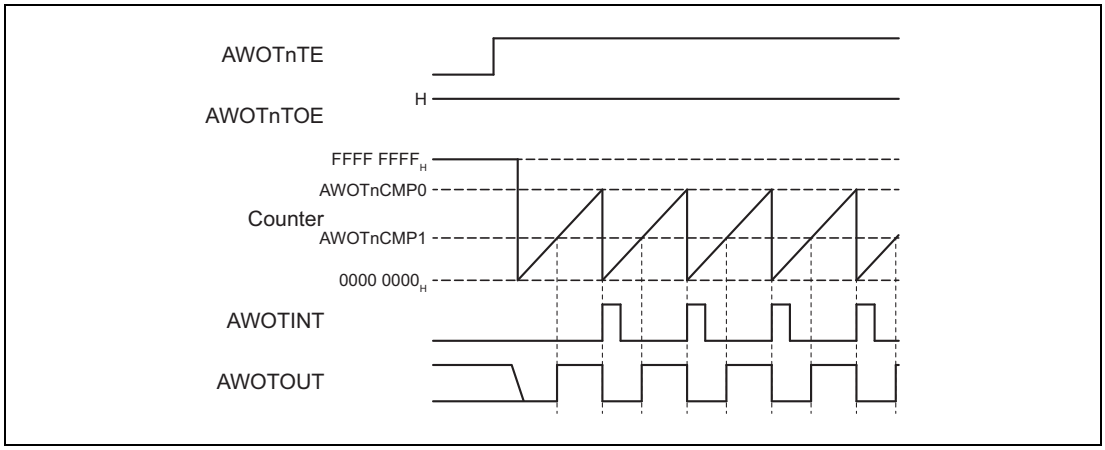


Figure 28.4 PWM mode

PWM with AWOTnCMP0 = 0000_H

Setting AWOTnCMP0 = 0000 0000_H delivers 0 % duty cycle PWM output, i.e. AWOTOUT constant low level.

AWOTINT keeps asserted all the time until AWOTn is disable by AWOTnCTL.AWOTnTE = 0.

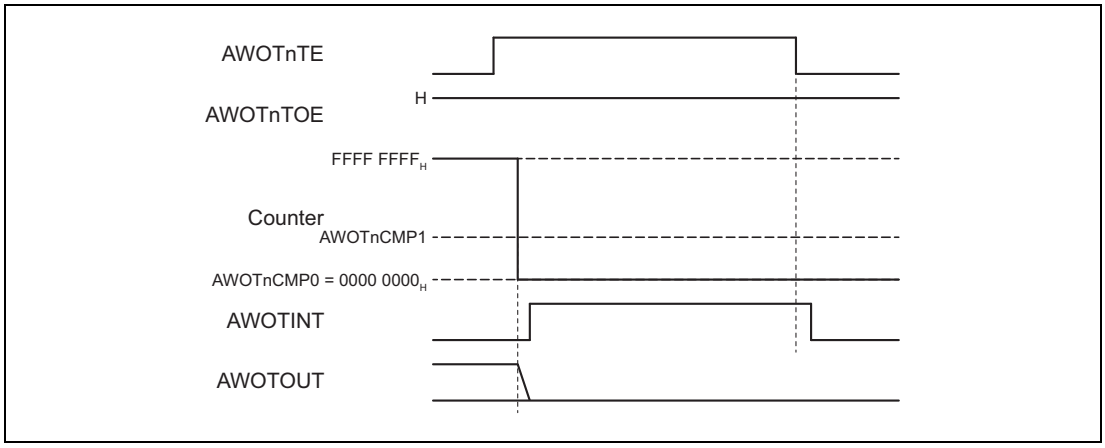


Figure 28.5 PWM mode with AWOTnCMP0 = 0000 0000_H

PWM with 100 % duty cycle

Setting AWOTnCMP1 = 0000 0000_H delivers 100 % duty cycle PWM output, i.e. AWOTOUT constant high level.

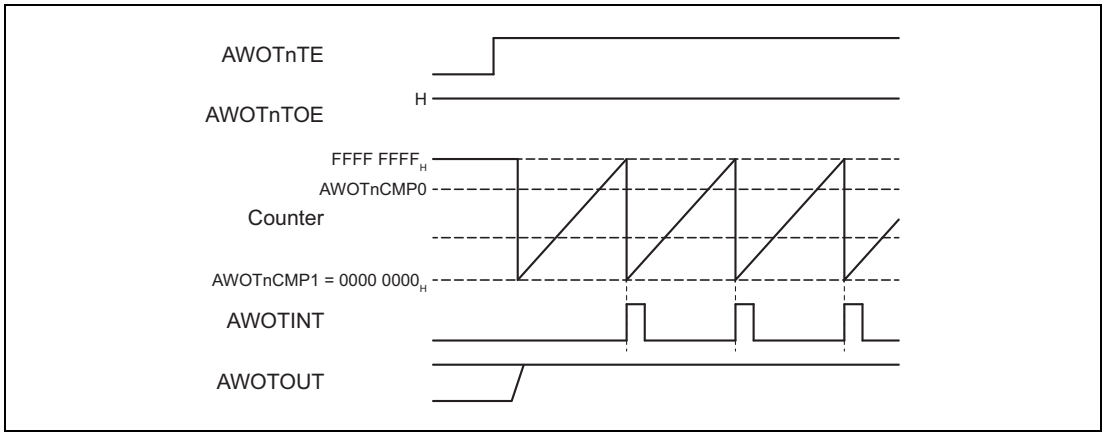


Figure 28.6 PWM mode with 100 % duty cycle

PWM with 0 % duty cycle

Setting AWOTnCMP1 ≥ (AWOTnCMP0 + 1) delivers 0 % duty cycle PWM output, i.e. AWOTOUT constant low level.

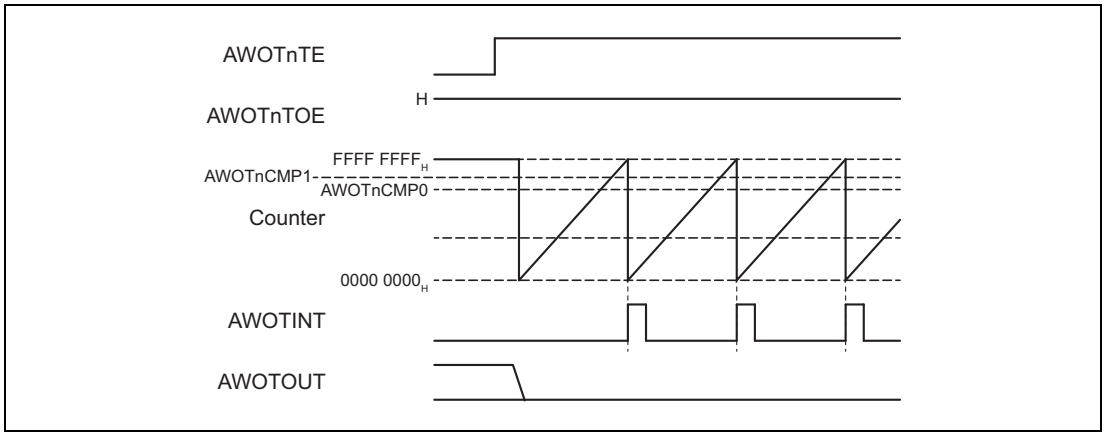


Figure 28.7 PWM mode with 0 % duty cycle

28.4.2 Start and Stop of the AWOTn Operation

28.4.2.1 AWOT disable and registers settings

If the AWOT is disabled (AWOTnCTL.AWOTnTE = 0) all previous register settings become invalid. Thus all registers must be initialized before enabled the AWOT again (AWOTnCTL.AWOTnTE = 1).

28.4.2.2 Start of AWOTn operation

The AWOTn operation is started up properly by the following sequence:

(1) AWOTn must be disabled
All configuration registers must be set up while the AWOTn is disabled, i.e. while AWOTnCTL.AWOTnTE = 0.

(2) Mode dependent registers configuration:
- in interval mode:
interval time AWOTnCCR, enable/disable AWOTOUT by AWOTnTOE
- in PWM mode:
cycle duration AWOTnCMP0, duty cycle AWOTnCMP1, output signal AWOTOUT enable AWOTnTOE = 1.

(3) Set AWOTnCTL register:
- mode selection via AWOTnMD[1:0]
- in capture mode: AWOTIN edge selection via AWOTnTIS
- start operation by AWOTnTE = 1.

Counter start delay

After starting the operation by AWOTnCTL.AWOTnTE = 1 the count operation starts after a delay time of at least three AWOTCLK periods due to synchronization delay of enable signal to AWOTCLK.

28.4.2.3 Stop of AWOTn operation

The AWOTn operation is stopped properly by the following sequence:

(1) Stop operation by AWOTnCTL.AWOTnTE = 0.
After setting AWOTnCTL.AWOTnTE = 0 the AWOTn stops operation after a delay of at least three AWOTCLK periods. Thus make sure to mask the AWOT interrupt before stopping operation.

(2) Disable output signal AWOTOUT (if it was enabled before) by writing AWOTnTOE = 0.
AWOTOUT will output continuous low level.

28.5 Registers

This section contains a description of all registers of the AWOT.

28.5.1 AWOT registers overview

The AWOT is controlled and operated by the following registers:

Table 28.8 AWOT register overview

Register Name	Symbol	Address
Capture compare register	AWOTnCCR	<AWOTn_base>
Compare register 0	AWOTnCMP0	<AWOTn_base> + 04_H
Compare register 1	AWOTnCMP1	<AWOTn_base> + 08_H
Flag register	AWOTnFLG	<AWOTn_base> + $0C_H$
Status clear register	AWOTnSTC	<AWOTn_base> + 10_H
Control register	AWOTnCTL	<AWOTn_base> + 20_H
Output enable register	AWOTnTOE	<AWOTn_base> + 24_H
Emulation register	AWOTnEMU	<AWOTn_base> + 28_H

<AWOTn_base>

The base addresses <AWOTn_base> of the AWOTn is defined in “Register base addresses <AWOTn_base>” in the section above.

28.5.2 AWOTn control registers details

28.5.2.1 AWOTnCTL – Control register

This register is used to enable the AWOT operation, select its operation mode and the valid edge for the capture signal.

Access: This register can be accessed in 8-bit units.

Address: <AWOTn_base> + 20_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	AWOTnTE	–	–	–	AWOTnTIS	–	AWOTnMD[1:0]	
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R/W	R	R/W	R/W

Table 28.9 AWOTnCTL register contents

Bit position	Bit name	Function
7	AWOTnTE	Enable/disable control 0: AWOT disabled 1: AWOT enabled
6 to 4	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset
3	AWOTnTIS	Capture input AWOTIN valid edge selection 0: falling edge 1: rising edge
2	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset
1 to 0	AWOTnMD[1:0]	Operation mode selection 00_B: interval timer mode 01_B: capture mode $1x_B$: PWM mode

NOTES

1. If the AWOT is disabled (AWOTnCTL.AWOTnTE = 0) all previous register settings become invalid.
   Thus all registers must be initialized before enabling the AWOT again (AWOTnCTL.AWOTnTE = 1).
2. After writing the AWOTnTE value it can take up to three AWOTCLK periods until the setting becomes effective.
   The interrupt AWOTINT should be masked before AWOTnTE is set to 0.
3. The output AWOTOUT keeps it's current state after writing AWOTnTE = 0.

28.5.2.2 AWOTnTOE – Output enable register

This register is used to enable or disable the AWOT output signal AWOTOUT in interval and PWM mode.

Access: This register can be accessed in 8-bit units.

Address: <AWOTn_base> + 24_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	AWOTnTOE
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 28.10 AWOTnTOE register contents

Bit position	Bit name	Function
7 to 1	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset
0	AWOTnTOE	Output AWOTOUT control 0: AWOTOUT disabled (fixed to low level) 1: AWOTOUT output enabled

NOTE

The output AWOTOUT is not automatically disabled (fixed to low level) after writing AWOTnTE = 0.

28.5.2.3 AWOTnFLG – Flag register

This register provides various AWOT status information in capture mode.

Access: This register can be accessed in 8-bit units.

Address: <AWOTn_base> + $0C_H$

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	AWOTnRPF	–	–	–	–	–	–	AWOTnOVF
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 28.11 AWOTnFLG register contents

Bit position	Bit name	Function
7	AWOTnRPF	AWOTnCCR read status 0: AWOTNCCR not ready to read 1: AWOTNCCR ready to read To clear AWOTnRPF use the AWOTnSTC register.
6 to 1	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset
0	AWOTnOVF	Counter overflow 0: Counter did not overflow 1: Counter overflow AWOTnOVF is cleared or set with a valid edge of the capture input signal AWOTIN.

28.5.2.4 AWOTnSTC – Status clear register

This register allows to clear the AWOTnCCR read status bit of the flag register AWOTnFLG.

Access: This register can be accessed in 8-bit units.

Address: <AWOTn_base> + 10_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	AWOTnCLRP	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R

Table 28.12 AWOTnSTC register contents

Bit position	Bit name	Function
7	AWOTnCLRP	Clear AWOTnFLG.AWOTnRPF flag 0: no function 1: clear AWOTnFLG.AWOTnRPF
6 to 0	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset

28.5.2.5 AWOTnCCR – Capture compare register

This register

- defines the cycle duration in AWOT interval timer mode
- holds the capture mode result in capture mode

Access: This register can be accessed in 32-bit units.

Address: <AWOTn_base>

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	AWOTnCCR[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AWOTnCCR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.13 AWOTnCCR register contents

Bit position	Bit name	Function
31 to 0	AWOTn CCR[31:0]	The function of this register depends on the AWOTn operation mode, selected by the AWOTnCTL.AWOTnMD[1:0]: • AWOTnMD[1:0] = 00_B: interval timer mode This register defines the interval time in AWOTCLK cycles. • AWOTnMD[1:0] = 01_B: capture mode This register holds the capture result. • AWOTnMD[1:0] = $1x_B$: PWM modes This register has no function.

CAUTIONS

1. **Reading the AWOTnCCR register in capture mode requires to confirm AWOTnFLG.AWOTnRPF = 1.**
   Refer to Section 28.4.1.2, Capture mode (AWOTnCTL.AWOTnMD[1:0] = 01B) for further details.
2. **Writing to this register is prohibited in PWM mode. The AWOT operation is undefined if this register is written when AWOTnMD[1:0] = $1x_B$ (PWM mode).**

NOTE

This register may only be written when the AWOT is disabled (AWOTnCTL.AWOTnTE = 0).

28.5.2.6 AWOTnCMP0 – Compare register 0

This register is used to define the cycle duration of a PWM signal.

Access: This register can be accessed in 16-bit units.

Address: <AWOTn_base> + 04_H

Initial value: 0000_H

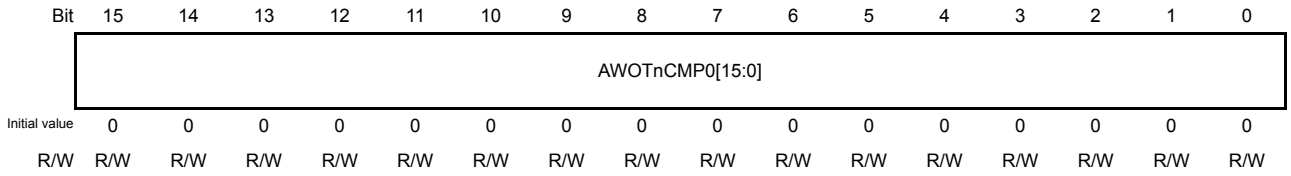


Table 28.14 AWOTnCMP0 register contents

Bit position	Bit name	Function
15 to 0	AWOTn CMP0[15:0]	Cycle duration of a PWM signal. • AWOTnMD[1:0] = $0x_B$: non-PWM modes This register has no function. • AWOTnMD[1:0] = $1x_B$: PWM mode This register defines the PWM cycle duration. PWM cycle duration = (AWOTnCMP0 + 1) AWOTCLK clock periods

CAUTION

Writing to this register is prohibited in any other than PWM mode. The AWOT operation is undefined if this register is written when AWOTnMD[1:0] = $0x_B$ (non PWM mode).

NOTE

This register may only be written when the AWOT is disabled (AWOTnCTL.AWOTnTE = 0).

28.5.2.7 AWOTnCMP1 – Compare register 1

This register is used to define the duty cycle of a PWM signal.

Access: This register can be accessed in 16-bit units.

Address: <AWOTn_base> + 08_H

Initial value: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AWOTnCMP1[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28.15 AWOTnCMP1 register contents

Bit position	Bit name	Function
15 to 0	AWOTn CMP1[15:0]	Duty cycle of a PWM signal. • AWOTnMD[1:0] = $0x_B$: non-PWM modes This register has no function. • AWOTnMD[1:0] = $1x_B$: PWM modes This register defines the PWM duty cycle. PWM cycle duration = AWOTnCMP1 AWOTCLK clock periods

CAUTION

Writing to this register is prohibited in any other than PWM mode. The AWOT operation is undefined if this register is written when AWOTnMD[1:0] = $0x_B$ (non PWM mode).

NOTE

This register may only be written when the AWOT is disabled (AWOTnCTL.AWOTnTE = 0).

28.5.2.8 AWOTnEMU – Emulation register

This register controls whether the AWOTn can be stopped during emulation, for instance upon a breakpoint hit.

Access: This register can be accessed in 8-bit units.

Address: <AWOTn_base> + 28_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	AWOTnSVSDIS	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 28.16 AWOTnEMU register contents

Bit position	Bit name	Function
7	AWOTn SVSDIS	Emulation control 0: AWOTn can be stopped during emulation 1: AWOTn continuous operating during emulation
6 to 0	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset

Section 29 Timer Array Unit B (TAUB)

This section contains a generic description of the timer array unit B (TAUB).

The first part in this section describes all RH850/D1L/D1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the TAUB.

29.1 Features of RH850/D1L/D1M TAUB

29.1.1 Number of Units and Channels

This microcontroller has the following number of TAUB units.

Table 29.1 Number of Units

Product Name	All products
Number of Units	3
Name	TAUBn (n = 0 to 2)

TAUBn has the following number of channels of timers.

Table 29.2 Index

Index	Meaning
n	Throughout this section, the individual TAUB units are identified by the index “n”; for example, TAUBnTOM is the TAUBn channel output mode register.
m	The TAUB has 16 channels. Throughout this section, the individual channels are identified by the index “m” (m = 0 to 15), thus a certain channel is denoted as CHm. The even numbered channels (m = 0, 2, 4, 6, 8, 10, 12, 14) are denoted as CHm_even. The odd numbered channels (m = 1, 3, 5, 7, 9, 11, 13, 15) are denoted as CHm_odd.

29.1.2 Register Base Addresses

TAUBn base addresses are listed in the following table.

TAUBn register addresses are given as offsets from the base addresses in general.

Table 29.3 Register Base Addresses <TAUBn_base>

Name	Base Address
<TAUB0_base>	FFE3 0000_H
<TAUB1_base>	FFE3 1000_H
<TAUB2_base>	FFE3 2000_H

29.1.3 Clock Supply

The TAUBn clock supply is shown in the following table.

Table 29.4 TAUBn Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
TAUB0	PCLK	C_ISO_TAUB01
TAUB1	PCLK	C_ISO_TAUB01
TAUB2	PCLK	C_ISO_TAUB2

29.1.4 Interrupt Requests

TAUBn interrupt requests are listed in the following table.

Table 29.5 Interrupt Requests (1/3)

TAUBn signals	Function	Connected to
TAUB0:		
INTTAUB0I0	Channel 0 interrupt	Interrupt Controller INTTAUB0I0 DMA Controller trigger ID 35
INTTAUB0I1	Channel 1 interrupt	Interrupt Controller INTTAUB0I1 DMA Controller trigger ID 36
INTTAUB0I2	Channel 2 interrupt	Interrupt Controller INTTAUB0I2 DMA Controller trigger ID 37
INTTAUB0I3	Channel 3 interrupt	Interrupt Controller INTTAUB0I3 DMA Controller trigger ID 38
INTTAUB0I4	Channel 4 interrupt	Interrupt Controller INTTAUB0I4 DMA Controller trigger ID 39
INTTAUB0I5	Channel 5 interrupt	Interrupt Controller INTTAUB0I5 DMA Controller trigger ID 40
INTTAUB0I6	Channel 6 interrupt	Interrupt Controller INTTAUB0I6 DMA Controller trigger ID 41
INTTAUB0I7	Channel 7 interrupt	Interrupt Controller INTTAUB0I7 DMA Controller trigger ID 42
INTTAUB0I8	Channel 8 interrupt	Interrupt Controller INTTAUB0I8 DMA Controller trigger ID 43
INTTAUB0I9	Channel 9 interrupt	Interrupt Controller INTTAUB0I9 DMA Controller trigger ID 44
INTTAUB0I10	Channel 10 interrupt	Interrupt Controller INTTAUB0I10 DMA Controller trigger ID 45
INTTAUB0I11	Channel 11 interrupt	Interrupt Controller INTTAUB0I11 DMA Controller trigger ID 46
INTTAUB0I12	Channel 12 interrupt	Interrupt Controller INTTAUB0I12 DMA Controller trigger ID 47
INTTAUB0I13	Channel 13 interrupt	Interrupt Controller INTTAUB0I13 DMA Controller trigger ID 48
INTTAUB0I14	Channel 14 interrupt	Interrupt Controller INTTAUB0I14 DMA Controller trigger ID 49
INTTAUB0I15	Channel 15 interrupt	Interrupt Controller INTTAUB0I15 DMA Controller trigger ID 50

Table 29.5 Interrupt Requests (2/3)

TAUBn signals	Function	Connected to
TAUB1:		
INTTAUB1I0	Channel 0 interrupt	Interrupt Controller INTTAUB1I0 DMA Controller trigger ID 51
INTTAUB1I1	Channel 1 interrupt	Interrupt Controller INTTAUB1I1 DMA Controller trigger ID 52
INTTAUB1I2	Channel 2 interrupt	Interrupt Controller INTTAUB1I2 DMA Controller trigger ID 53
INTTAUB1I3	Channel 3 interrupt	Interrupt Controller INTTAUB1I3 DMA Controller trigger ID 54
INTTAUB1I4	Channel 4 interrupt	Interrupt Controller INTTAUB1I4 DMA Controller trigger ID 55
INTTAUB1I5	Channel 5 interrupt	Interrupt Controller INTTAUB1I5 DMA Controller trigger ID 56
INTTAUB1I6	Channel 6 interrupt	Interrupt Controller INTTAUB1I6 DMA Controller trigger ID 57
INTTAUB1I7	Channel 7 interrupt	Interrupt Controller INTTAUB1I7 DMA Controller trigger ID 58
INTTAUB1I8	Channel 8 interrupt	Interrupt Controller INTTAUB1I8 DMA Controller trigger ID 59
INTTAUB1I9	Channel 9 interrupt	Interrupt Controller INTTAUB1I9 DMA Controller trigger ID 60
INTTAUB1I10	Channel 10 interrupt	Interrupt Controller INTTAUB1I10 DMA Controller trigger ID 61
INTTAUB1I11	Channel 11 interrupt	Interrupt Controller INTTAUB1I11 DMA Controller trigger ID 62
INTTAUB1I12	Channel 12 interrupt	Interrupt Controller INTTAUB1I12 DMA Controller trigger ID 63
INTTAUB1I13	Channel 13 interrupt	Interrupt Controller INTTAUB1I13 DMA Controller trigger ID 64
INTTAUB1I14	Channel 14 interrupt	Interrupt Controller INTTAUB1I14 DMA Controller trigger ID 65
INTTAUB1I15	Channel 15 interrupt	Interrupt Controller INTTAUB1I15 DMA Controller trigger ID 66

Table 29.5 Interrupt Requests (3/3)

TAUBn signals	Function	Connected to
TAUB2:		
INTTAUB2I0	Channel 0 interrupt	Interrupt Controller INTTAUB2I0 DMA Controller trigger ID 67
INTTAUB2I1	Channel 1 interrupt	Interrupt Controller INTTAUB2I1 DMA Controller trigger ID 68
INTTAUB2I2	Channel 2 interrupt	Interrupt Controller INTTAUB2I2 DMA Controller trigger ID 69
INTTAUB2I3	Channel 3 interrupt	Interrupt Controller INTTAUB2I3 DMA Controller trigger ID 70
INTTAUB2I4	Channel 4 interrupt	Interrupt Controller INTTAUB2I4 DMA Controller trigger ID 71
INTTAUB2I5	Channel 5 interrupt	Interrupt Controller INTTAUB2I5 DMA Controller trigger ID 72
INTTAUB2I6	Channel 6 interrupt	Interrupt Controller INTTAUB2I6 DMA Controller trigger ID 73
INTTAUB2I7	Channel 7 interrupt	Interrupt Controller INTTAUB2I7 DMA Controller trigger ID 74
INTTAUB2I8	Channel 8 interrupt	Interrupt Controller INTTAUB2I8 DMA Controller trigger ID 75
INTTAUB2I9	Channel 9 interrupt	Interrupt Controller INTTAUB2I9 DMA Controller trigger ID 76
INTTAUB2I10	Channel 10 interrupt	Interrupt Controller INTTAUB2I10 DMA Controller trigger ID 77
INTTAUB2I11	Channel 11 interrupt	Interrupt Controller INTTAUB2I11 DMA Controller trigger ID 78
INTTAUB2I12	Channel 12 interrupt	Interrupt Controller INTTAUB2I12 DMA Controller trigger ID 79
INTTAUB2I13	Channel 13 interrupt	Interrupt Controller INTTAUB2I13 DMA Controller trigger ID 80
INTTAUB2I14	Channel 14 interrupt	Interrupt Controller INTTAUB2I14 DMA Controller trigger ID 81
INTTAUB2I15	Channel 15 interrupt	Interrupt Controller INTTAUB2I15 DMA Controller trigger ID 82

29.1.5 Reset Sources

TAUBn reset sources are listed in the following table. TAUBn is initialized by these reset sources.

Table 29.6 Reset Sources

Unit Name	Reset Source
TAUBn	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

29.1.6 External I/O Signals

External input/output signals of TAUBn are listed below.

Table 29.7 External I/O Signals (1/3)

TAUB Signals	Function	Alternative Port Pin Signal
TAUB0		
TAUBTTIN0	Channel 0 input	Port TAUB0I0*1*4
TAUBTTIN1	Channel 1 input	Port TAUB0I1*1
TAUBTTIN2	Channel 2 input	Port TAUB0I2*1
TAUBTTIN3	Channel 3 input	Port TAUB0I3*1
TAUBTTIN4	Channel 4 input	Port TAUB0I4*1*4
TAUBTTIN5	Channel 5 input	Port TAUB0I5*1
TAUBTTIN6	Channel 6 input	Port TAUB0I6*1
TAUBTTIN7	Channel 7 input	Port TAUB0I7*1
TAUBTTIN8	Channel 8 input	Port TAUB0I8*1*3
TAUBTTIN9	Channel 9 input	Port TAUB0I9*1
TAUBTTIN10	Channel 10 input	Port TAUB0I10*1
TAUBTTIN11	Channel 11 input	Port TAUB0I11*1
TAUBTTIN12	Channel 12 input	Port TAUB0I12*1*3
TAUBTTIN13	Channel 13 input	Port TAUB0I13*1
TAUBTTIN14	Channel 14 input	Port TAUB0I14*1
TAUBTTIN15	Channel 15 input	Port TAUB0I15*1
TAUBTTOUT0	Channel 0 output	Port TAUB0O0*4
TAUBTTOUT1	Channel 1 output	Port TAUB0O1
TAUBTTOUT2	Channel 2 output	Port TAUB0O2
TAUBTTOUT3	Channel 3 output	Port TAUB0O3
TAUBTTOUT4	Channel 4 output	Port TAUB0O4*4
TAUBTTOUT5	Channel 5 output	Port TAUB0O5
TAUBTTOUT6	Channel 6 output	Port TAUB0O6
TAUBTTOUT7	Channel 7 output	Port TAUB0O7
TAUBTTOUT8	Channel 8 output	Port TAUB0O8*3
TAUBTTOUT9	Channel 9 output	Port TAUB0O9
TAUBTTOUT10	Channel 10 output	Port TAUB0O10
TAUBTTOUT11	Channel 11 output	Port TAUB0O11
TAUBTTOUT12	Channel 12 output	Port TAUB0O12*3
TAUBTTOUT13	Channel 13 output	Port TAUB0O13
TAUBTTOUT14	Channel 14 output	Port TAUB0O14
TAUBTTOUT15	Channel 15 output	Port TAUB0O15
TAUB1		
TAUBTTIN0	Channel 0 input	Port TAUB1I0*1*3
TAUBTTIN1	Channel 1 input	Port TAUB1I1*1
TAUBTTIN2	Channel 2 input	Port TAUB1I2*1
TAUBTTIN3	Channel 3 input	Port TAUB1I3*1
TAUBTTIN4	Channel 4 input	Port TAUB1I4*1*3
TAUBTTIN5	Channel 5 input	Port TAUB1I5*1
TAUBTTIN6	Channel 6 input	Port TAUB1I6*1

Table 29.7 External I/O Signals (2/3)

TAUB Signals	Function	Alternative Port Pin Signal
TAUBTTIN7	Channel 7 input	Port TAUB1I7*1
TAUBTTIN8	Channel 8 input	Port TAUB1I8*1*3
TAUBTTIN9	Channel 9 input	Port TAUB1I9*1*2
TAUBTTIN10	Channel 10 input	Port TAUB1I10*1
TAUBTTIN11	Channel 11 input	Port TAUB1I11*1
TAUBTTIN12	Channel 12 input	Port TAUB1I12*1
TAUBTTIN13	Channel 13 input	Port TAUB1I13*1
TAUBTTIN14	Channel 14 input	Port TAUB1I14*1
TAUBTTIN15	Channel 15 input	Port TAUB1I15*1
TAUBTTOUT0	Channel 0 output	Port TAUB1O0*3
TAUBTTOUT1	Channel 1 output	Port TAUB1O1
TAUBTTOUT2	Channel 2 output	Port TAUB1O2
TAUBTTOUT3	Channel 3 output	Port TAUB1O3
TAUBTTOUT4	Channel 4 output	Port TAUB1O4*3
TAUBTTOUT5	Channel 5 output	Port TAUB1O5
TAUBTTOUT6	Channel 6 output	Port TAUB1O6
TAUBTTOUT7	Channel 7 output	Port TAUB1O7
TAUBTTOUT8	Channel 8 output	Port TAUB1O8*3
TAUBTTOUT9	Channel 9 output	Port TAUB1O9
TAUBTTOUT10	Channel 10 output	Port TAUB1O10
TAUBTTOUT11	Channel 11 output	Port TAUB1O11
TAUBTTOUT12	Channel 12 output	Port TAUB1O12
TAUBTTOUT13	Channel 13 output	Port TAUB1O13
TAUBTTOUT14	Channel 14 output	Port TAUB1O14
TAUBTTOUT15	Channel 15 output	Port TAUB1O15
TAUB2		
TAUBTTIN0	Channel 0 input	Port TAUB2I0*1
TAUBTTIN1	Channel 1 input	Port TAUB2I1*1
TAUBTTIN2	Channel 2 input	Port TAUB2I2*1
TAUBTTIN3	Channel 3 input	Port TAUB2I3*1
TAUBTTIN4	Channel 4 input	Port TAUB2I4*1
TAUBTTIN5	Channel 5 input	Port TAUB2I5*1
TAUBTTIN6	Channel 6 input	Port TAUB2I6*1
TAUBTTIN7	Channel 7 input	Port TAUB2I7*1
TAUBTTIN8	Channel 8 input	Port TAUB2I8*1
TAUBTTIN9	Channel 9 input	Port TAUB2I9*1
TAUBTTIN10	Channel 10 input	Port TAUB2I10*1
TAUBTTIN11	Channel 11 input	Port TAUB2I11*1
TAUBTTIN12	Channel 12 input	Port TAUB2I12*1
TAUBTTIN13	Channel 13 input	Port TAUB2I13*1
TAUBTTIN14	Channel 14 input	Port TAUB2I14*1
TAUBTTIN15	Channel 15 input	Port TAUB2I15*1
TAUBTTOUT0	Channel 0 output	Port TAUB2O0
TAUBTTOUT1	Channel 1 output	Port TAUB2O1

Table 29.7 External I/O Signals (3/3)

TAUB Signals	Function	Alternative Port Pin Signal
TAUBTTOUT2	Channel 2 output	Port TAUB2O2
TAUBTTOUT3	Channel 3 output	Port TAUB2O3
TAUBTTOUT4	Channel 4 output	Port TAUB2O4
TAUBTTOUT5	Channel 5 output	Port TAUB2O5
TAUBTTOUT6	Channel 6 output	Port TAUB2O6
TAUBTTOUT7	Channel 7 output	Port TAUB2O7
TAUBTTOUT8	Channel 8 output	Port TAUB2O8
TAUBTTOUT9	Channel 9 output	Port TAUB2O9
TAUBTTOUT10	Channel 10 output	Port TAUB2O10
TAUBTTOUT11	Channel 11 output	Port TAUB2O11
TAUBTTOUT12	Channel 12 output	Port TAUB2O12
TAUBTTOUT13	Channel 13 output	Port TAUB2O13
TAUBTTOUT14	Channel 14 output	Port TAUB2O14
TAUBTTOUT15	Channel 15 output	Port TAUB2O15 Error Control Module timer input

Note 1. These input signals are passed through a noise filter, refer to the section “Port Filters” in the section “Port Functions”.

Note 2. For availability of these signals, refer to the Section 2.3.2, List of Alternative Function Pins.

Note 3. The TAUB channel below can not use Clock Divide Function, One-Pulse Output Function, TAUBTTINm Input Interval Timer Function among all RH850/D1L/D1M products.
- TAUB0_8, TAUB0_12
- TAUB1_0, TAUB1_4, TAUB1_8

Note 4. The TAUB channel below can not use Clock Divide Function, One-Pulse Output Function, TAUBTTINm Input Interval Timer Function in D1L1, D1L2(H), D1M1(H) and D1M1A.
- TAUB0_0, TAUB0_4

29.1.7 A/D Converter trigger signals

The following TAUBn output signals can be used for triggering an A/D conversion:

Table 29.8 A/D Converter trigger signals

TAUBn unit	Signals	Connected to A/D Converter hardware trigger expansion of scan group			
		SG0	SG1	SG2	SG3
TAUB0	INTTAUB0I2	yes	yes	yes	yes
	INTTAUB0I4	yes	yes	yes	yes
	INTTAUB0I6	yes	yes	yes	yes
	INTTAUB0I8	yes	yes	yes	yes
	INTTAUB0I10	–	–	yes	–
	INTTAUB0I12	–	–	yes	–
	INTTAUB0I14	–	–	yes	–
TAUB1	INTTAUB1I2	yes	yes	yes	yes
	INTTAUB1I4	yes	yes	yes	yes
	INTTAUB1I6	yes	yes	yes	yes
	INTTAUB1I8	yes	yes	yes	yes
	INTTAUB1I10	–	–	yes	–
TAUB2	INTTAUB2I0	yes	yes	–	yes
	INTTAUB2I1	yes	yes	–	yes
	INTTAUB2I2	yes	yes	–	yes
	INTTAUB2I3	yes	yes	–	yes

29.1.8 XOR Compare Unit check of TAUB output signals

The following TAUBn output signals can be checked by the XOR Compare Unit:

- TAUB0: TAUB0O1, TAUB0O3, TAUB0O5, TAUB0O7, TAUB0O9, TAUB0O11, TAUB0O13
- TAUB1: TAUB1O1, TAUB1O3, TAUB1O5, TAUB1O7, TAUB1O9

29.2 Overview

29.2.1 Features Summary

The TAUB has the following functions:

- Independent channel operation function (operated using a single channel)
- Synchronous channel operation function (operated using a master channel and multiple slave channels)

The TAUB is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 16 channels, each equipped with a 16-bit counter TAUBnCNTm and a 16-bit data register TAUBnCDRm to hold the start or compare value of the counter.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate either independently or in combination with other channels (synchronously) When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, it can be operated independent of all other channels. The synchronous operation function is implemented using a combination of channel groups (consisted of master and slave channels).

Several rules apply to the settings of channels.

29.2.2 Terms

In this section, the following terms are used:

- Independent channel operation function/synchronous operation channel operation function

  TAUB has 16 channels, and provides an independent channel operation function that individual channels operate independently and a synchronous channel operation function that is implemented using a combination of channels.

  – The independent channel operation function can use any channel independent of all other channels.
  – The synchronous channel operation function is implemented using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

- Channel group

  In the synchronous channel operation function, all channels that depend on each other are referred to as a “channel group”.
  A channel group has one master channel and one or more slave channels.

- Upper/lower channel

  Depending on the channel number m, a channel with a smaller number or with a larger number is referred to as “upper” or “lower” channel, respectively.

– Upper channel: Channel with a smaller channel number
– Lower channel: Channel with a larger channel number

Example:
For channel 5, channel 3 is an upper channel and channel 9 is a lower channel. Channel 0 is the highest channel and channel 15 is the lowest channel.

The following describes the functional blocks:

Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK3 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- TAUBTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- Operating mode (selected with the TAUBnCMORm.TAUBnMD[4:0] bits)
- Counter start enable (TAUBnTS. TAUBnTSm) and counter stop (TAUBnTT. TAUBnTTm) When counter start is enabled, status flag TAUBnTE. TAUBnTEm is set.
- Count direction (up/down) (can be controlled by the master channel)

Trigger selector

The counter starts automatically when it is enabled (TAUBnTE.TAUBnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger:

- INTTAUBnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUBTTOUTm generation unit.

Simultaneous rewrite controller

Simultaneous rewrite control is a special function that can be used in synchronous operation functions. The data registers of all channels in a channel group (TAUBnCDRm) can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time

TAUBnTO Controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves

29.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 29.9 Functional List of TAUB Operations

Operation Function	Setting Example
Independent Channel Operation Functions	**Section 29.12**
Interval Timer Function	**Section 29.12.1**
TAUBTTINm Input Interval Timer Function	**Section 29.12.2**
Clock Divide Function	**Section 29.12.3**
External Event Count Function	**Section 29.12.4**
One-Pulse Output Function	**Section 29.12.5**
TAUBTTINm Input Pulse Interval Measurement Function	**Section 29.12.6**
TAUBTTINm Input Signal Width Measurement Function	**Section 29.12.7**
TAUBTTINm Input Position Detection Function	**Section 29.12.8**
TAUBTTINm Input Period Count Detection Function	**Section 29.12.9**
TAUBTTINm Input Pulse Interval Judgment Function	**Section 29.12.10**
TAUBTTINm Input Signal Width Judgment Function	**Section 29.12.11**
Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)	**Section 29.12.12**
Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)	**Section 29.12.13**
Independent Channel Simultaneous Rewrite Functions	**Section 29.13**
Simultaneous Rewrite Trigger Generation Function Type 1	**Section 29.13.1**
Synchronous Channel Operation Functions	**Section 29.14**
PWM Output Function	**Section 29.14.1**
One-Shot Pulse Output Function	**Section 29.14.2**
Delay Pulse Output Function	**Section 29.14.3**
AD Conversion Trigger Output Function Type 1	**Section 29.14.4**
Triangle PWM Output Function	**Section 29.14.5**
Triangle PWM Output Function with Dead Time	**Section 29.14.6**
AD Conversion Trigger Output Function Type 2	**Section 29.14.7**

29.2.4 Input/Output Interrupt Request Signals

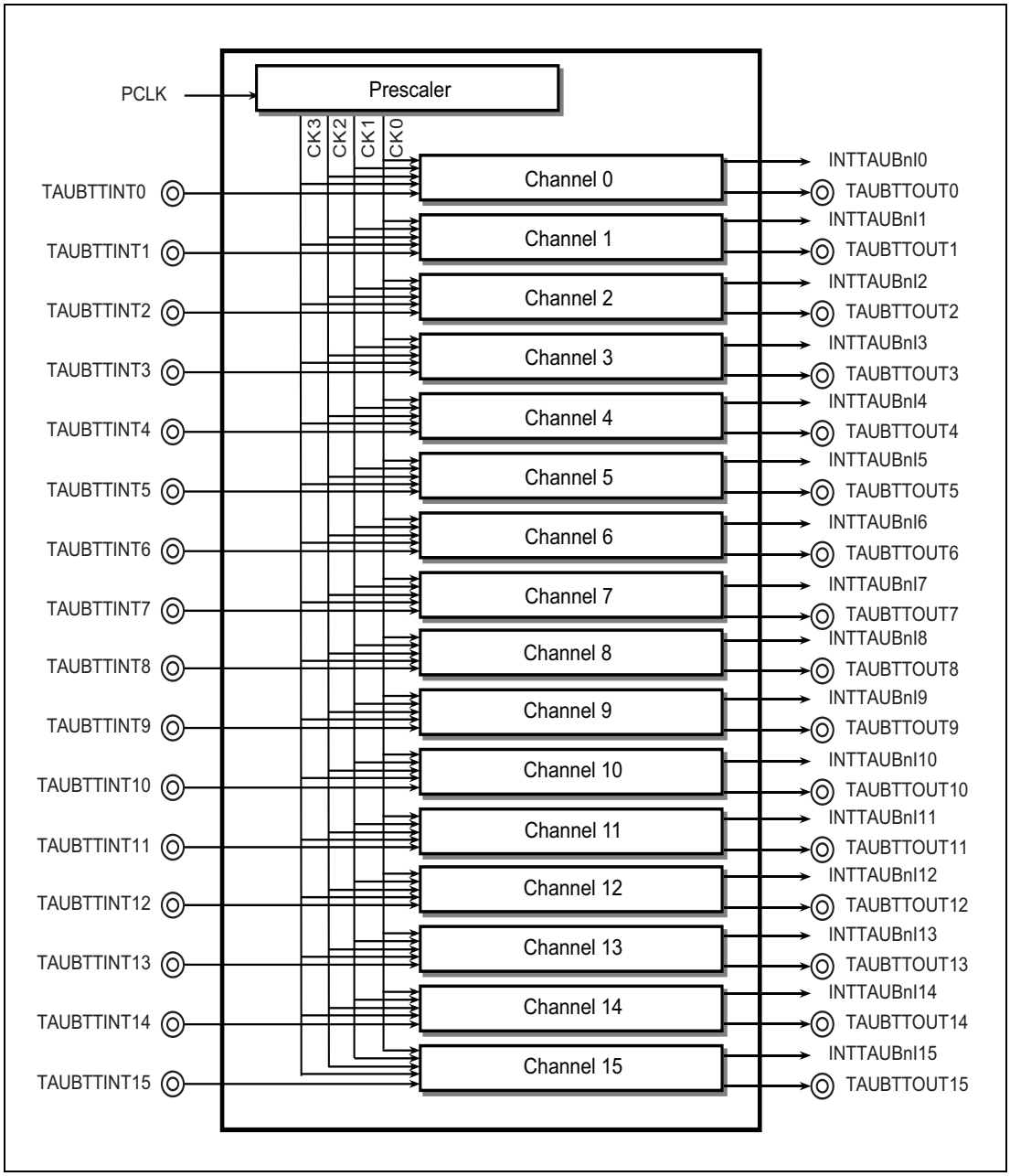


Figure 29.1 TAUB Input/Output and Interrupt Request Signals

29.2.5 Block Diagram

The following figure shows the main components of the TAUB.

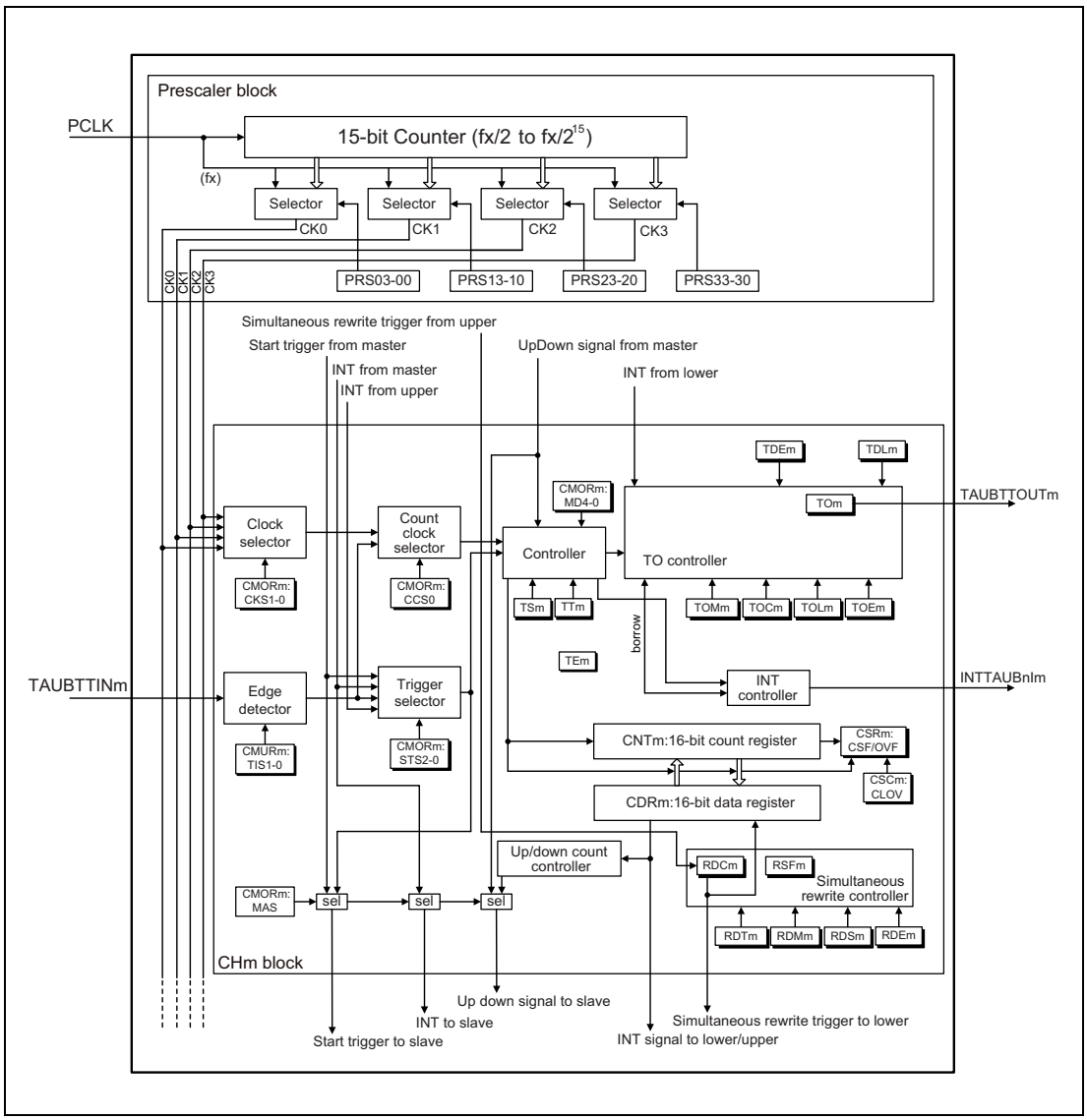


Figure 29.2 Block Diagram of the TAUB

The prefix “TAUBn” has been omitted from the register names for the sake of clarity in the above figure.

29.2.6 Description of Blocks

The following describes the functional blocks:

Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK3 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- TAUBTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- Operation mode (selected by bits TAUBnCMORm.TAUBnMD[4:0])
- Counter start enable (TAUBnTS.TAUBnTSm) and counter stop (TAUBnTT.TAUBnTTm)
  When counter start is enabled, status flag TAUBnTE.TAUBnTEm is set.
- Count direction (up/down) (can be controlled by master channel)

Trigger selector

Depending on the selected operation mode, the counter starts automatically when it is enabled (TAUBnTE.TAUBnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger:

- Effective TAUBTTINm input edge
- INTTAUBnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUBTTOUTm generation unit

Simultaneous rewrite controller

Simultaneous rewrite control is a special function that can be used in synchronous operation functions. The data registers (TAUBnCDRm) of all channels in a channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUBnTO controller

The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves.

29.3 Registers

29.3.1 List of Registers

TAUB registers are listed in the following table.

For details about <TAUBn_base>, see Section 29.1.2, Register Base Addresses.

Table 29.10 List of Registers

Module Name	Register Name	Symbol	Address
TAUBn prescaler registers			
TAUBn	TAUBn prescaler clock select register	TAUBnTPS	<TAUBn_base> + 240_H
TAUBn control registers			
TAUBn	TAUBn channel data register m	TAUBnCDRm	<TAUBn_base> + 0_H + m × 4_H
TAUBn	TAUBn channel counter register m	TAUBnCNTm	<TAUBn_base> + 80_H + m × 4_H
TAUBn	TAUBn channel mode OS register m	TAUBnCMORm	<TAUBn_base> + 200_H + m × 4_H
TAUBn	TAUBn channel mode user register m	TAUBnCMURm	<TAUBn_base> + $C0_H$ + m × 4_H
TAUBn	TAUBn channel status register m	TAUBnCSRm	<TAUBn_base> + 140_H + m × 4_H
TAUBn	TAUBn channel status clear trigger register m	TAUBnCSCm	<TAUBn_base> + 180_H + m × 4_H
TAUBn	TAUBn channel start trigger register	TAUBnTS	<TAUBn_base> + $1C4_H$
TAUBn	TAUBn channel enable status register	TAUBnTE	<TAUBn_base> + $1C0_H$
TAUBn	TAUBn channel stop trigger register	TAUBnTT	<TAUBn_base> + $1C8_H$
TAUBn output registers			
TAUBn	TAUBn channel output enable register	TAUBnTOE	<TAUBn_base> + $5C_H$
TAUBn	TAUBn channel output register	TAUBnTO	<TAUBn_base> + 58_H
TAUBn	TAUBn channel output mode register	TAUBnTOM	<TAUBn_base> + 248_H
TAUBn	TAUBn channel output configuration register	TAUBnTOC	<TAUBn_base> + $24C_H$
TAUBn	TAUBn channel output active level register	TAUBnTOL	<TAUBn_base> + 040_H
TAUBn	TAUBn channel dead time output enable register	TAUBnTDE	<TAUBn_base> + 250_H
TAUBn	TAUBn channel dead time output level register	TAUBnTDL	<TAUBn_base> + 54_H
TAUBn reload data registers			
TAUBn	TAUBn channel reload data enable register	TAUBnRDE	<TAUBn_base> + 260_H
TAUBn	TAUBn channel reload data mode register	TAUBnRDM	<TAUBn_base> + 264_H
TAUBn	TAUBn channel reload data control CH select register	TAUBnRDS	<TAUBn_base> + 268_H
TAUBn	TAUBn channel reload data control register	TAUBnRDC	<TAUBn_base> + $26C_H$
TAUBn	TAUBn channel reload data trigger register	TAUBnRDT	<TAUBn_base> + 44_H
TAUBn	TAUBn channel reload status register	TAUBnRSF	<TAUBn_base> + 48_H
TAUBn emulation register			
TAUBn	TAUBn emulation register	TAUBnEMU	<TAUBn_base> + 290_H

NOTE

In the IO header files, availability of each register depends on whether they can be used by each product. The register has no corresponding function terminal has been removed from the IO header file by 1-bit. For availability of terminals, refer to the Section 2.3.2, List of Alternative Function Pins.

29.3.2 Details of TAUBn Prescaler Registers

29.3.2.1 TAUBnTPS — TAUBn Prescaler Clock Select Register

This register specifies the PCLK predeclares for clocks CK0, CK1, CK2, and CK3 for all channels.

Access: This register can be read/written in 16-bit units.

Address: <TAUBn_base> + 240_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnPRS3[3:0]				TAUBnPRS2[3:0]				TAUBnPRS1[3:0]				TAUBnPRS0[3:0]			
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.11 TAUBnTPS Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 12	TAUBnPRS3 [3:0]	Specifies the CK3 clock.

TAUBnPRS3[3:0]	CK3 clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

These bits can only be rewritten when all counters using CK3 are stopped (TAUBnTE.TAUBnTEm = 0).

Table 29.11 TAUBnTPS Register Contents (2/3)

Bit Position	Bit Name	Function
11 to 8	TAUBnPRS2 [3:0]	Specifies the CK2 clock.

TAUBnPRS2[3:0]	CK2 Clock
0000_B	$PCLK/2^0$
0001_B	$PCLK/2^1$
0010_B	$PCLK/2^2$
0011_B	$PCLK/2^3$
0100_B	$PCLK/2^4$
0101_B	$PCLK/2^5$
0110_B	$PCLK/2^6$
0111_B	$PCLK/2^7$
1000_B	$PCLK/2^8$
1001_B	$PCLK/2^9$
1010_B	$PCLK/2^{10}$
1011_B	$PCLK/2^{11}$
1100_B	$PCLK/2^{12}$
1101_B	$PCLK/2^{13}$
1110_B	$PCLK/2^{14}$
1111_B	$PCLK/2^{15}$

These bits can only be rewritten when all counters using CK2 are stopped (TAUBnTE.TAUBnTEm = 0).

7 to 4	TAUBnPRS1 [3:0]	Specifies the CK1 clock.

TAUBnPRS1[3:0]	CK1 Clock
0000_B	$PCLK/2^0$
0001_B	$PCLK/2^1$
0010_B	$PCLK/2^2$
0011_B	$PCLK/2^3$
0100_B	$PCLK/2^4$
0101_B	$PCLK/2^5$
0110_B	$PCLK/2^6$
0111_B	$PCLK/2^7$
1000_B	$PCLK/2^8$
1001_B	$PCLK/2^9$
1010_B	$PCLK/2^{10}$
1011_B	$PCLK/2^{11}$
1100_B	$PCLK/2^{12}$
1101_B	$PCLK/2^{13}$
1110_B	$PCLK/2^{14}$
1111_B	$PCLK/2^{15}$

These bits can only be rewritten when all counters using CK1 are stopped (TAUBnTE.TAUBnTEm = 0).

Table 29.11 TAUBnTPS Register Contents (3/3)

Bit Position	Bit Name	Function
3 to 0	TAUBnPRS0 [3:0]	Specifies the CK0 clock.

TAUBnPRS0[3:0]	CK0 Clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

These bits can only be rewritten when all counters using CK0 are stopped (TAUBnTE.TAUBnTEm = 0).

NOTE

The TAUBn clock input PCLK is specified in the first part of this section, Section 29.1.3, Clock Supply.

29.3.3 Details of TAUBn Control Registers

29.3.3.1 TAUBnCDRm — TAUBn Channel Data Register

This register functions either as a compare register or as a capture register, depending on the operation mode specified in TAUBnCMORm.TAUBnMD[4:1].

Access: This register can be read/written in 16-bit units.
- When this register functions as a capture register, only reading is possible. Write operation is ignored.
- When this register functions as a compare register, reading and writing is possible.

Address: <TAUBn_base> + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.12 TAUBnCDRm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnCDR [15:0]	Data register for the capture/compare value.

29.3.3.2 TAUBnCNTm — TAUBn Channel Counter Register

This register is the channel m counter register.

Access: This register can only be read in 16-bit units.

Address: <TAUBn_base> + 80_H + m × 4_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.13 TAUBnCNTm Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnCNT [15:0]	16-bit counter value.

The read value depends on the counter, the operation mode change, and the values of the TAUBnTS.TAUBnTSm and TAUBnTT.TAUBnTTm bits.

The initial counter read value depends on the operation mode and how the counter was stopped:

- by a reset
- by a counter stop trigger (TAUBnTT.TAUBnTTm = 1)

The following table lists the initial counter read values after the counter has stopped (TAUBnTE.TAUBnTEm = 0) and re-enabled (TAUBnTS.TAUBnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUBnTS.TAUBnTSm = 1) for modes where the counter waits for a start trigger.

Table 29.14 TAUBnCNTm Read Values after Re-Enabling Counter

		TAUBnCNTm		
Mode Name	Count Method (Up/Down)	Start Value*1	After Stop Trigger	After One Count
Interval timer mode	Count down	$FFFF_H$	Stop value	—
Judge mode	Count down	$FFFF_H$	Stop value	—
Capture mode	Count up	0000_H	Stop value	—
Event count mode	Count down	$FFFF_H$	Stop value	—
One-count mode	Count down	$FFFF_H$	Stop value	Stop value
Capture and one-count mode	Count up	0000_H	Stop value	Capture value + 1 (TAUBnCDRm)
Judge and one-count mode	Count down	$FFFF_H$	Stop value	TAUBnCNTm value – 1
Count-up/-down mode	Count up/down	$FFFF_H$	Stop value	—
Pulse one-count mode	Count down	$FFFF_H$	Stop value	0000_H
Count capture mode	Count up	0000_H	Stop value	—
Gate count mode	Count down	$FFFF_H$	Stop value	Stop value
Capture and gate count mode	Count up	0000_H	Stop value	Stop value

Note 1. The value set for TAUBnCNTm when operation mode is changed after reset release

29.3.3.3 TAUBnCMORm — TAUBn Channel Mode OS Register

This register controls channel m operation.

Access: This register can be read/written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

Address: <TAUBn_base> + 200_H + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.15 TAUBnCMORm Register Contents (1/3)

Bit Position	Bit Name	Function
15, 14	TAUBnCKS [1:0]	Selects the operation clock. The operation clock is used for the TAUBTTINm input edge detection circuit. It can also be used as the count clock of TAUBnCNTm depending on bit TAUBnCMORm.TAUBnCCS0. **TAUBn CKS1 / TAUBn CKS0 / Selected Operation Clock** 0 / 0 / CK0 0 / 1 / CK1 1 / 0 / CK2 1 / 1 / CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Selects the count clock for TAUBnCNTm counter: **TAUBnCCS0 / Selected Count Clock** 0 / Operation clock as specified by TAUBnCMORm.TAUBnCKS[1:0]. 1 / Valid edge of TAUBTTINm input signal
11	TAUBnMAS	Specifies the channel as master or slave channel during synchronous channel operation: 0: Slave 1: Master This bit is only valid for even channels (CHm_even). For odd channels (CHm_odd), it is fixed to 0.

Table 29.15 TAUBnCMORm Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUBnSTS [2:0]</td><td>Selects the external start trigger:
<table>
<tr><th>TAUBn STS2</th><th>TAUBn STS1</th><th>TAUBn STS0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of the TAUBTTINm input signal. TAUBnCMURm.TAUBnTIS[1:0] specifies the valid edge.</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of the TAUBTTINm input signal is the tart trigger and the reverse edge is the stop (capture) trigger</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INTTAUBnIm is the start trigger of master channel</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>INTTAUBnIm of upper channel (m – 1) is the start trigger regardless of master setting</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Dead-time output signal of the TAUBTTOUTm generation unit</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Up/down output trigger signal of the master channel.</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>TAUBnCOS [1:0]</td><td>Specifies when the capture register TAUBnCDRm and the overflow flag TAUBnCSRm.TAUBnOVF of channel m are updated.
These bits are only valid if channel m is in capture function (capture mode and capture & one-count mode).
<table>
<tr><th>TAUBn COS1</th><th>TAUBn COS0</th><th>TAUBnCDRm</th><th>TAUBnCSRm.TAUBnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated upon detection of a TAUBTTINm input valid edge.</td><td>Updated (cleared or set) upon detection of a TAUBTTINm input valid edge:
• If a counter overflow has occurred since the last valid edge detection, TAUBnCSRm.TAUBnOVF is set.
• If no counter overflow has occurred since the last valid edge detection, TAUBnCSRm.TAUBnOVF is cleared.</td></tr>
<tr><td>0</td><td>1</td><td>Set upon counter overflow and cleared by a TAUBnCSCm.TAUBnCLOV = 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated upon detection of a TAUBTTINm input valid edge and upon counter overflow:
• TAUBTTINm input valid edge: Counter value is written to TAUBnCDRm
• Overflow: $FFFF_H$ is written to TAUBnCDRm. The next TAUBTTINm input valid edge detection is ignored.</td><td>Not set.</td></tr>
<tr><td>1</td><td>1</td><td>Set upon counter overflow and cleared by a TAUBnCSCm.TAUBnCLOV = 1.</td></tr>
</table></td></tr>
<tr><td>5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing to this bit, write the value after reset.</td></tr>
</table>

Table 29.15 TAUBnCMORm Register Contents (3/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>4 to 0</td><td>TAUBnMD [4:0]</td><td>Specifies the operation mode. For details, refer to the settings for individual functions.
<table>
<tr><th>TAUBn MD4</th><th>TAUBn MD3</th><th>TAUBn MD2</th><th>TAUBn MD1</th><th>TAUBn MD0</th><th>Functional Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1/0</td><td>Interval timer mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1/0</td><td>Judge mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Capture mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>Event count mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1/0</td><td>One-count mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>Capture and one-count mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1/0</td><td>Judge and one-count mode</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>Count-up/-down mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Pulse one-count mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1/0</td><td>Count capture mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>Gate count mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>Capture and gate count mode</td></tr>
<tr><td colspan="5">Others</td><td>Setting prohibited</td></tr>
</table>
<table>
<tr><th>Mode</th><th>Role of TAUBnMD0 Bit</th></tr>
<tr><td>Interval timer mode
Capture mode
Count capture mode</td><td>Specifies whether INTTAUBnIm is output at the beginning of count operation (when a start trigger is entered) or not.
0: INTTAUBnIm is not output.
1: INTTAUBnIm is output.</td></tr>
<tr><td>Event count mode
Count-up/-down mode</td><td>This bit should be set to 0 (INTTAUBnIm signal is not output at the beginning of count operation).</td></tr>
<tr><td>One-count mode
Pulse one-count mode</td><td>Enables/disables start trigger detection during counting.
0: Disables detection.
1: Enables detection.
CAUTION
• In one-count mode, INTTAUBnIm signal is not output at the beginning of count operation.
• In pulse one-count mode, INTTAUBnIm signal is output at the beginning of count operation.</td></tr>
<tr><td>Gate count mode</td><td>This bit should be set to 0 (start trigger detection during counting is disabled).</td></tr>
<tr><td>Capture and one-count mode
Capture and gate count mode</td><td>This bit should be set to 0.
CAUTION
INTTAUBnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.</td></tr>
<tr><td>Judge mode
Judge and one-count mode</td><td>Specifies INTTAUBnIm output timing.
0: When TAUBnCNTm ≤ TAUBnCDRm
1: When TAUBnCNTm > TAUBnCDRm</td></tr>
</table>
</td></tr>
</table>

29.3.3.4 TAUBnCMURm — TAUBn Channel Mode User Register

This register specifies the type of valid edge detection used for the TAUBTTINm input.

Access: This register can be read/written in 8-bit units.

Address: <TAUBn_base> + $C0_H$ + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.16 TAUBnCMURm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS [1:0]	Specifies the valid edge of the TAUBTTINm input:

TAUBn TIS1	TAUBn TIS0	Description
0	0	Falling edge
0	1	Rising edge
1	0	Rising and falling edges (low-width measurement selection). Start trigger: falling edge Stop trigger (capture): rising edge
1	1	Rising and falling edges (high-width measurement selection). Start trigger: rising edge Stop trigger (capture): falling edge

- Edge detection for TAUBTTINm input signals is performed based on the operation clock selected by TAUBnCMORm.TAUBnCKS[1:0].

29.3.3.5 TAUBnCSRm — TAUBn Channel Status Register

This register indicates the count direction and the overflow status of channel m's counter.

Access: This register can only be read in 8-bit units.

Address: <TAUBn_base> + 140_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnCSF	TAUBnOVF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 29.17 TAUBnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	TAUBnCSF	Indicates the count direction: 0: Counts up 1: Counts down The read value of this bit is only valid in the following mode: • Up Down Count mode
0	TAUBnOVF	Indicates the counter overflow status: 0: No overflow occurred 1: Overflow occurred This bit is used only in the following modes: • Capture mode • Capture and one-count mode The function of this bit depends on the setting of control bits TAUBnCMORm.TAUBnCOS[1:0].

29.3.3.6 TAUBnCSCm — TAUBn Channel Status Clear Register

This register is a trigger register for clearing the overflow flag TAUBnCSRm.TAUBnOVF of a channel m.

Access: This register can only be written in 8-bit units. It is always read as 00_H

Address: <TAUBn_base> + 180_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUBnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 29.18 TAUBnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	TAUBnCLOV	0: No function 1: Clears the overflow flag TAUBnCSRm.TAUBnOVF

29.3.3.7 TAUBnTS — TAUBn Channel Start Trigger Register

This register enables the counter for each channel.

Access: This register can only be written in 16-bit units. It is always read as 0000_H

Address: <TAUBn_base> + $1C4_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnTS15	TAUBnTS14	TAUBnTS13	TAUBnTS12	TAUBnTS11	TAUBnTS10	TAUBnTS09	TAUBnTS08	TAUBnTS07	TAUBnTS06	TAUBnTS05	TAUBnTS04	TAUBnTS03	TAUBnTS02	TAUBnTS01	TAUBnTS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 29.19 TAUBnTS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTSm	Enables the counter for channel m: 0: No function 1: Enables the counter and sets TAUBnTE.TAUBnTEm = 1. TAUBnTE.TAUBnTEm = 1 only enables counter. Whether the counter starts depends on the selected operation mode.

Note that index m is representing a double-digit number for this register. See above.

29.3.3.8 TAUBnTE — TAUBn Channel Enable Status Register

This register indicates whether counter is enabled or disabled.

Access: This register can only be read in 16-bit units.

Address: <TAUBn_base> + $1C0_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnTE15	TAUBnTE14	TAUBnTE13	TAUBnTE12	TAUBnTE11	TAUBnTE10	TAUBnTE09	TAUBnTE08	TAUBnTE07	TAUBnTE06	TAUBnTE05	TAUBnTE04	TAUBnTE03	TAUBnTE02	TAUBnTE01	TAUBnTE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.20 TAUBnTE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTEm	Indicates whether counter for channel m is enabled or disabled: 0: Counter disabled 1: Counter enabled Setting TAUBnTS.TAUBnTSm to 1 sets this bit to 1. Setting TAUBnTT.TAUBnTTm to 1 resets this bit to 0.

Note that index m is representing a double-digit number for this register. See above.

29.3.3.9 TAUBnTT — TAUBn Channel Stop Trigger Register

This register stops the counter for each channel.

Access: This register can only be written in 16-bit units. It is always read as 0000_H.

Address: <TAUBn_base> + $1C8_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnTT15	TAUBnTT14	TAUBnTT13	TAUBnTT12	TAUBnTT11	TAUBnTT10	TAUBnTT09	TAUBnTT08	TAUBnTT07	TAUBnTT06	TAUBnTT05	TAUBnTT04	TAUBnTT03	TAUBnTT02	TAUBnTT01	TAUBnTT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 29.21 TAUBnTT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTTm	Stops the counter of channel m: 0: No function 1: Stops the counter and resets TAUBnTE.TAUBnTEm = 0. TAUBnCNTm, TAUBnTO.TAUBnTOm, and TAUBTTOUTm all retain the values they had before the counter was stopped.

Note that index m is representing a double-digit number for this register. See above.

29.3.4 Details of TAUBn Simultaneous Rewrite Registers

29.3.4.1 TAUBnRDE — TAUBn Channel Reload Data Enable Register

This register enables and disables simultaneous rewrite of the data register TAUBnCDRm/TAUBnTOLm.

Access: This register can be read/written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0.

Address: <TAUBn_base> + 260_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDE15	TAUBn RDE14	TAUBn RDE13	TAUBn RDE12	TAUBn RDE11	TAUBn RDE10	TAUBn RDE09	TAUBn RDE08	TAUBn RDE07	TAUBn RDE06	TAUBn RDE05	TAUBn RDE04	TAUBn RDE03	TAUBn RDE02	TAUBn RDE01	TAUBn RDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.22 TAUBnRDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDEm	Enables/disables simultaneous rewrite of the data register of channel m: 0: Disables simultaneous rewrite 1: Enabled simultaneous rewrite

Note that index m is representing a double-digit number for this register. See above.

29.3.4.2 TAUBnRDS — TAUBn Channel Reload Data Control Channel Select Register

This register selects the control channel for simultaneous rewrite.

Access: This register can be read/written in 16-bit. It can only be written when TAUBnTE.TAUBnTEm = 0.

Address: <TAUBn_base> + 268_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDS15	TAUBn RDS14	TAUBn RDS13	TAUBn RDS12	TAUBn RDS11	TAUBn RDS10	TAUBn RDS09	TAUBn RDS08	TAUBn RDS07	TAUBn RDS06	TAUBn RDS05	TAUBn RDS04	TAUBn RDS03	TAUBn RDS02	TAUBn RDS01	TAUBn RDS00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.23 TAUBnRDS Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDSm	Specifies which channel is controlled for the simultaneous rewrite trigger: 0: Master channel 1: Another upper channel

Note that index m is representing a double-digit number for this register. See above.

29.3.4.3 TAUBnRDM — TAUBn Channel Reload Data Mode Register

This register selects when the signal that controls simultaneous rewrite is loaded.

Access: This register can be read/written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0.

Address: <TAUBn_base> + 264_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDM15	TAUBn RDM14	TAUBn RDM13	TAUBn RDM12	TAUBn RDM11	TAUBn RDM10	TAUBn RDM09	TAUBn RDM08	TAUBn RDM07	TAUBn RDM06	TAUBn RDM05	TAUBn RDM04	TAUBn RDM03	TAUBn RDM02	TAUBn RDM01	TAUBn RDM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.24 TAUBnRDM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDMm	Selects when the signal that triggers simultaneous is generated: 0: When the master channel counter starts counting 1: At the top of a triangle wave cycle These bits only apply when TAUBnRDE.TAUBnRDEm = 1 and TAUBnRDS.TAUBnRDSm = 0.

Note that index m is representing a double-digit number for this register. See above.

29.3.4.4 TAUBnRDC — TAUBn Channel Reload Data Control Register

This register specifies the channel that generates the INTTAUBnIm signal that triggers simultaneous rewrite.

Access: This register can be read/written in 16-bit units. It can only be written when TAUBnTE.TAUBnTEm = 0

Address: <TAUBn_base> + $26C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDC15	TAUBn RDC14	TAUBn RDC13	TAUBn RDC12	TAUBn RDC11	TAUBn RDC10	TAUBn RDC09	TAUBn RDC08	TAUBn RDC07	TAUBn RDC06	TAUBn RDC05	TAUBn RDC04	TAUBn RDC03	TAUBn RDC02	TAUBn RDC01	TAUBn RDC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.25 TAUBnRDC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDCm	Specifies whether the channel generates a simultaneous rewrite trigger signal or not. 0: Does not operate as a simultaneous rewrite trigger channel. 1: Operates as a simultaneous rewrite trigger channel. These bits only apply when TAUBnRDE.TAUBnRDEm = 1 and TAUBnRDS.TAUBnRDSm = 1.

Note that index m is representing a double-digit number for this register. See above.

29.3.4.5 TAUBnRDT — TAUBn Channel Reload Data Trigger Register

This register triggers the simultaneous rewrite enabling state.

Access: This register can only be written in 16-bit units. It is always read as 0000_H.

Address: <TAUBn_base> + 044_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RDT15	TAUBn RDT14	TAUBn RDT13	TAUBn RDT12	TAUBn RDT11	TAUBn RDT10	TAUBn RDT09	TAUBn RDT08	TAUBn RDT07	TAUBn RDT06	TAUBn RDT05	TAUBn RDT04	TAUBn RDT03	TAUBn RDT02	TAUBn RDT01	TAUBn RDT00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 29.26 TAUBnRDT Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRDTm	Triggers the simultaneous rewrite enabling state: 0: No function. The operation writing 0 is ignored. 1: The simultaneous rewrite enabling flag (TAUBnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger. These bits only apply when: • TAUBnRDE.TAUBnRDEm = 1

Note that index m is representing a double-digit number for this register. See above.

29.3.4.6 TAUBnRSF — TAUBn Channel Reload Status Register

This flag register indicates the simultaneous rewrite status.

Access: This register can only be read in 16-bit units.

Address: <TAUBn_base> + 048_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn RSF15	TAUBn RSF14	TAUBn RSF13	TAUBn RSF12	TAUBn RSF11	TAUBn RSF10	TAUBn RSF09	TAUBn RSF08	TAUBn RSF07	TAUBn RSF06	TAUBn RSF05	TAUBn RSF04	TAUBn RSF03	TAUBn RSF02	TAUBn RSF01	TAUBn RSF00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 29.27 TAUBnRSF Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnRSFm	Indicates the simultaneous rewrite status: 0: Indicates simultaneous rewrite is completed due to the generation of the simultaneous rewrite trigger. 1: Indicates the simultaneous rewrite trigger waiting state when simultaneous rewrite is enabled (TAUBnRDTm = 1).

Note that index m is representing a double-digit number for this register. See above.

29.3.5 Details of TAUBn Output Registers

29.3.5.1 TAUBnTOE — TAUBn Channel Output Enable Register

This register enables and disables independent channel output mode controlled by software.

Access: This register can be read/written in 16-bit units.

Address: <TAUBn_base> + $5C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOE15	TAUBn TOE14	TAUBn TOE13	TAUBn TOE12	TAUBn TOE11	TAUBn TOE10	TAUBn TOE09	TAUBn TOE08	TAUBn TOE07	TAUBn TOE06	TAUBn TOE05	TAUBn TOE04	TAUBn TOE03	TAUBn TOE02	TAUBn TOE01	TAUBn TOE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.28 TAUBnTOE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOEm	Enables/disables independent channel output mode: 0: Disables independent channel output mode (controlled by software) 1: Enables independent channel output mode

Note that index m is representing a double-digit number for this register. See above.

29.3.5.2 TAUBnTO — TAUBn Channel Output Register

This register specifies and reads the level of TAUBTTOUTm.

Access: This register can be read/written in 16-bit units.

Address: <TAUBn_base> + 58_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TO15	TAUBn TO14	TAUBn TO13	TAUBn TO12	TAUBn TO11	TAUBn TO10	TAUBn TO09	TAUBn TO08	TAUBn TO07	TAUBn TO06	TAUBn TO05	TAUBn TO04	TAUBn TO03	TAUBn TO02	TAUBn TO01	TAUBn TO00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.29 TAUBnTO Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOm	Specifies/reads the level of TAUBTTOUTm: 0: Low 1: High Only TAUBnTOm bits for which Independent Channel Output function is disabled (TAUBnTOEm = 0) can be written.

Note that index m is representing a double-digit number for this register. See above.

29.3.5.3 TAUBnTOM — TAUBn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read/written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

Address: <TAUBn_base> + 248_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOM15	TAUBn TOM14	TAUBn TOM13	TAUBn TOM12	TAUBn TOM11	TAUBn TOM10	TAUBn TOM09	TAUBn TOM08	TAUBn TOM07	TAUBn TOM06	TAUBn TOM05	TAUBn TOM04	TAUBn TOM03	TAUBn TOM02	TAUBn TOM01	TAUBn TOM00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.30 TAUBnTOM Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOMm	Specifies the channel output mode: 0: Independent channel output mode 1: Synchronous channel output mode

Note that index m is representing a double-digit number for this register. See above.

29.3.5.4 TAUBnTOC — TAUBn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUBnTOMm.

Access: This register can be read/written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

Address: <TAUBn_base> + $24C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TOC15	TAUBn TOC14	TAUBn TOC13	TAUBn TOC12	TAUBn TOC11	TAUBn TOC10	TAUBn TOC09	TAUBn TOC08	TAUBn TOC07	TAUBn TOC06	TAUBn TOC05	TAUBn TOC04	TAUBn TOC03	TAUBn TOC02	TAUBn TOC01	TAUBn TOC00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.31 TAUBnTOC Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOCm	Specifies the output mode: 0: Operation mode 1 1: Operation mode 2 The output mode also depends on TAUBnTOM.TAUBnTOMm, as can be seen in the following table.

TOMm	TOCm	Description
0	0	Toggle mode: TAUBTTOUTm toggles when INTTAUBnIm occurs.
	1	Set/reset mode: TAUBTTOUTm set when INTTAUBnIm occurs upon count start and reset when INTTAUBnIm occurs due to detection of a match between TAUBnCNTm and TAUBnCDRm.
1	0	Synchronous Channel Operation Mode 1: TAUBTTOUTm set when INT occurs on the master channel and reset when INT occurs on the slave channel.
	1	Synchronous Channel Operation Mode 2: TAUBTTOUTm set when INTTAUBnIm occurs while the slave channel is counting down and reset when INTTAUBnIm occurs while the slave channel is counting up

Note that index m is representing a double-digit number for this register. See above.

29.3.5.5 TAUBnTOL — TAUBn Channel Output Level Register

This register specifies the output logic of the channel output bit (TAUBnTO.TAUBnTOm).

Access: This register can be read/written in 16-bit units.

Address: <TAUBn_base> + 040_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnTOL15	TAUBnTOL14	TAUBnTOL13	TAUBnTOL12	TAUBnTOL11	TAUBnTOL10	TAUBnTOL09	TAUBnTOL08	TAUBnTOL07	TAUBnTOL06	TAUBnTOL05	TAUBnTOL04	TAUBnTOL03	TAUBnTOL02	TAUBnTOL01	TAUBnTOL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.32 TAUBnTOL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTOLm	Specifies the output logic of the channel m output bit (TAUBnTO.TAUBnTOm): 0: Positive logic (active high) 1: Negative logic (active low) The setting of these bits applies to all channel output modes other than independent channel output mode controlled by software and independent channel output mode 1.

Note that index m is representing a double-digit number for this register. See above.

29.3.6 Details of TAUBn Dead Time Output Registers

29.3.6.1 TAUBnTDE — TAUBn Channel Dead Time Output Enable Register

This register enables/disables dead time operation for each channel.

Access: This register can be read/written in 16-bit units. It can only be written when the counter is stopped (TAUBnTE.TAUBnTEm = 0).

Address: <TAUBn_base> + 250_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TDE15	TAUBn TDE14	TAUBn TDE13	TAUBn TDE12	TAUBn TDE11	TAUBn TDE10	TAUBn TDE09	TAUBn TDE08	TAUBn TDE07	TAUBn TDE06	TAUBn TDE05	TAUBn TDE04	TAUBn TDE03	TAUBn TDE02	TAUBn TDE01	TAUBn TDE00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.33 TAUBnTDE Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTDEm	Enables/disables dead time control operation of channel m: 0: Disables dead time operation 1: Enables dead time operation The same settings must be set for the even and the odd slave channel that comprise a set. These bits only apply when: • TAUBnTOE.TAUBnTOEm, TAUBnTOM.TAUBnTOMm, TAUBnTOC.TAUBnTOCm = 1

Note that index m is representing a double-digit number for this register. See above.

29.3.6.2 TAUBnTDL — TAUBn Channel Dead Time Output Level Register

This register selects the phase period to which dead time is added.

Access: This register can be read/written in 16-bit units.

Address: <TAUBn_base> + 54_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBn TDL15	TAUBn TDL14	TAUBn TDL13	TAUBn TDL12	TAUBn TDL11	TAUBn TDL10	TAUBn TDL09	TAUBn TDL08	TAUBn TDL07	TAUBn TDL06	TAUBn TDL05	TAUBn TDL04	TAUBn TDL03	TAUBn TDL02	TAUBn TDL01	TAUBn TDL00
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29.34 TAUBnTDL Register Contents

Bit Position	Bit Name	Function
15 to 0	TAUBnTDLm	Selects the phase period to which dead time is added: 0: Positive phase period 1: Negative phase period These bits only apply when: • TAUBnTOE.TAUBnTOEm, TAUBnTOM.TAUBnTOMm, TAUBnTOC.TAUBnTOCm, TAUBnTDE.TAUBnTDEm = 1

Note that index m is representing a double-digit number for this register. See above.

29.3.7 TAUBn Emulation Register

29.3.7.1 TAUBnEMU — TAUBn Emulation Register

This register controls SVSTOP operations.

Access: This register can be read/written in 8-bit units only when TAUBnTE.TEm = 0. This register can be written only while the counter is stopped (TAUBnTE.TAUBnTEm = 0 and EPC.SVSTOP=0).

Address: <TAUBn_base> + 290_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TAUBnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 29.35 TAUBnEMU Register Contents

Bit Position	Bit Name	Function
7	TAUBnSVSDIS	(When EPC.SVSTOP = 0) Regardless of the value of this bit (1/0), the count clock is continuously supplied when the debugger obtains the control of the microcontroller (e.g., at a breakpoint). (When EPC.SVSTOP = 1) 0: The count clock stops when the debugger obtains the control of the microcontroller (e.g., at a breakpoint). 1: The count clock is continuously supplied when the debugger obtains the control of the microcontroller (e.g., at a breakpoint).
6 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

29.4 General Operating Procedure

The following lists the general operation procedure for the TAUBn:

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUBTTOUTm is also initialized and outputs a low level.

1. Set the TAUBnTPS register to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUBn function:
   - Set the operation mode
   - Set the channel output mode
   - Set any other control bits
3. Enable the counter by setting the TAUBnTS.TAUBnTSm bit to 1.
   The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.
4. If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUBnTT.TAUBnTTm bit to 1. The counter can be forcibly restarted by setting the TAUBnTS.TAUBnTSm bit to 1.
5. Stop the function by setting the TAUBnTT.TAUBnTTm bit to 1.

NOTE

- A detailed description of the required control bits and the operation of the individual functions is given below.
  - Section 29.12, Independent Channel Operation Functions
  - Section 29.14, Synchronous Channel Operation Functions
- The function can be changed while the counter is stopped (TAUBnTE.TAUBnTEm=0).

29.5 Concepts of Synchronous Channel Operation

The synchronous channel operation function is implemented using a combination of channel groups (consisted of master and slave channels).

Several rules apply to the settings of channels.

These rules are detailed in Section 29.5.1, Rules of Synchronous Channel Operation Function.

Two special features for synchronous channel operation are detailed in the following:

- Section 29.5.2, Simultaneous Start and Stop of Synchronous Channel Counters
- Section 29.6, Simultaneous Rewrite

29.5.1 Rules of Synchronous Channel Operation Function

Number of masters and slaves

- Only even channels (CH0, CH2, CH4, …) can be set as master channels.
  Any channel apart from CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and several slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 and the lower channels (CH3, CH4, CH5, …) can be set as slave channels.
- If multiple master channels are used, slave channels cannot cross the master channels.
  Example: If CH0 and CH4 are master channels, CH1 to CH3 can be set as slave channels for CH0, but CH5 to CH15 cannot.

Operation clock

- The same operation clock must be set for the master channel and the synchronized slave channel. This is achieved by setting the same value to the TAUBnCMORm.TAUBnCKS[1:0] bits of the slave and master channel.

The basic concepts of master/slave channel usage and operation clocks are illustrated in the following figure.

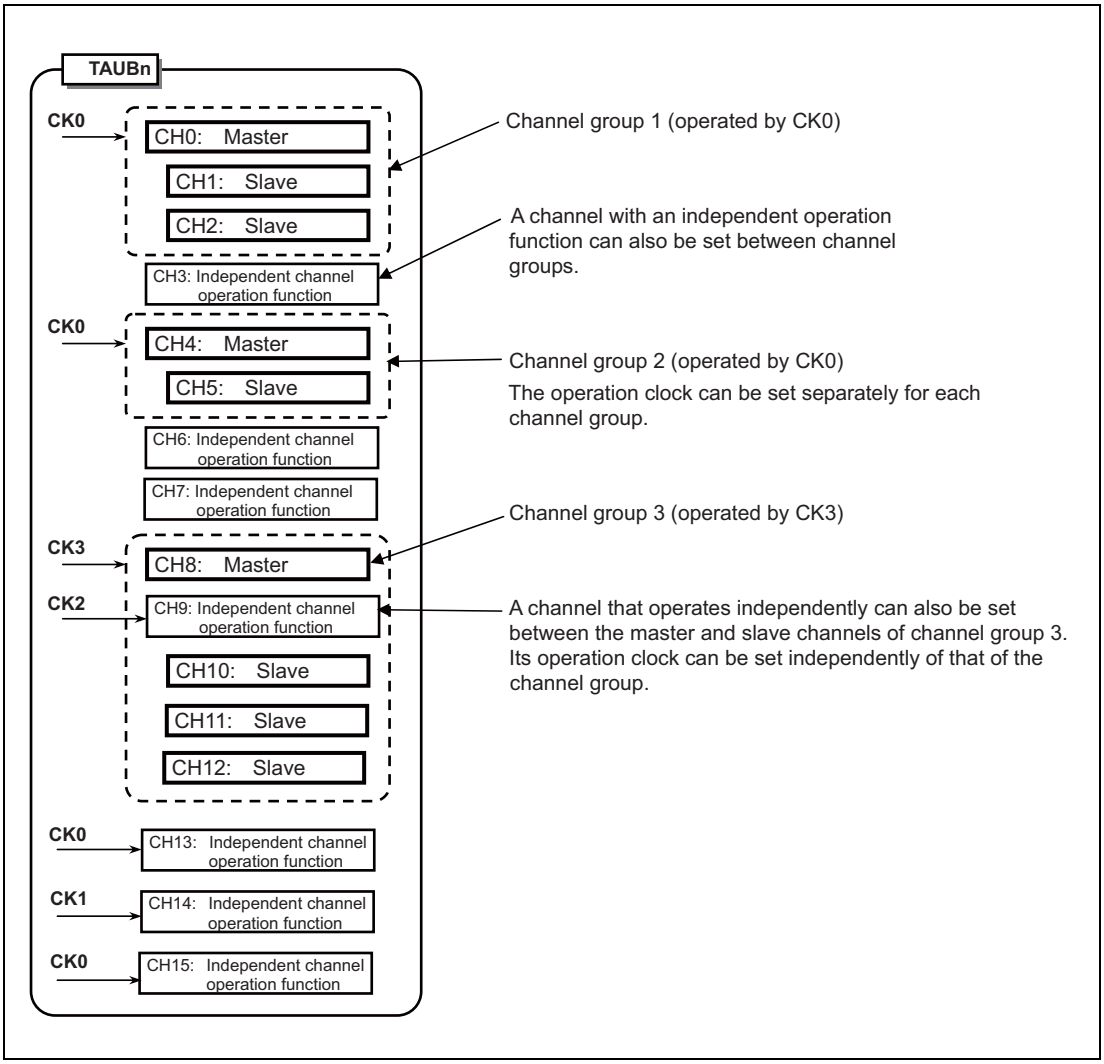


Figure 29.3 Grouping of the Channels and Assignment of Operation Clocks

29.5.2 Simultaneous Start and Stop of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously within the same unit and between the units.

29.5.2.1 Simultaneous Start and Stop within the Same Unit

- To simultaneously start synchronized channels, the TAUBnTS.TAUBnTSm bits of the channels must be set at the same time.
- To simultaneously stop synchronized channels, the TAUBnTT.TAUBnTTm bits of the channels must be set at the same time.

Setting the TAUBnTS.TAUBnTSm bits to 1 sets the corresponding TAUBnTE.TAUBnTEm bits to 1, enabling counting. The exact time that it starts depends on the operation mode.

29.5.2.2 Simultaneous Start between TAUB Units

Counters in different TAUB units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

29.6 Simultaneous Rewrite

29.6.1 Introduction

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data and control registers (TAUBnCDRm and TAUBnTOLm) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite can be triggered by:

- The counter on the master channel or upper channel (depending on the selected operation mode) reaching a certain value
- INTTAUBnIm being issued on the upper channel specified by TAUBnRDC.TAUBnRDCm

There are three methods for simultaneous rewrite. These are listed in the following table, along with how to specify them and when they cause simultaneous rewrite to be triggered.

Table 29.36 Simultaneous Rewrite Methods and when They are Triggered

Method	Simultaneous Rewrite Triggered when	TAUBnRDE.TAUBnRDEm	TAUBnRDS.TAUBnRDSm	TAUBnRDM.TAUBnRDMm
—	No simultaneous rewrite	0	0	0
A	The master channel (re)starts counting	1	0	0
B	Counting is started in the master channel. The master channel starts counting down at the peak of triangular wave of the corresponding slave channel.	1	0	1
C1	INTTAUBnIm is generated on an upper channel specified by TAUBnRDC.TAUBnRDCm	1	1	0/1

The following table lists which of these three methods is available for each channel operation function. For more information about the individual channel operation functions, see the corresponding sections in **Section 29.13, Independent Channel Simultaneous Rewrite Functions** and Section 29.14, Synchronous Channel Operation Functions.

Table 29.37 Simultaneous Rewrite Methods and when They are Triggered

Functions	A	B	C	TAUBnTOL.TAUBnTOLm
Simultaneous Rewrite Trigger Output Function Type 1			X	
PWM Output Function	X		X	X
One-Shot Pulse Output Function	X			
Delay Pulse Output Function	X			
Triangle PWM Output Function		X	X	X
Triangle PWM Output Function with Dead Time		X	X	
AD Conversion Trigger Output Function Type 1	X		X	
AD Conversion Trigger Output Function Type 2		X	X	

29.6.2 How to Control Simultaneous Rewrite

The following figure shows the general procedure for simultaneous rewrite.
The three main blocks (Initial settings, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

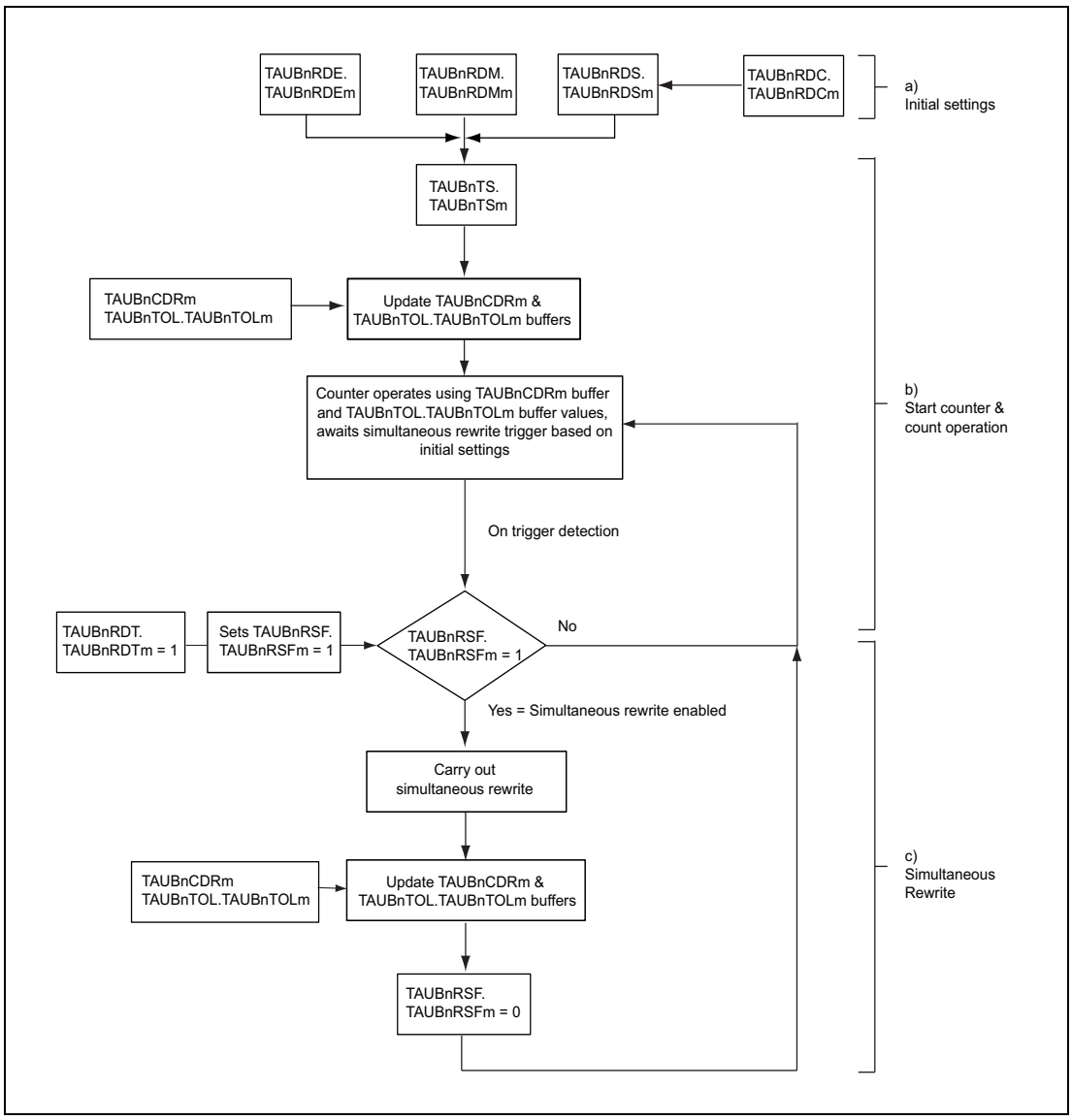


Figure 29.4 General Procedure for Simultaneous Rewrite

29.6.2.1 Initial Settings

- To enable simultaneous rewrite in channel m, set TAUBnRDE.TAUBnRDEm = 1
- To select the type of simultaneous rewrite, set TAUBnRDM.TAUBnRDMm and TAUBnRDS.TAUBnRDSm according to the values in **Table 29.36, Simultaneous Rewrite Methods and when They are Triggered**.
- To select which upper channel is monitored for the simultaneous rewrite trigger use TAUBnRDC.TAUBnRDCm (prerequisite: TAUBnRDS.TAUBnRDSm is set to upper channel)

29.6.2.2 Start Counter and Count Operation

- To start all the TAUBnCNTm counters in the channel group, set the corresponding TAUBnTS.TAUBnTSm bits to 1. TAUBnTOL.TAUBnTOLm and the values in the data registers (TAUBnCDRm) are written to the corresponding TAUBnTOL.TAUBnTOLm buffer (TAUBnTOL.TAUBnTOLm buf) and data buffer registers (TAUBnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUBnRDT.TAUBnRDTm) to 1 sets the reload flag (TAUBnRSF.TAUBnRSFm) to 1, enabling simultaneous rewrite. TAUBnRSF.TAUBnRSFm remains at 1 until simultaneous rewrite has taken place.
- When the specified trigger for simultaneous rewrite is detected, the TAUBnRSF.TAUBnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUBnRSF.TAUBnRSFm = 1). If it is, simultaneous rewrite is carried out.
  Otherwise, simultaneous rewire is not carried out, and the system awaits the next simultaneous rewrite trigger detection.

29.6.2.3 Simultaneous Rewrite

- When simultaneous rewrite is enabled (TAUBnRSF.TAUBnRSFm = 1) and the simultaneous rewrite trigger is detected, the current values of the data registers are copied to their buffers. These values are then written to the corresponding counters and the values are applied the next time the counter starts or restarts.
- When simultaneous rewrite is finished, the TAUBnRSF.TAUBnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

29.6.3 Other General Rules of Simultaneous Rewrite

The following rules also apply:

- TAUBnRDE.TAUBnRDEm, TAUBnRDS.TAUBnRDSm, TAUBnRDM.TAUBnRDMm, and TAUBnRDC.TAUBnRDCm cannot be changed while the counter is in operation (TAUBnTE.TAUBnTEm = 1).
- TAUBnTOL.TAUBnTOLm can only be rewritten during operation when in PWM output function or triangle PWM output function. For all other output functions, TAUBnTOL.TAUBnTOLm must be written before the counter starts. If it is rewritten in another function, TAUBTTOUTm outputs an invalid wave.
- When an upper channel is used as the channel issuing the simultaneous rewrite trigger generation channel(TAUBnRDS.TAUBnRDSm = 1), the TAUBnRDC.TAUBnRDCm bit controls all the lower channels. This means that if the TAUBnRDC.TAUBnRDCm bits of CH2 and CH7 are set to 1 and the TAUBnRDC.TAUBnRDCm bits of other channels are set to 0, CH2 and CH7 serve as simultaneous rewrite trigger generation channels. CH2 controls the lower channels CH3 to CH6, and CH7 controls the lower channels CH8 to CH15.
- If simultaneous rewrite is enabled and an upper channel is selected for the simultaneous rewrite trigger generation channel (TAUBnRDE.TAUBnRDEm and TAUBnRDS.TAUBnRDSm = 1) but no upper channel is set (TAUBnRDC.TAUBnRDC[15:0] = 0), simultaneous rewrite cannot take place.

29.6.4 Types of Simultaneous Rewrite

In the following section, the three simultaneous rewrite methods are explained using timing diagrams.

29.6.4.1 Simultaneous Rewrite when the Master Channel (Re)Starts Counting (Method A)

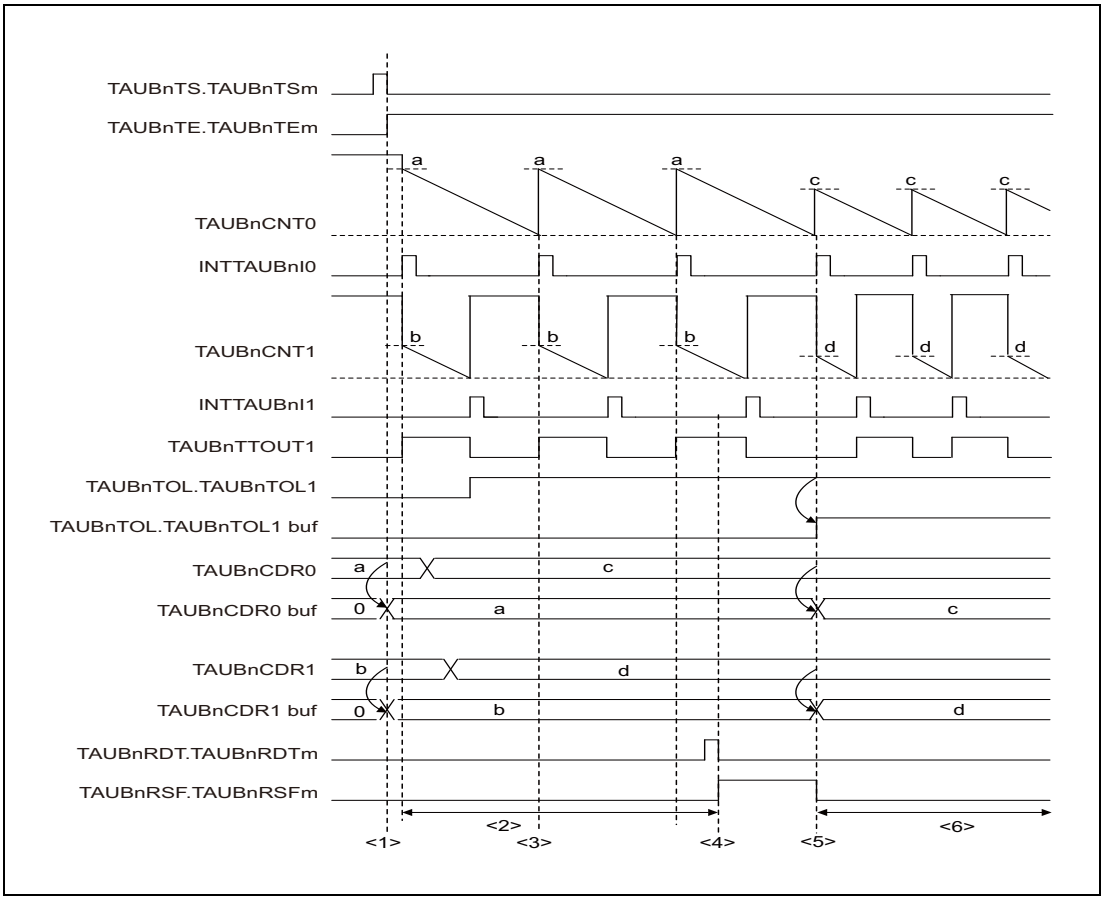


Figure 29.5 Simultaneous Rewrite when the Master Channel (Re)Starts Counting

Setting:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method A is applied.

Description:

(1) When TAUBnTS.TAUBnTSm = 1 is set, the value of TAUBnCDRm is copied to the TAUBnCDRm buffer and the value of TAUBnTOL.TAUBnTOLm is copied to the TAUBnTOL.TAUBnTOLm buffer.

(2) The TAUBnCDRm and TAUBnTOL.TAUBnTOLm registers can be written at any time.

(3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUBnRSF.TAUBnRSFm = 0)

(4) The reload data trigger bit (TAUBnRDT.TAUBnRDTm) is set to 1 which sets the status flag (TAUBnRSF.TAUBnRSFm = 1), enabling simultaneous rewrite.

(5) Because simultaneous rewrite is enabled, it is triggered when CH0 restarts counting. The TAUBnCDRm value is loaded into the TAUBnCDRm buffer, and the TAUBnTOL.TAUBnTOLm value is loaded into the TAUBnTOL.TAUBnTOLm buffer.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUBnCDRm and TAUBnTOL.TAUBnTOLm can be changed again.

29.6.4.2 Simultaneous Rewrite at the Peak of a Triangular Wave of the Slave Channel (Method B)

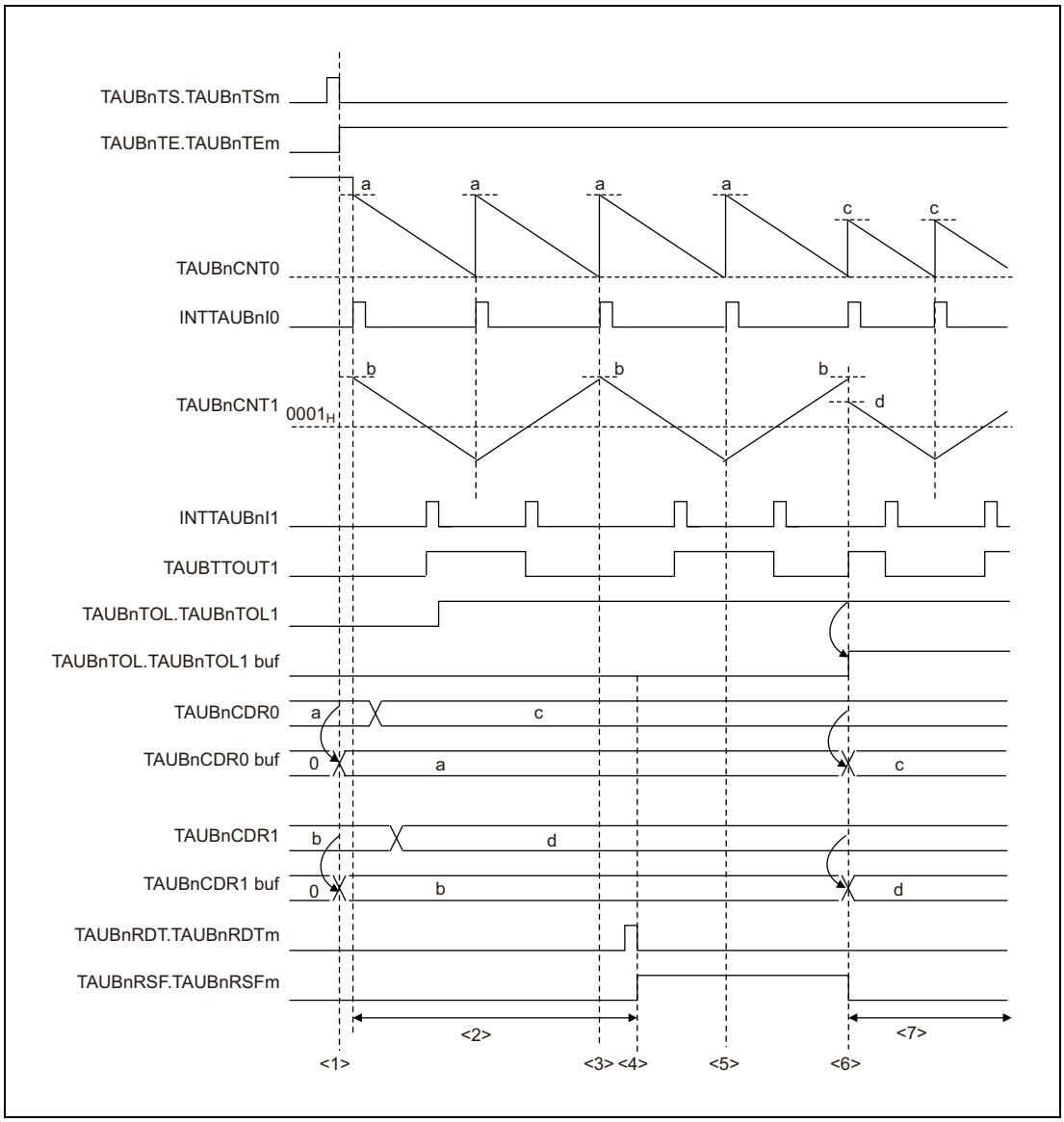


Figure 29.6 Simultaneous Rewrite at the Peak of a Triangular Wave of the Slave Channel

Setting:

CH0 is the master channel, which starts counting down, and CH1 represents an arbitrary slave channel. The simultaneous rewrite method B is applied.

Description:

(1) When TAUBnTS.TAUBnTSm = 1 is set, the value of TAUBnCDRm is copied to the TAUBnCDRm buffer.

(2) The TAUBnCDRm and TAUBnTOL registers can be written at any time.

(3) Simultaneous rewrite does not occur because it is disabled (TAUBnRSF.TAUBnRSFm = 0).

(4) The reload data trigger bit (TAUBnRDT.TAUBnRDTm) is set to 1 which sets the status flag (TAUBnRSF.TAUBnRSFm = 1), enabling simultaneous rewrite.

(5) Simultaneous rewrite does not take place at the bottom of the triangular cycle.

(6) Simultaneous rewrite takes place at the start timing of the top of the triangular cycle. The TAUBnCDRm value is loaded into the TAUBnCDRm buffer, and the TAUBnTOL.TAUBnTOLm value is loaded into the TAUBnTOL.TAUBnTOLm buffer.

(7) The counters count down and await the next simultaneous rewrite trigger. The values of TAUBnCDRm and TAUBnTOL.TAUBnTOLm can be changed again.

29.6.4.3 Simultaneous Rewrite when INTTAUBnIm is Generated on an Upper Channel Specified by TAUBnRDC.TAUBnRDCm (Method C1)

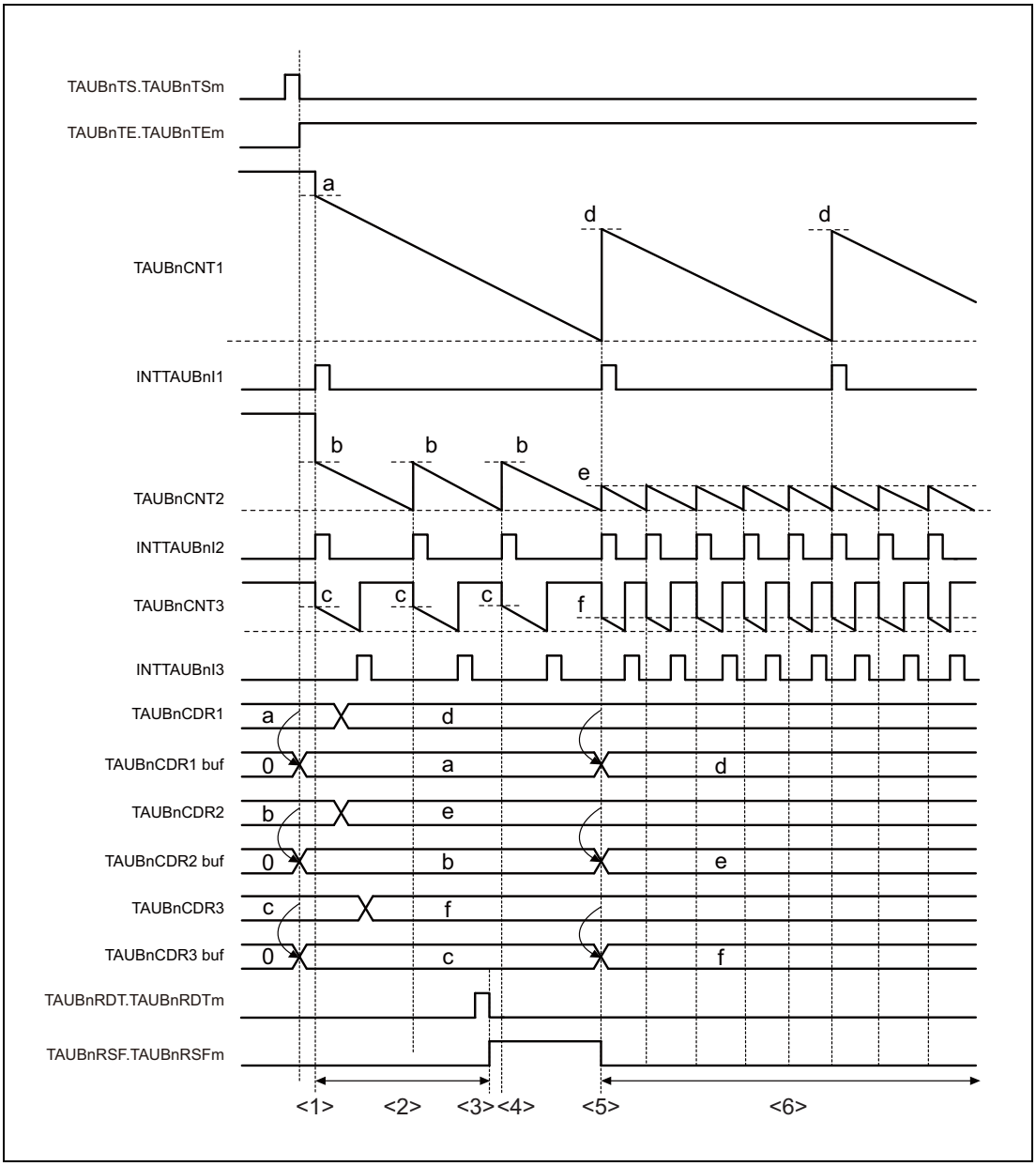


Figure 29.7 Simultaneous Rewrite when INTTAUBnIm is Generated on an Upper Channel Specified by TAUBnRDC.TAUBnRDCm

Setting:

CH1 is an upper channel used counting down, CH2 is a master channel, and CH3 is the slave channel. The simultaneous rewrite method C1 is applied. The TAUBnRDC register specifies a channel which generates simultaneous rewrite triggers.

Description:

(1) When TAUBnTS.TAUBnTSm is set to 1, the TAUBnCDRm value is copied to the TAUBnCDRm buffer.

(2) The TAUBnCDRm register is always ready to write.

(3) By setting the reload data trigger bit (TAUBnRDT.TAUBnRDTm) to 1, the status flag is set (TAUBnRSF.TAUBnRSFm = 1) to enable simultaneous rewrite.

(4) Simultaneous rewrite is triggered only by a CH1 interrupt. Therefore, simultaneous rewrite is not conducted even if enabled.

(5) Simultaneous rewrite is triggered by INT1 which is generated when counter 1 reaches 0000_H. The TAUBnCDRm values are loaded into the corresponding TAUBnCDRm buffers.

(6) The counter counts down and awaits the next simultaneous rewrite trigger. The values of the TAUBnCDRm registers can be rechanged.

29.7 Channel Output Modes

The output of the TAUBTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUBnTOE.TAUBnTOEm = 0)
  When controlled by software, the value written in the output register bit (TAUBnTO.TAUBnTOm) is sent to the output pin (TAUBTTOUTm).
- By TAUB signals (TAUBnTOE.TAUBnTOEm = 1)
  When controlled by TAUB signals, the output level of TAUBTTOUTm is set or reset or toggled by internal signals. The value of TAUBnTO.TAUBnTOm is updated accordingly to reflect the value of TAUBTTOUTm.
  - Independently (TAUBnTOM.TAUBnTOMm = 0)
    In case of independent operation, the output of the TAUBTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUBnTOM.TAUBnTOMm = 0).
  - Synchronously (TAUBnTOM.TAUBnTOMm = 1)
    In case of synchronous operation, the output of the TAUBTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUBnTOM.TAUBnTOMm = 1).

The TAUBnTO.TAUBnTOm bit can always be read to determine the current value of TAUBTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in Table 29.38, Channel Output Modes.

The channel output modes are described in details below.

- Section 29.7.2, Channel Output Modes Controlled Independently by TAUBn Signals
- Section 29.7.3, Channel Output Modes Controlled Synchronously by TAUBn Signals

Batch operation of TAUBnTOm bit

Whether a set value is reflected to the TAUBnTOm bit or not is controlled by the TAUBnTOE.TAUBnTOEm bit.

The TAUBnTOm setting is written only to the bit (channel) set with TAUBnTOE.TAUBnTOEm bit = 0 when a write to the TAUBnTO register is attempted. No TAUBnTOm setting is reflected to the bit (channel) set with TAUBnTOE.TAUBnTOEm bit = 1.

NOTE

TAUBnTO.TAUBnTOm bit is placed so that its bit number corresponds to a channel number.

Output logic

Positive logic or negative logic of the output is specified by control bit TAUBnTOL.TAUBnTOLm.

The value of TAUBnTOL.TAUBnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function or triangle PWM output function. Otherwise, changes to TAUBnTOL.TAUBnTOLm result in an invalid TAUBTTOUTm signal output.

See Section 29.6, Simultaneous Rewrite.

The various channel output modes and the channel output control bits are listed in **Table 29.38**.

Table 29.38 Channel Output Modes

Channel Output Mode	TAUBnTOE. TAUBnTOEm	TAUBnTOM. TAUBnTOMm	TAUBnTOC. TAUBnTOCm	TAUBnTDE. TAUBnTDEm
By software				
Independent channel output mode controlled by software	0	x		
By TAUB signals, independently				
Independent channel output mode 1	1	0	0	0
Independent channel output mode 2			1	
By TAUB signals, synchronously				
Synchronous channel output mode 1	1	1	0	0
Synchronous channel output mode 2			1	0
Synchronous channel output mode 2 with dead time output				1

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

NOTE

The following bits cannot be changed during count operation (TAUBnTE.TAUBnTEm = 1):

- TAUBnTOM.TAUBnTOMm
- TAUBnTOC.TAUBnTOCm
- TAUBnTDE.TAUBnTDEm

29.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUBTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUBnTOE.TAUBnTOEm = 0).

(1) Set TAUBnTO.TAUBnTOm to specify the initial level of the TAUBTTOUTm output.

(2) Set channel output mode according to Table 29.38, Channel Output Modes, and the output logic using the TAUBnTOL.TAUBnTOLm bit.

(3) Start the counter (TAUBnTS.TAUBnTSm = 1).

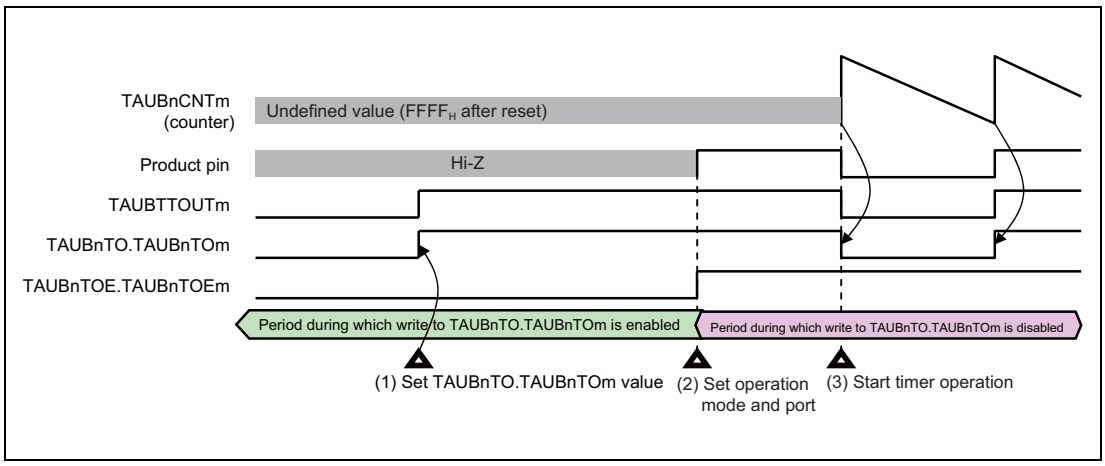


Figure 29.8 General Procedure for Specifying a TAUBTTOUTm Channel Output Mode

29.7.2 Channel Output Modes Controlled Independently by TAUBn Signals

This section lists the channel output modes that are controlled independently by TAUBn signals. The control bits used to specify a mode are listed in Table 29.38, Channel Output Modes.

29.7.2.1 Independent Channel Output Mode 1

Set/reset conditions

In this output mode, TAUBTTOUTm toggles when INTTAUBnIm is detected. The value of TAUBnTOL.TAUBnTOLm is ignored.

Prerequisites

There are no prerequisites other than those shown in Table 29.38, Channel Output Modes.

29.7.2.2 Independent Channel Output Mode 2

Set/reset conditions

In this output mode, TAUBTTOUTm is set when INTTAUBnIm occurs at the time of count start, and reset when INTTAUBnIm occurs due to a match between TAUBnCNTm and TAUBnCDRm.

Prerequisites

There are no prerequisites other than those shown in Table 29.38, Channel Output Modes.

29.7.3 Channel Output Modes Controlled Synchronously by TAUBn Signals

This section lists the channel output modes that are controlled synchronously by TAUBn signals. The control bits used to specify a mode are listed in Table 29.38, Channel Output Modes.

29.7.3.1 Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUBnIm of master channel serves as a set signal and INTTAUBnIm of the slave channel as a reset signal. If INTTAUBnIm of master channel and INTTAUBnIm of the slave channel are generated at the same time, INTTAUBnIm of the slave channel (reset signal) has priority over INTTAUBnIm (set signal) of master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in Table 29.38, Channel Output Modes.

29.7.3.2 Synchronous Channel Output Mode 2

In this output mode, the operating mode should be set to count-up/-down mode. The result is a triangle PWM wave at TAUBTTOUTm. For details, see Section 29.14.5, Triangle PWM Output Function.

Set/reset conditions

TAUBnCNTm of the slave channel counts down and up alternatively. When it passes 0001_H it generates an interrupt, causing TAUBTTOUTm to toggle.

Prerequisites

A set of two channels is required to generate the triangle PWM output. TAUBTTOUTm should be set to 0 before the function starts.

29.7.3.3 Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to TAUBTTOUTm. The set/reset conditions are shown in **Figure 29.9**.

Set/reset conditions

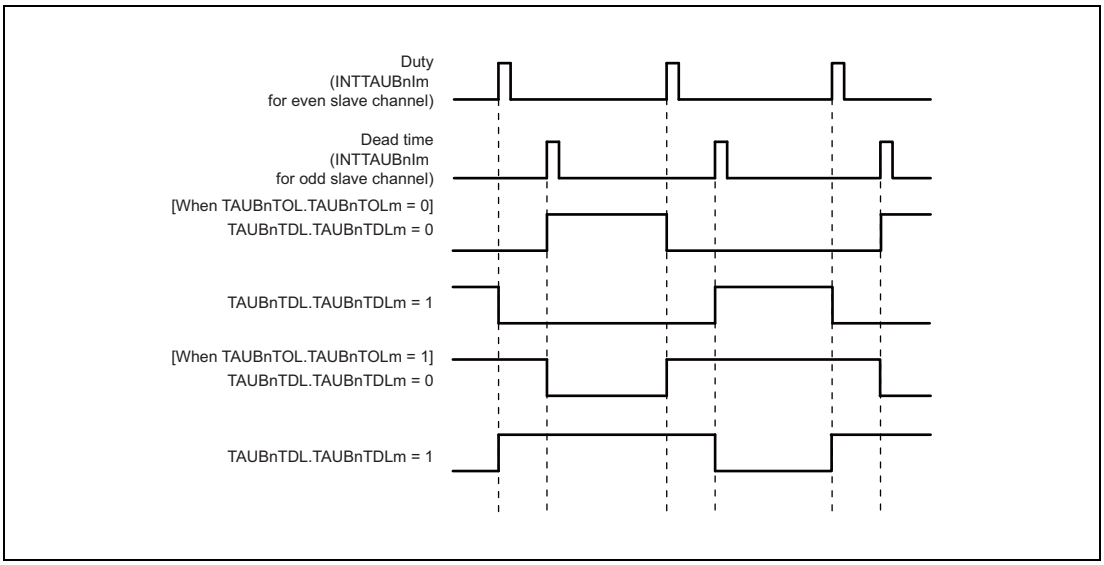


Figure 29.9 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output

With regard to the edge to which dead time is added, set TAUBnTDL.TAUBnTDLm = 0 for rising edges and TAUBnTDL.TAUBnTDLm = 1 for falling edges.

Prerequisites

Dead time control requires a set of three channels, each operating in the following modes:

- One master channel
  The master channel should be set to interval timer mode.
- One even slave channel
  The even slave channel should be set to count-up/-down mode.
- One odd slave channel (even channel + 1)
  The odd slave channel should be set to one-count mode.

The values of the following bits should be the same for the odd channel and the even channel:

- TAUBnTOE.TAUBnTOEm
- TAUBnTOM.TAUBnTOMm
- TAUBnTOC.TAUBnTOCm
- TAUBnTDE.TAUBnTDEm

29.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUBnTS.TAUBnTSm is set to 1 in each operating mode.

In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

29.8.1 Interval Timer Mode, Judge Mode, Capture Mode, Count-Up/-Down Mode, and Count Capture Mode

The counter starts operating with the next count clock after TAUBnTS.TAUBnTSm is set to 1. The value of data register is also loaded when the counter starts.

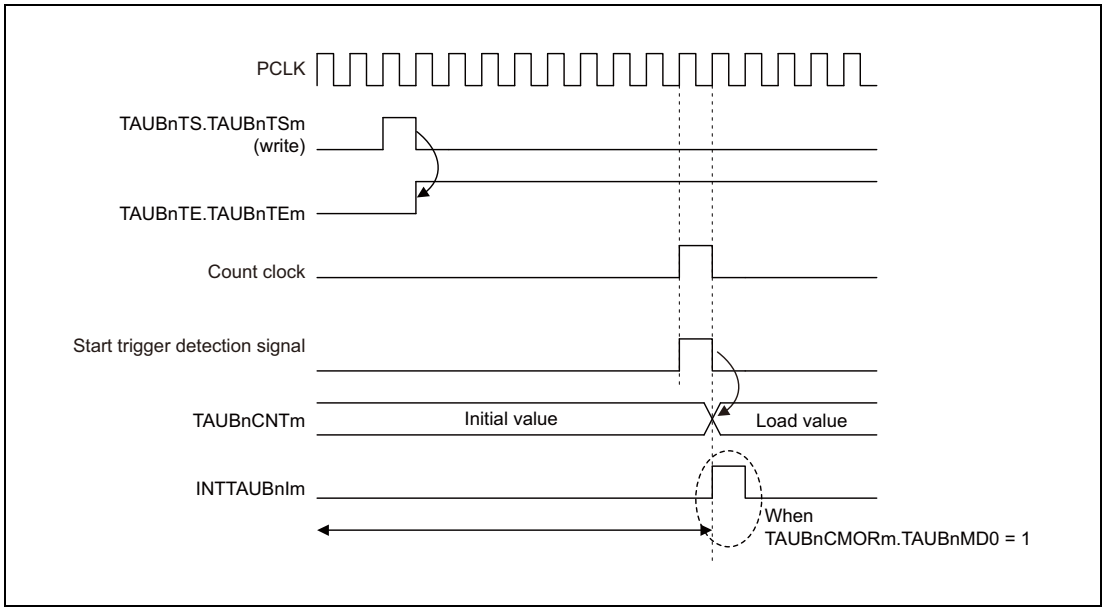


Figure 29.10 Start Timing in Interval Timer Mode, Judge Mode, Capture Mode, Up/Down Count Mode, and Count Capture Mode

NOTE

Make sure to set TAUBnCMORm.TAUBnMD0 to 0 when using the count-up/-down mode.

29.8.2 Event Count Mode

The value of data register is loaded as soon as TAUBnTS.TAUBnTSm is set to 1. The counter also starts immediately. The value of data register decrements when the subsequent count clock cycle starts.

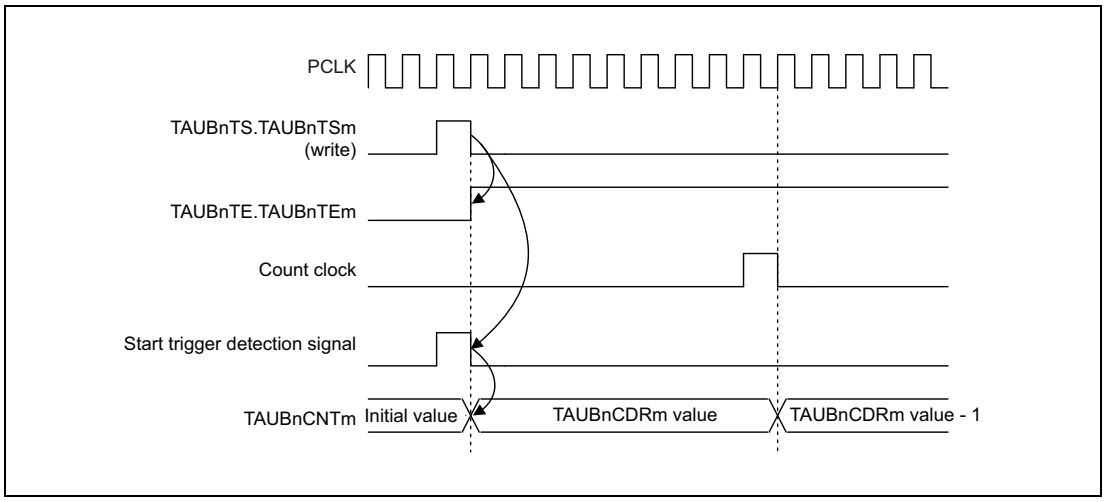


Figure 29.11 Start Timing in Event Count Mode

29.8.3 Other Operating Modes

In other operating modes, the counter operation start timing is triggered only upon detection of a valid edge of TAUBTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which is irrelevant to start of counter operation, determine the frequency with which all operations take place.

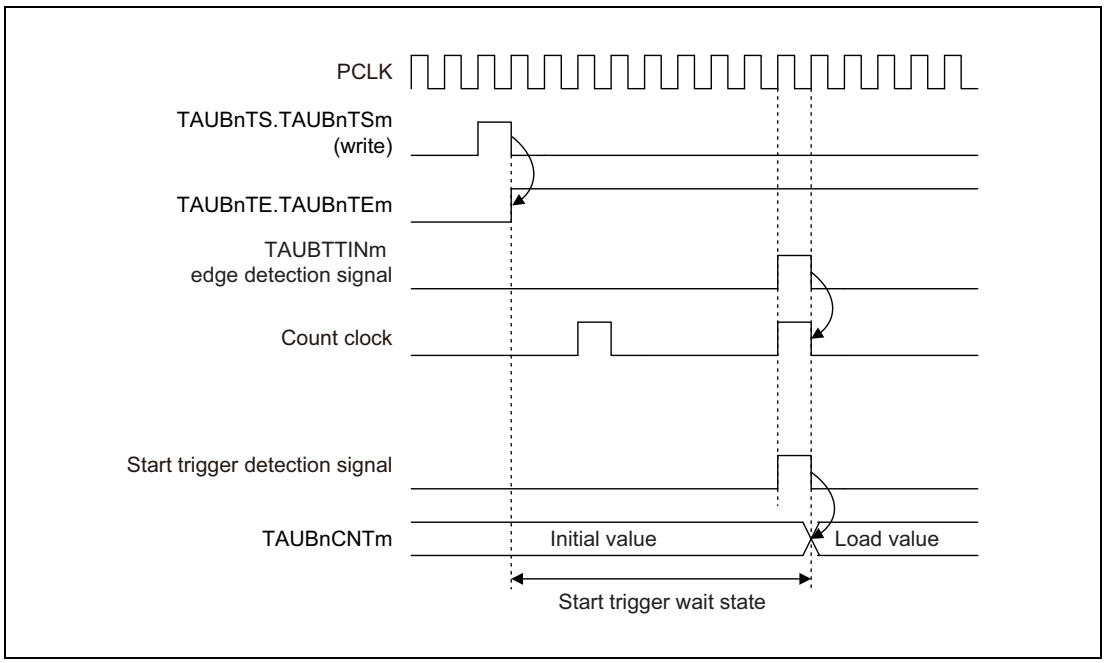


Figure 29.12 Start Timing in Other Operating Modes

29.9 TAUBTTOUTm Output and INTTAUBnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUBnIm is generated using the TAUBnCMORm.TAUBnMD0 bit. The generation of INTTAUBnIm when the TAUBnCMORm.TAUBnMD0 bit starts counting and the effect to TAUBTTOUTm depend on the selected function. For details, refer to the description of TAUBnCMORm.TAUBnMD0 of each function.

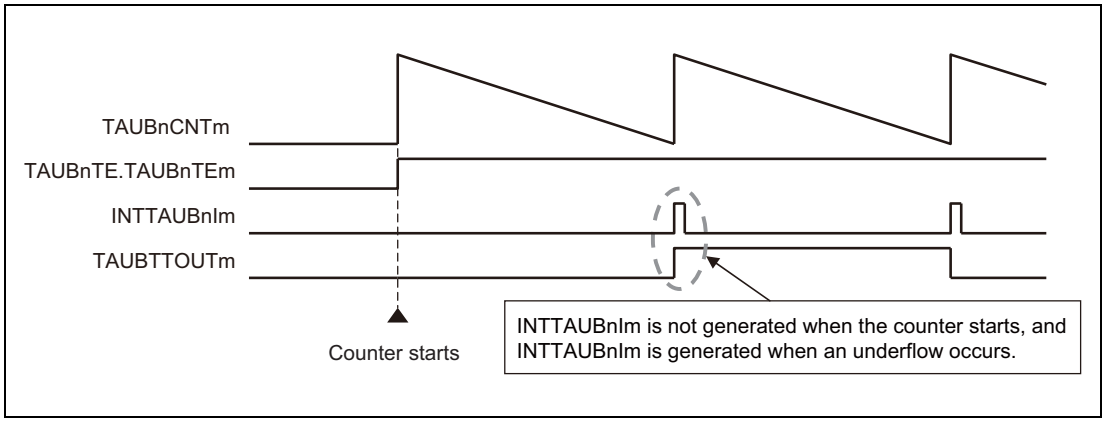


Figure 29.13 INTTAUBnIm Generation Timing (TAUBnCMORm.TAUBnMD0=0)

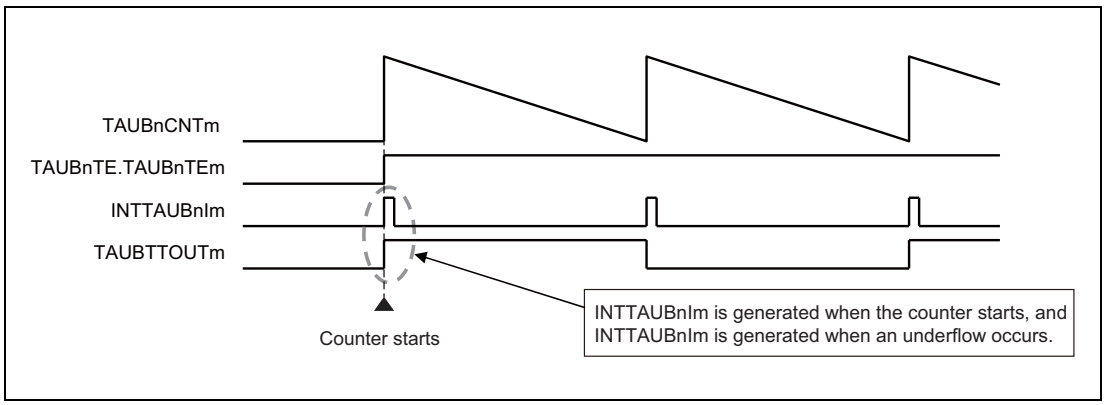


Figure 29.14 INTTAUBnIm Generation Timing (TAUBnCMORm.TAUBnMD0=1)

29.10 Interrupt Generation upon Overflow

Certain independent functions that count up, overflow without generating an interrupt when they reach $FFFF_H$. This section describes how it is possible to generate an interrupt, by combining a channel operating in one of these modes with a channel in a different operation mode which counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches 0000_H at the same time as the first channel overflows (TAUBnCNTm = $FFFF_H$).
- Set TAUBnCDRm of the second channel to $FFFF_H$.
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same TAUBTTINm input.
- The trigger detection settings (TAUBnCMORm.TAUBnSTS[2:0] and TAUBnCMURm.TAUBnTIS[1:0]) must be identical for both channels.

Result:

The down-counter of the second channel reaches 0000_H at exactly the same time as the up-counter of the first channel overflows (TAUBnCNTm = $FFFF_H$). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

29.10.1 Example of Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function

When the capture trigger is input simultaneously to TAUBTTINm of both channels, INTTAUBnIm of the TAUBTTINm input interval timer function can detect the overflow when TAUBnCNTm of the TAUBTTINm input pulse interval measurement function exceeds $FFFF_H$.

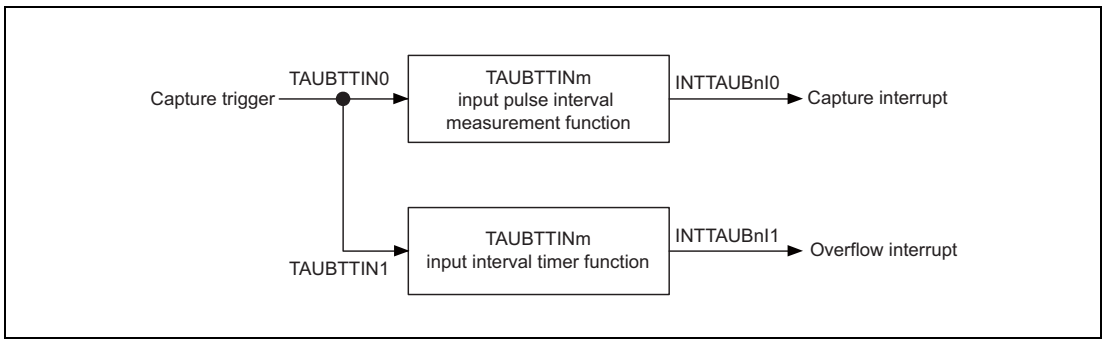


Figure 29.15 Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function

Timing diagram

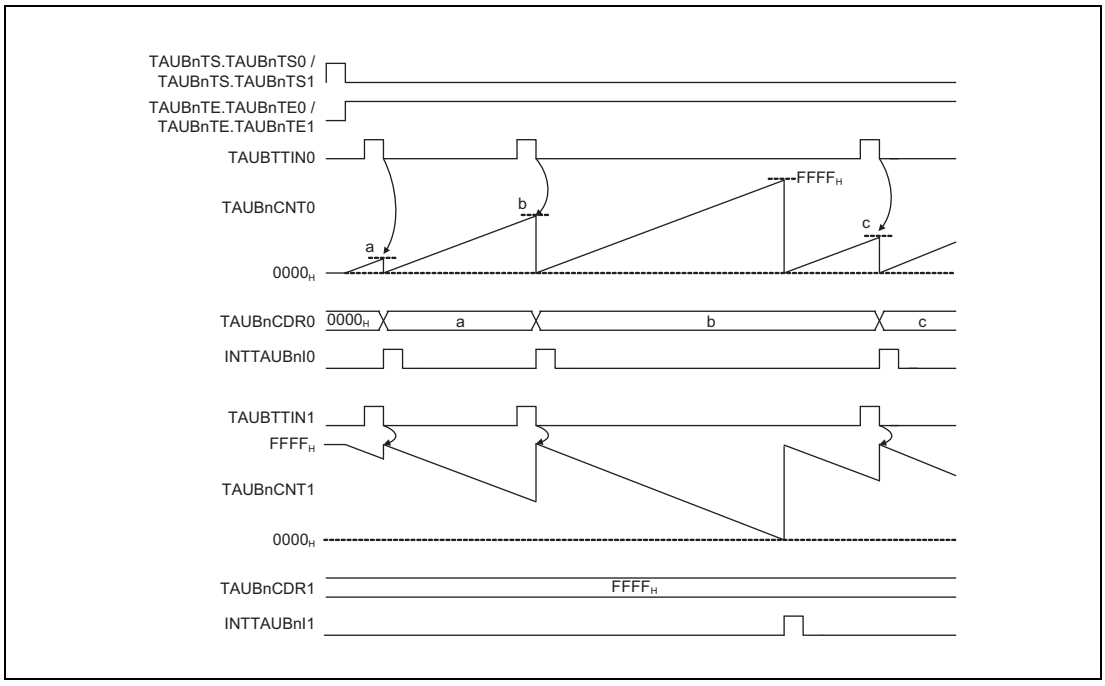


Figure 29.16 Interrupt Generation by Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function

29.10.2 Example of Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

When the capture trigger is input simultaneously to TAUBTTINm of both channels, INTTAUBnIm of the overflow interrupt output function (during TAUBTTINm width measurement) can detect the overflow when TAUBnCNTm of the TAUBTTINm input signal width measurement function exceeds $FFFF_H$.

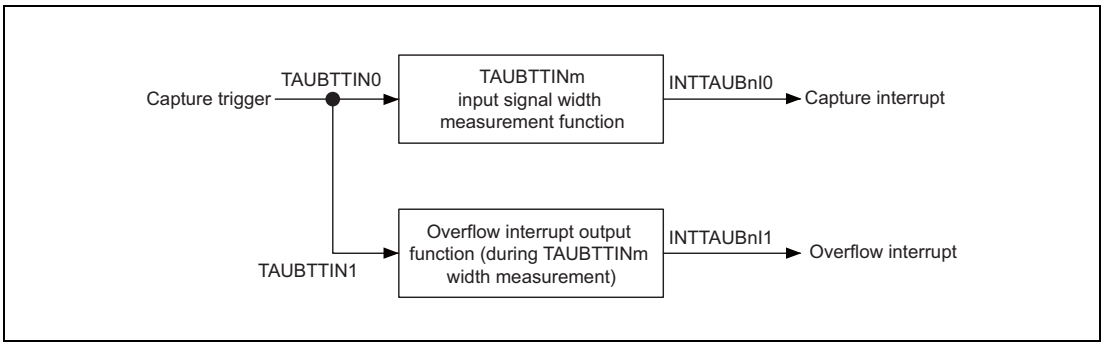


Figure 29.17 Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Timing diagram

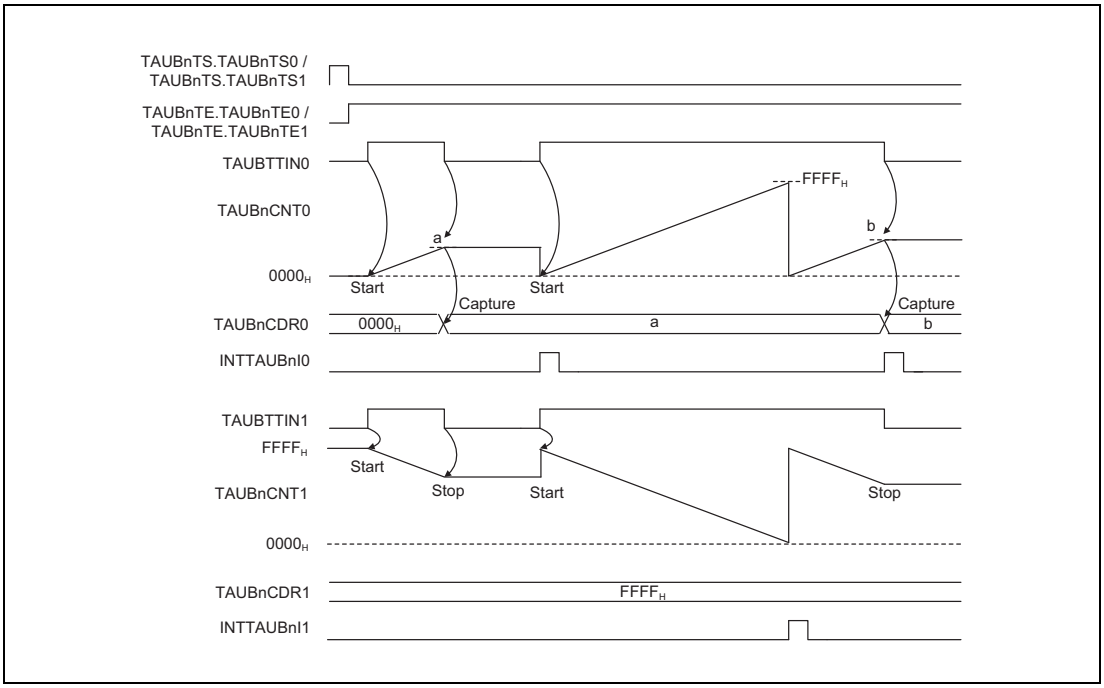


Figure 29.18 Interrupt Generation by Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

29.10.3 Example of Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

When the counters of both channels are started simultaneously, INTTAUBnIm of the interval timer function can detect the overflow when TAUBnCNTm of the TAUBTTINm input position detection function exceeds $FFFF_H$.

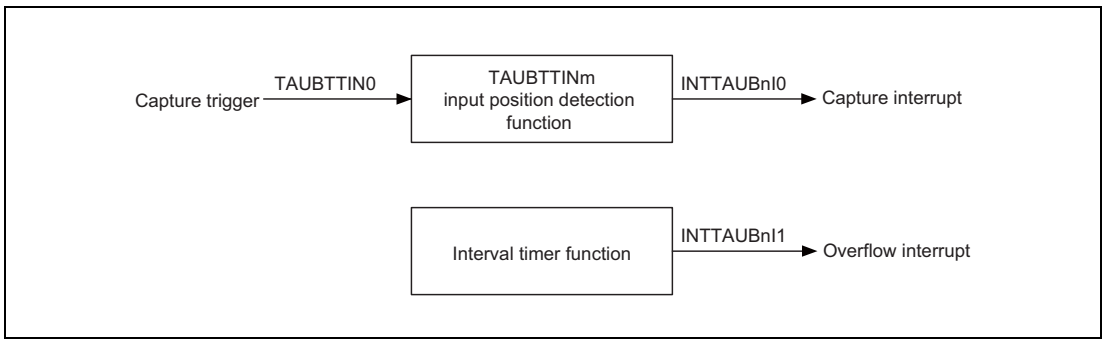


Figure 29.19 Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

Timing diagram

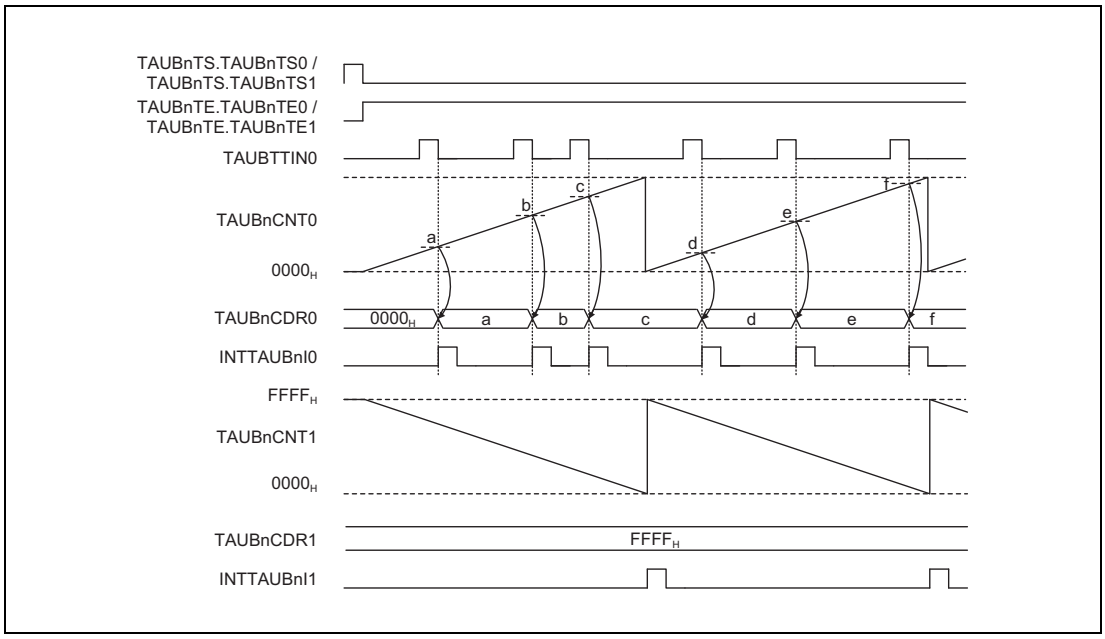


Figure 29.20 Interrupt Generation by Combination of TAUBTTINm Input Position Detection Function and Interval Timer Function

29.10.4 Example of Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

When the capture trigger is input simultaneously to TAUBTTINm of both channels, INTTAUBnIm of the overflow interrupt output function (during TAUBTTINm input period count detection) can detect the overflow when TAUBnCNTm of the TAUBTTINm input period count detection function exceeds $FFFF_H$.

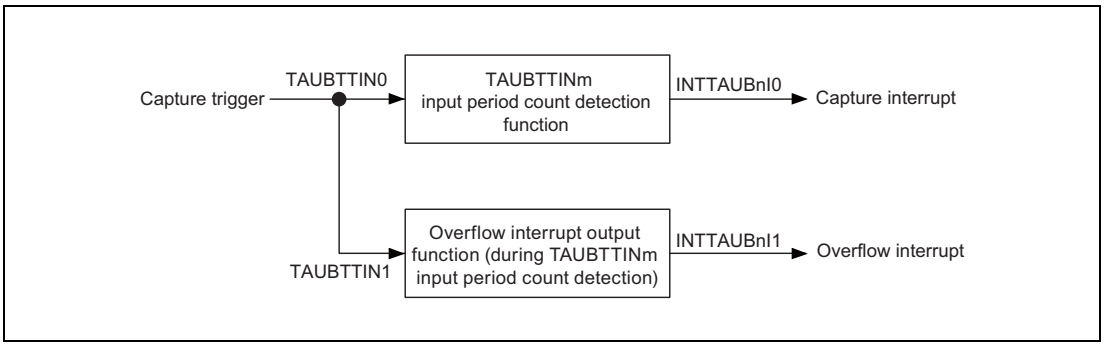


Figure 29.21 Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (TAUBTTINm Input Period Count Detection)

Timing diagram

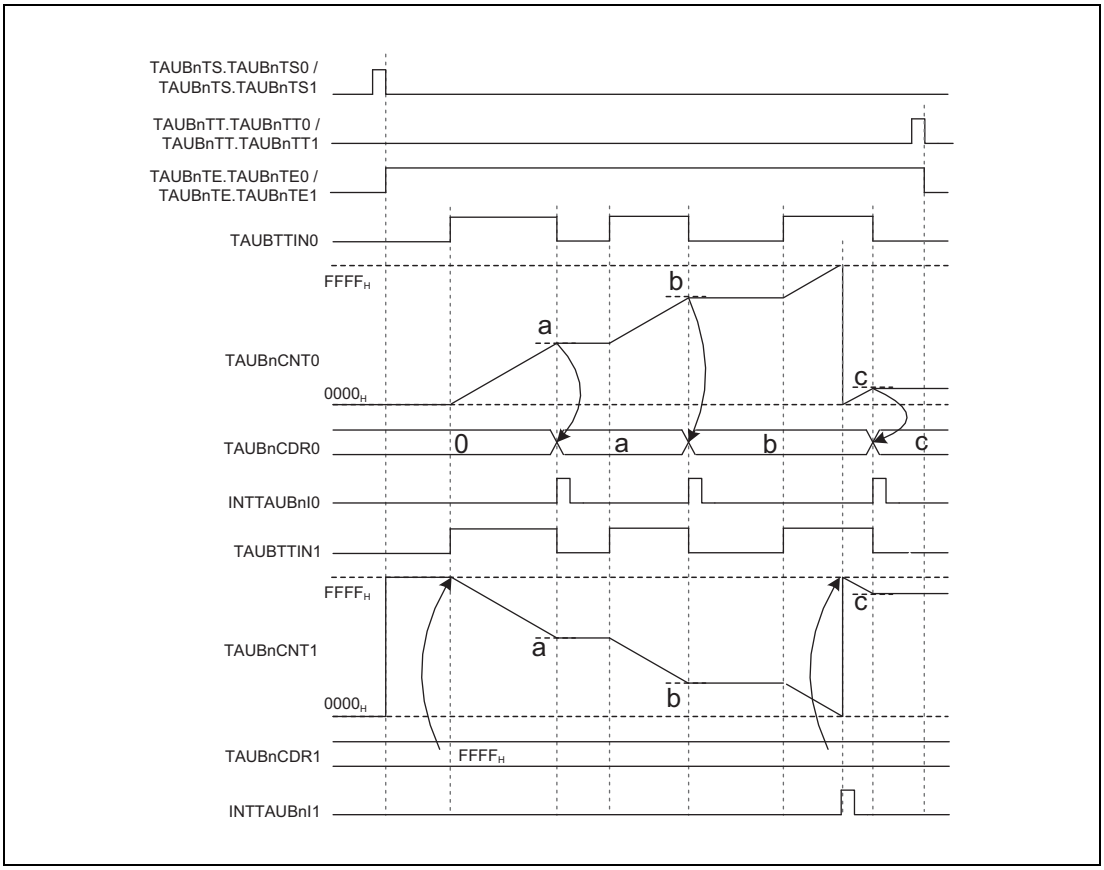


Figure 29.22 Interrupt Generation by Combination of TAUBTTINm Input Period Count Detection Function and Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

29.11 TAUBTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

The following figure shows when edge detection takes place.

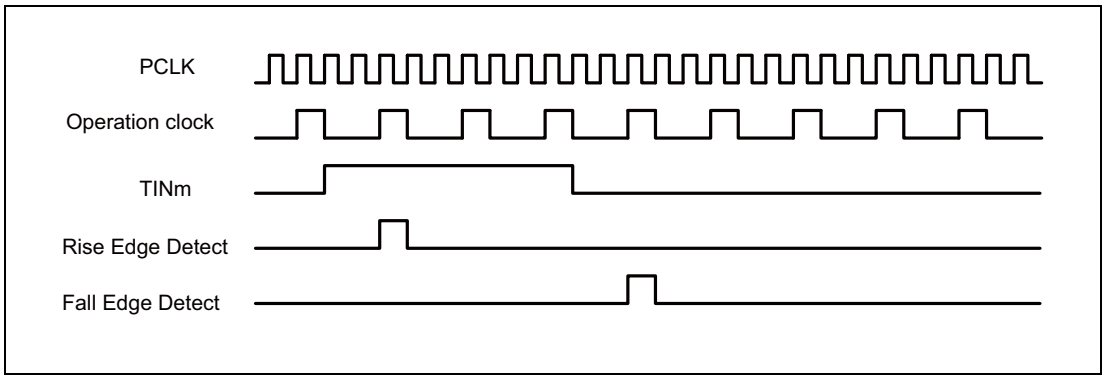


Figure 29.23 Basic Edge Detection Timing

Figure 29.23 is an image of the operation timing. Actually, the delay time caused by the noise filter and the synchronization circuit between the TAUBnIm terminal and TAUBn will be generated.

29.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the TAUB. For a general overview of independent channel operation functions, see Section 29.2, Overview.

29.12.1 Interval Timer Function

29.12.1.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUBnIm) at regular intervals. When an interrupt is generated, the TAUBTTOUTm signal toggles, resulting in a square wave.

Description

The counter is started by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of TAUBnCDRm is loaded to TAUBnCNTm and the counter starts to count down from this value.

When the counter reaches 0000_H, INTTAUBnIm is generated and the TAUBTTOUTm signal toggles. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm stop but retain their values. The counter can be reset by setting TAUBnTS.TAUBnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm to 1 during operation.

Conditions

If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUBTTOUTm does not toggle. This results in an inverted TAUBTTOUTm signal compared to when TAUBnCMORm.TAUBnMD0 is set to 1.

29.12.1.2 Equations

INTTAUBnIm cycle = count clock cycle × (TAUBnCDRm + 1)

TAUBTTOUTm square wave cycle = count clock cycle × (TAUBnCDRm + 1) × 2

29.12.1.3 Block Diagram and General Timing Diagram

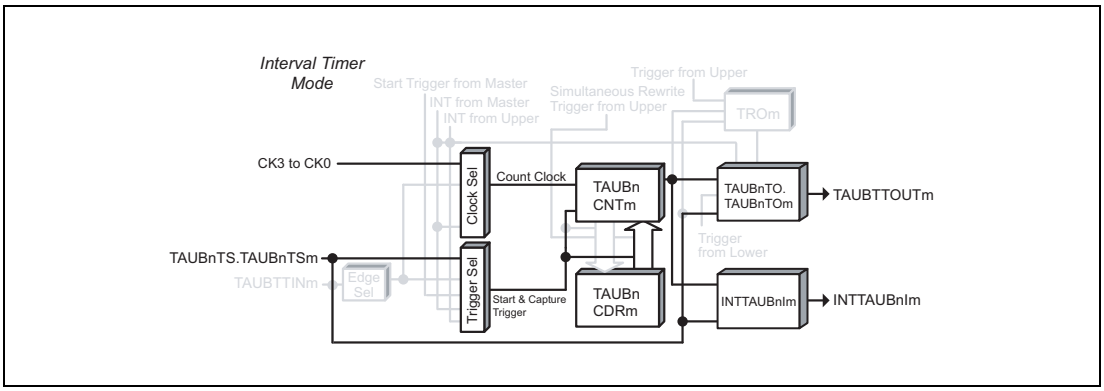


Figure 29.24 Block Diagram for Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

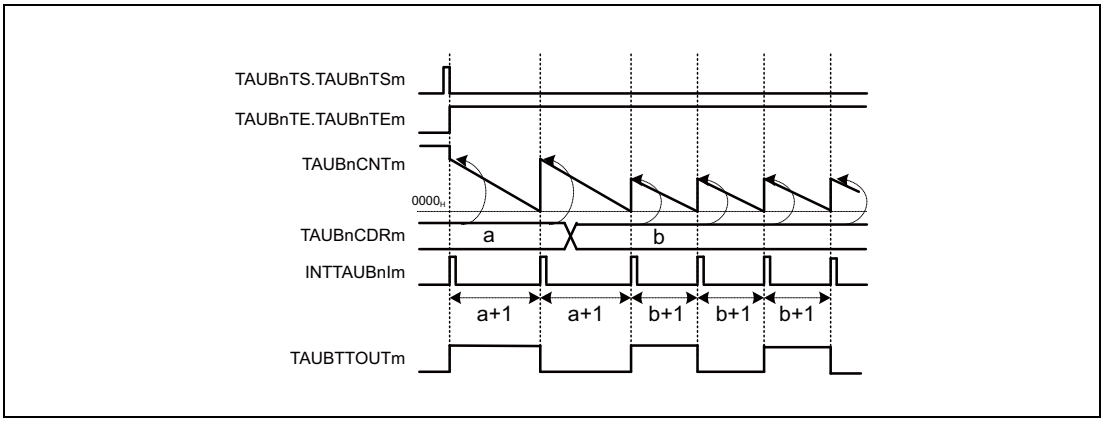


Figure 29.25 General Timing Diagram for Interval Timer Function

29.12.1.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.39 Contents of the TAUBnCMORm Register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm is not generated and TAUBTTOUTm does not toggle at operation start. 1: INTTAUBnIm is generated and TAUBTTOUTm toggles at operation start or restart.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.40 Contents of the TAUBnCMURm Register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode

Table 29.41 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	Write 0_B.
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUBnTOE.TAUBnTOEm = 0. TAUBTTOUTm can then be controlled independently of the interrupts.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the Interval Timer Function. Therefore, these registers must be set to 0.

Table 29.42 Simultaneous Rewrite Settings for Interval Timer Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.1.5 Operating Procedure for Interval Timer Function

Table 29.43 Operating Procedure for Interval Timer Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in Table 29.39, Contents of the TAUBnCMORm Register for Interval Timer Function and **Table 29.40, Contents of the TAUBnCMURm Register for Interval Timer Function** Set the value of the TAUBnCDRm register Set the channel output mode by setting the control bits as described in Table 29.41, Control Bit Settings for Independent Channel Output Mode 1	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated and TAUBTTOUTm toggles.
During operation	The TAUBnCDRm register value can be changed at any time. The TAUBnCNTm register can be read at all times.	TAUBnCNTm counts down. When the counter reaches 0000_H: • TAUBnCNTm reloads the TAUBnCDRm value and continues count operation • INTTAUBnIm is generated and TAUBTTOUTm toggles.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

29.12.1.6 Specific Timing Diagrams

(1) TAUBnCDRm = 0000_H, count clock = PCLK/2

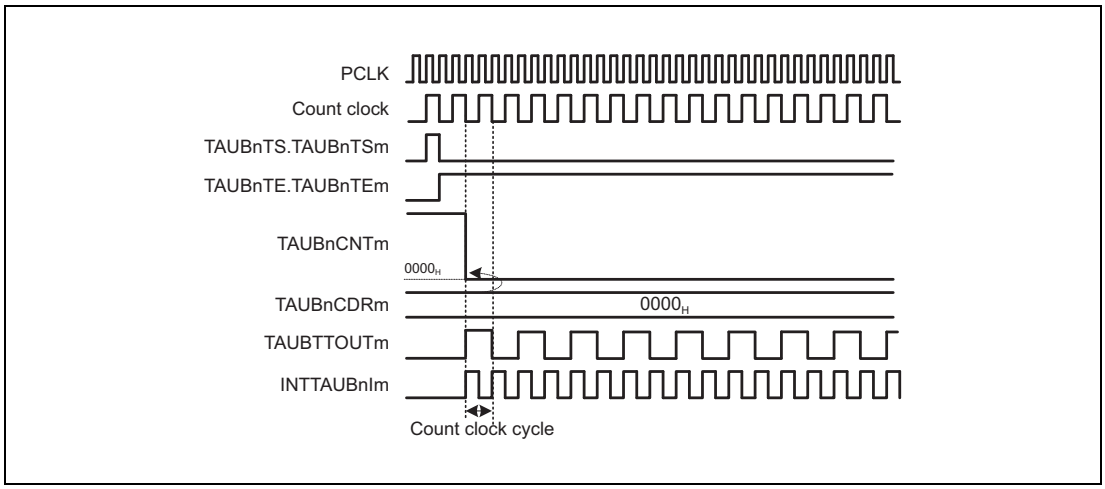


Figure 29.26 TAUBnCDRm = 0000_H, Count Clock = PCLK/2

- TAUBnCDRm = 0000_H, and the count clock = PCLK/2, the TAUBnCDRm value is written to TAUBnCNTm every count clock, meaning that TAUBnCNTm is always 0000_H.
- INTTAUBnIm is generated every count clock, resulting in TAUBTTOUTm toggling every count clock.

(2) TAUBnCDRm = 0000_H, count clock = PCLK

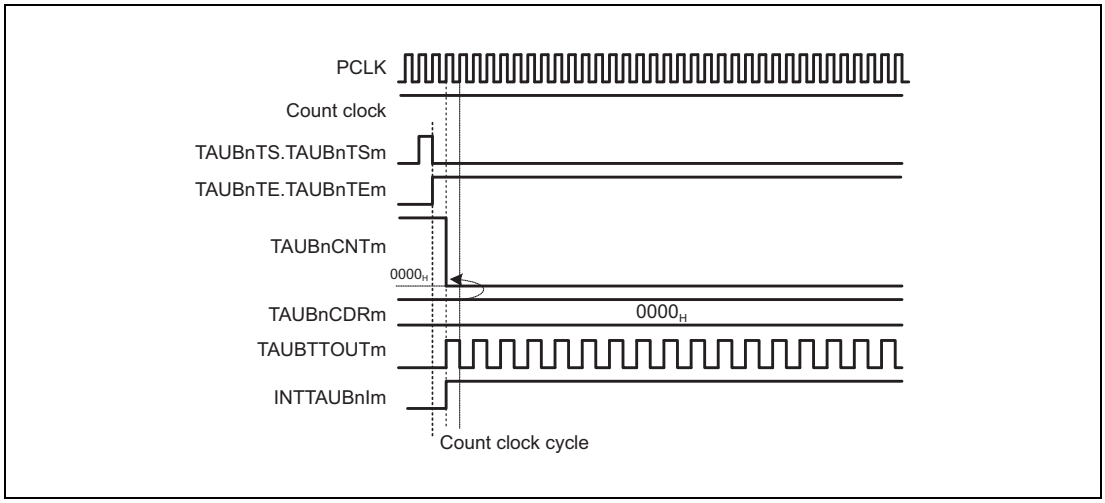


Figure 29.27 TAUBnCDRm = 0000_H, Count Clock = PCLK

- TAUBnCDRm = 0000_H, and the count clock = PCLK, the TAUBnCDRm value is written to TAUBnCNTm every PCLK clock, meaning that TAUBnCNTm is always 0000_H.
- INTTAUBnIm is fixed to the high level. Though the first interrupt is generated, subsequent interrupts are not generated. TAUBTTOUTm is toggling every PCLK clock.

(3) Operation stop and restart (TAUBnCMORm TAUBnMD0 = 1)

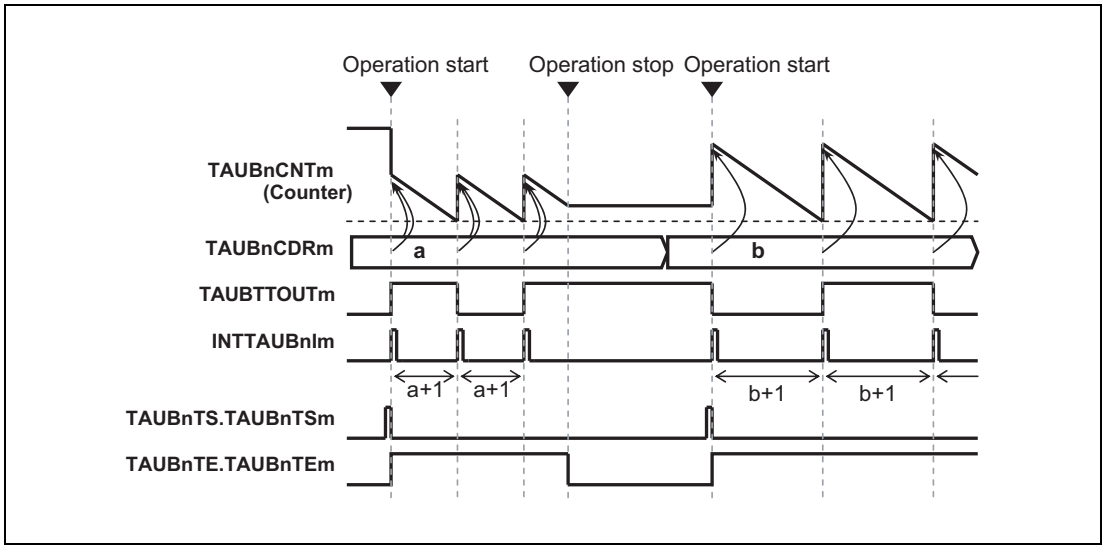


Figure 29.28 Operation Stop and Restart, TAUBnCMORm.TAUBnMD0 = 1

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm and TAUBTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1.

(4) Operation stop and restart (TAUBnCMORm.TAUBnMD0 = 0)

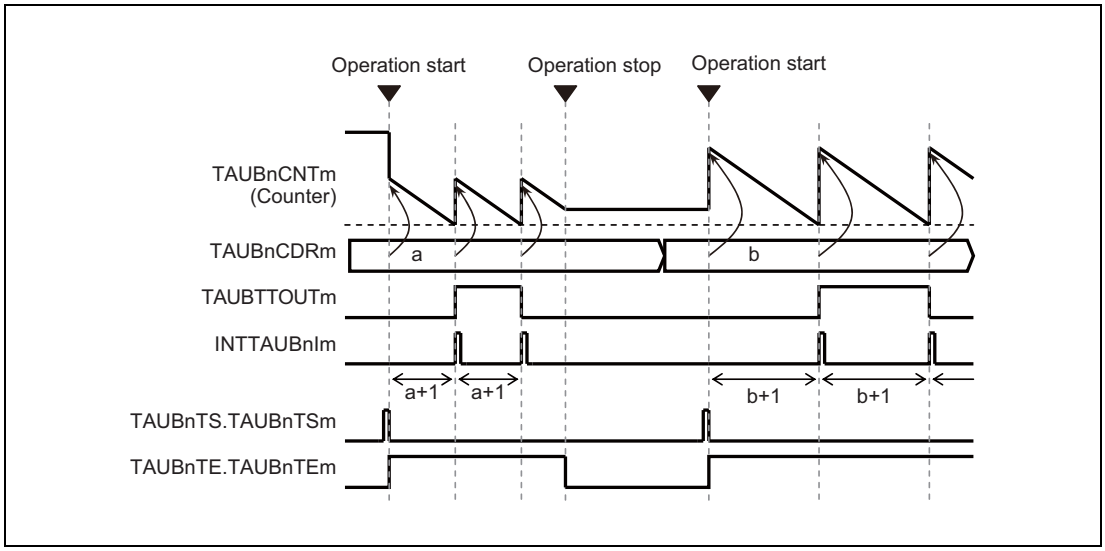


Figure 29.29 Operation Stop and Restart, TAUBnCMORm.TAUBnMD0 = 0

(5) Forced restart (TAUBnCMORm.TAUBnMD0 = 1)

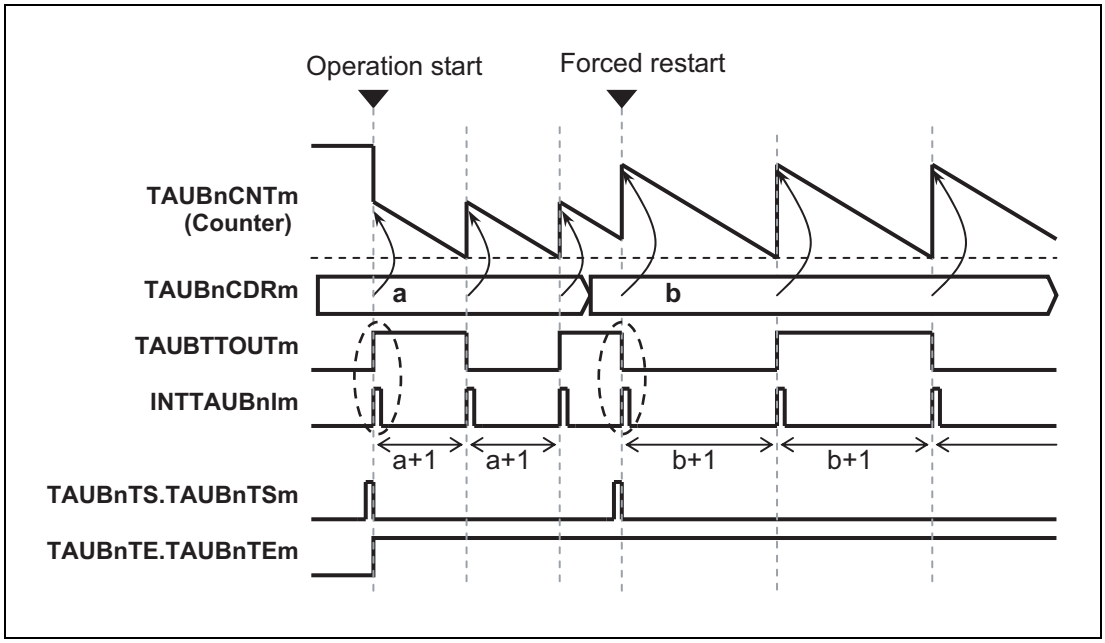


Figure 29.30 Forced Restart Operation, TAUBnCMORm.TAUBnMD0 = 1

- The counter can be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm to 1 during operation.
- If the TAUBnCMORm.TAUBnMD0 bit is set to 1, an interrupt at start or restart is generated and the output TAUBTTOUTm toggles.

(6) Forced restart (TAUBnCMORm.TAUBnMD0 = 0)

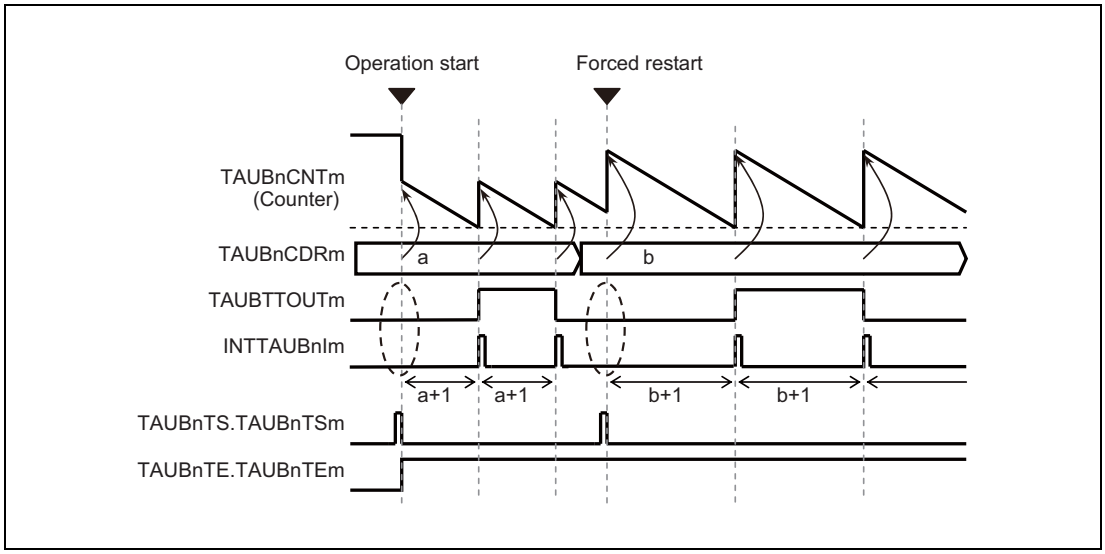


Figure 29.31 Forced Restart Operation (TAUBnCMORm.TAUBnMD0 = 0)

- The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm = 1 during operation.
- If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUBTTOUTm does not toggle.

29.12.2 TAUBTTINm Input Interval Timer Function

29.12.2.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUBnIm) at regular intervals or when a valid TAUBTTINm input edge is detected. When an interrupt is generated, the TAUBTTOUTm signal toggles, resulting in a square wave.

Description

The counter is started by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The current value of TAUBnCDRm is loaded to TAUBnCNTm and the counter starts to count down from this value.

INTTAUBnIm is generated when the counter reaches 0000H or by an effective TAUBTTINm input edge. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm stop but retain their values. The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm to 1 during operation.

The type of edge used as the trigger is specified using the TAUBnCMURm.TAUBnTIS[1:0] bits.

Either rising edge, falling edge, or rising and falling edges can be selected.

29.12.2.2 Equations

INTTAUBnIm cycle = count clock cycle × (TAUBnCDRm + 1)

TAUBTTOUTm square wave cycle = count clock cycle × (TAUBnCDRm + 1) × 2

29.12.2.3 Block Diagram and General Timing Diagram

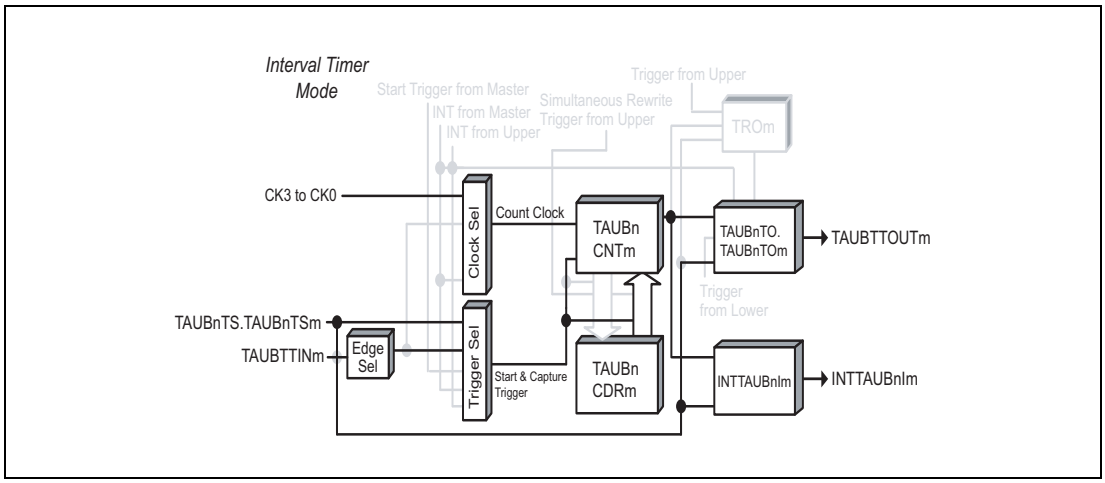


Figure 29.32 Block Diagram for TAUBTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1).
- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

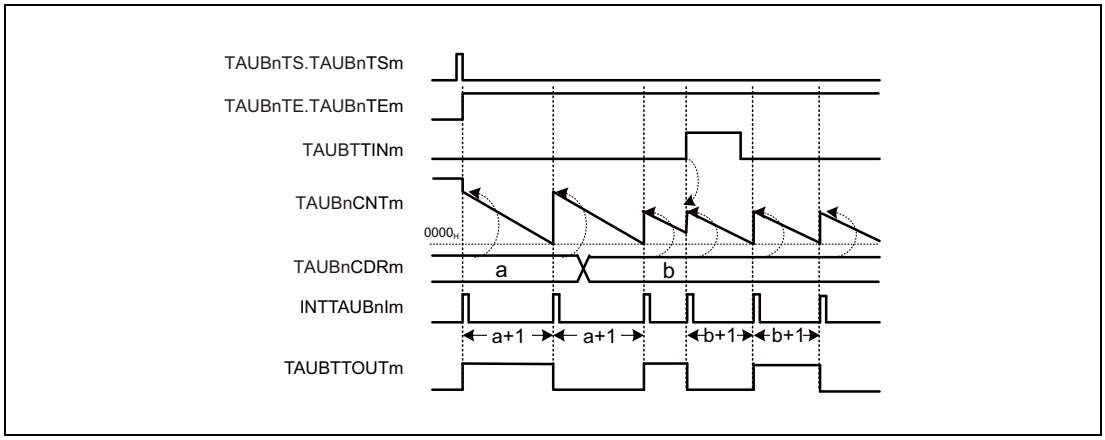


Figure 29.33 General Timing Diagram for TAUBTTINm Input Interval Timer Function

29.12.2.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.44 Contents of the TAUBnCMORm Register for TAUBTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm not generated and TAUBTTOUTm does not toggle at operation start 1: Generates INTTAUBnIm and toggles TAUBTTOUTm at operation start

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.45 Contents of the TAUBnCMURm Register for TAUBTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

Table 29.46 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	Write 0_B.
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUBnTOE.TAUBnTOEm = 0. TAUBTTOUTm can then be controlled independently of the interrupts.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Interval Timer Function. Therefore, these registers must be set to 0.

Table 29.47 Simultaneous Rewrite Settings for TAUBTTINm Input Interval Timer Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.2.5 Operating Procedure for TAUBTTINm Input Interval Timer Function

Table 29.48 Operating Procedure for TAUBTTINm Input Interval Timer Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in Table 29.44, Contents of the TAUBnCMORm Register for TAUBTTINm Input Interval Timer Function and Table 29.45, Contents of the TAUBnCMURm Register for TAUBTTINm Input Interval Timer Function Set the value of the TAUBnCDRm register Set the channel output mode by setting the control bits as described in Table 29.46, Control Bit Settings for Independent Channel Output Mode 1	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated and TAUBTTOUTm toggles.
During operation	The values of the TAUBnCMURm.TAUBnTIS[1:0] and the TAUBnCDRm register can be changed at any time. The TAUBnCNTm register can be read at all times. Detection of TAUBTTINm edge	TAUBnCNTm counts down. When the counter reaches 0000_H: • TAUBnCNTm reloads the TAUBnCDRm value and continues count operation • INTTAUBnIm is generated and TAUBTTOUTm toggles When a TAUBTTINm input valid edge is detected during count operation, TAUBnCNTm reloads the TAUBnCDRm value and continues count operation. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1 TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

29.12.2.6 Specific Timing Diagrams

The timing diagrams in Section 29.12.1, Interval Timer Function apply, and in addition the counter can also be restarted by an effective TAUBTTINm input edge.

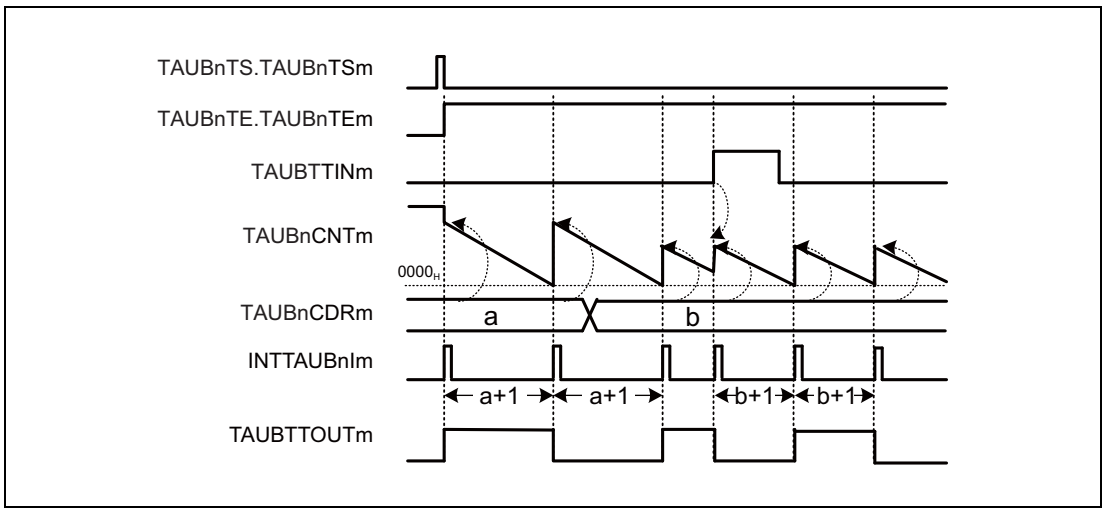


Figure 29.34 Counter Triggered by Rising TAUBTTINm Input Edge (TAUBnCMURm.TAUBnTIS[1:0] = 01_B), TAUBnCMORm.TAUBnMD0 = 1

- If a valid TAUBTTINm input edge is detected, an interrupt is generated which causes TAUBTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUBnCMURm.TAUBnTIS[1:0] = 01_B).

29.12.3 Clock Divide Function

29.12.3.1 Overview

Summary

This function is used as a frequency divider. The frequency of the input signal TAUBTTINm is divided by a factor related to TAUBnCDRm, and the resulting signal is output to TAUBTTOUTm.

Prerequisites

- TAUBTTINm must have a fixed frequency
- The operation mode must be set to interval timer mode, see **Table 29.49, Contents of the TAUBnCMORm Register for Clock Divide Function**
- The channel output mode must be set to independent channel output mode 1.

Description

The counter is started by setting the channel trigger bit (TAUBnTS.TSm) to 1.
This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value, using TAUBTTINm as the count clock.

When the counter value reaches 0000_H, INTTAUBnIm is generated and the TAUBTTOUTm signal toggles. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the function starts to count down.

The counter can be stopped by setting TAUBnTT.TAUBnTTm = 1, which in turn sets TAUBnTE.TAUBnTEm = 0. TAUBnCNTm and TAUBTTOUTm stop but retain their values. The function can be restarted by setting TAUBnTS.TAUBnTSm = 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm = 1 during operation.

Conditions

If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUBTTOUTm does not toggle. This results in an inverted TAUBTTOUTm signal compared to when TAUBnCMORm.TAUBnMD0 is set to 1.

NOTE

The TAUBTTINm input signal is sampled at the frequency of the operation clock, specified by TAUBnCMORm.TAUBnCKS[1:0] bits. As a result, the output cycle of TAUBTTOUTm has an error of ± 1 operation clock cycle.

29.12.3.2 Equations

- When rising edge detection is selected:
  TAUBTTOUTm frequency = TAUBTTINm frequency /
  [(TAUBnCDRm + 1) × 2]
- When falling edge detection is selected:
  TAUBTTOUTm frequency = TAUBTTINm frequency /
  [(TAUBnCDRm + 1) × 2]
- When rising and falling edge detection is selected:
  TAUBTTOUTm frequency = TAUBTTINm frequency / (TAUBnCDRm + 1)

29.12.3.3 Block Diagram and General Timing Diagram

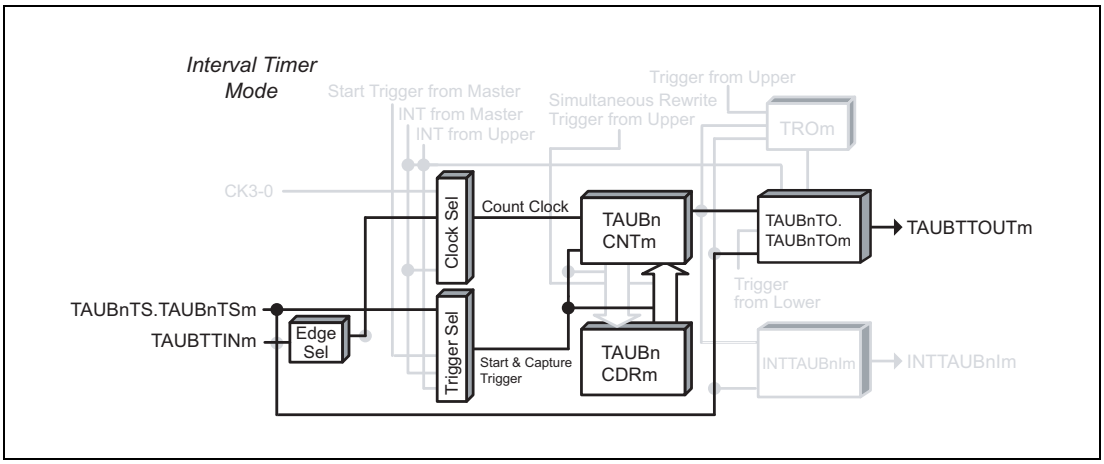


Figure 29.35 Block Diagram for Clock Divide Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

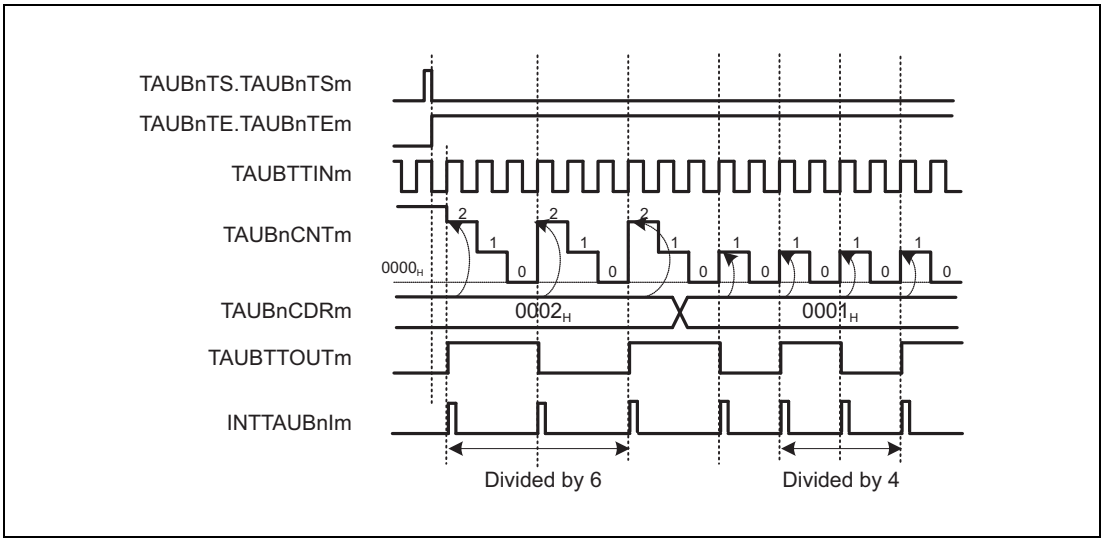


Figure 29.36 General Timing Diagram for Clock Divide Function

29.12.3.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.49 Contents of the TAUBnCMORm Register for Clock Divide Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 1_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm not generated and TAUBTTOUTm does not toggle at operation start 1: Generates INTTAUBnIm and toggles TAUBTTOUTm at operation start

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.50 Contents of the TAUBnCMURm Register for Clock Divide Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

Table 29.51 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	Write 0_B.
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUBnTOE.TAUBnTOEm = 0. TAUBTTOUTm can then be controlled independently of the interrupts.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the Clock Divide Function. Therefore, these registers must be set to 0.

Table 29.52 Simultaneous Rewrite Settings for Clock Divide Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.3.5 Operating Procedure for Clock Divide Function

Table 29.53 Operating Procedure for Clock Divide Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in Table 29.49, Contents of the TAUBnCMORm Register for Clock Divide Function and **Table 29.50, Contents of the TAUBnCMURm Register for Clock Divide Function** Set the value of the TAUBnCDRm register Set the channel output mode by setting the control bits as described in **Table 29.50, Contents of the TAUBnCMURm Register for Clock Divide Function**	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 is set to 1, INTTAUBnIm is generated and TAUBTTOUTm toggles.
During operation	The value of TAUBnCDRm can be changed at any time. The TAUBnCNTm register can be read at all times.	When a TAUBTTINm input edge is detected, TAUBnCNTm counts down. When the counter reaches 0000_H: • TAUBnCNTm loads the TAUBnCDRm value and continues count operation • INTTAUBnIm is generated • TAUBTTOUTm toggles. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

29.12.3.6 Specific Timing Diagrams

(1) TAUBnCDRm = 0000_H

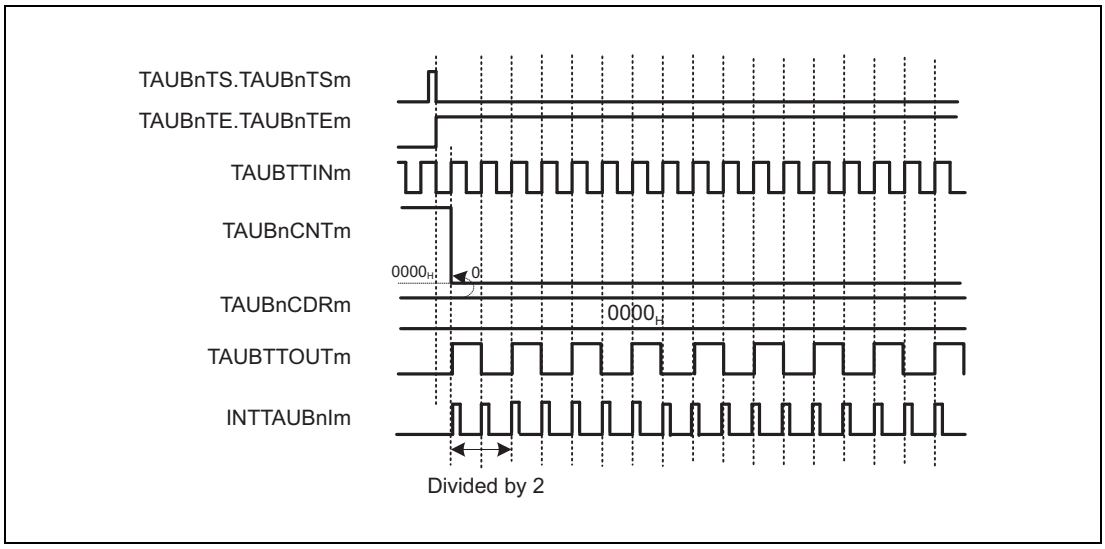


Figure 29.37 TAUBnCDRm = 0000_H, TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01_B

- If TAUBnCDRm is 0000_H, TAUBnCNTm is also always 0000_H.
- INTTAUBnIm is generated every count clock, resulting in TAUBTTOUTm toggling every count clock.

Figure 29.37 is an image of the operation timing. Actually, the delay time caused by the noise filter and the synchronization circuit between the TAUBnIm terminal and TAUBn will cause a delay from the TINm detection to the TOUTm output.

(2) Restart

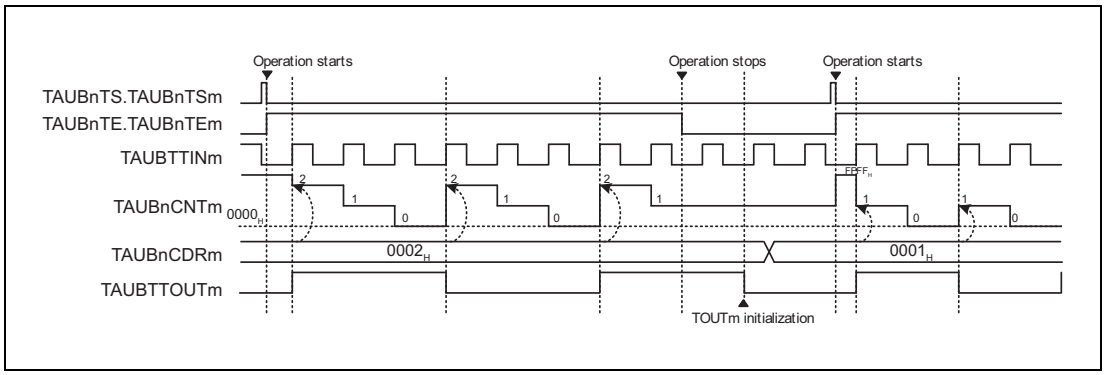


Figure 29.38 Restart (TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

To reset the value of TAUBTTOUTm:

- Set TAUBnTOE.TAUBnTOEm = 0 when the counter is stopped (TAUBnTE.TAUBnTEm = 0)
- Then write either 0 or 1 to TAUBnTO.TAUBnTOm to set the new start value of TAUBTTOUTm

(3) Forced restart

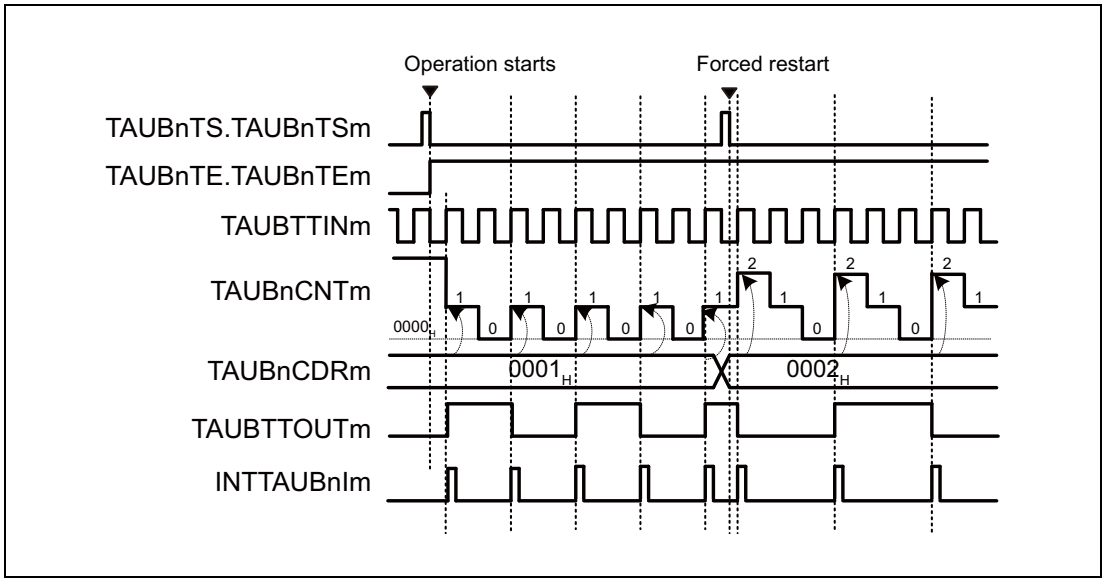


Figure 29.39 Forced Restart, (TAUBnCMORm.TAUBnMD0 = 1, TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

To reset the value of TAUBTTOUTm.

- The counter can be forcibly restarted (without stopping it first) by setting TAUBnTS.TAUBnTSm = 1 during operation.
- The value of TAUBnCDRm is written to TAUBnCNTm and the count operation restarts.
- TAUBTTOUTm restarts at the same level as before the forced restart.

29.12.4 External Event Count Function

29.12.4.1 Overview

Summary

This function is used as an event timer. It generates an interrupt (INTTAUBnIm) when a specific number of valid edges of TAUBTTINm input are detected.

Prerequisites

- The operation mode must be set to event count mode, see **Table 29.54, Contents of the TAUBnCMORm Register for External Event Count Function**
- TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. When the counter starts, the current value of TAUBnCDRm is written to TAUBnCNTm.

When a valid TAUBTTINm input edge is detected, the value of TAUBnCNTm reduces by 1. TAUBnCNTm retains this value until a valid TAUBTTINm input edge is detected or the counter is restarted.

When effective edges are detected (TAUBnCDRm + 1) times, INTTAUBnIm is generated. TAUBnCNTm then loads the TAUBnCDRm value and subsequently continues to operate.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0. The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. The counter can also be restarted without stopping it first (forced restart) by setting TAUBnTS.TAUBnTSm to 1 during operation.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as the trigger is specified by the TAUBnCMURm.TAUBnTIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

29.12.4.2 Equations

Number of valid edges,
detected before INTTAUBnIm is generated = TAUBnCDRm + 1

29.12.4.3 Block Diagram and General Timing Diagram

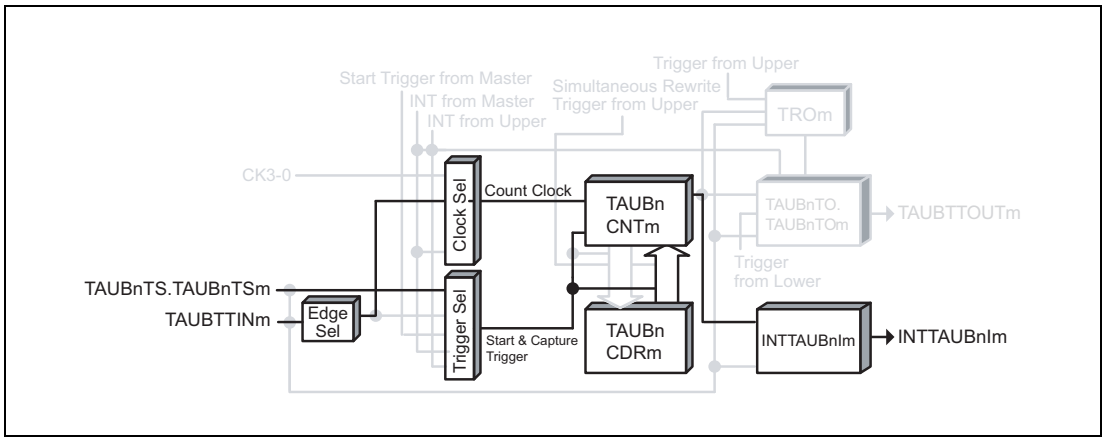


Figure 29.40 Block Diagram for External Event Count Function

The following settings apply to the general timing diagram.

- Rising edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

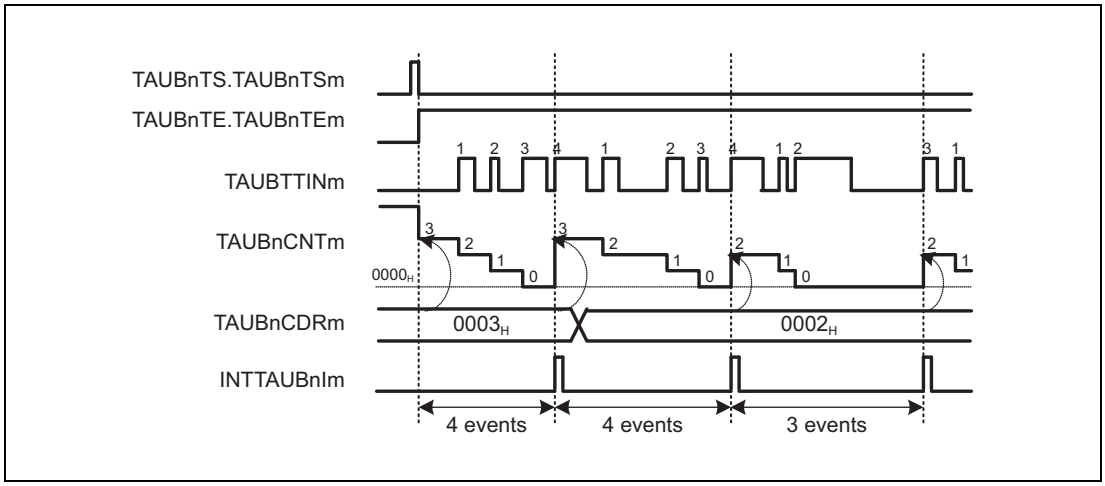


Figure 29.41 General Timing Diagram for External Event Count Function

29.12.4.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.54 Contents of the TAUBnCMORm Register for External Event Count Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 1_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0011_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.55 Contents of the TAUBnCMURm Register for External Event Count Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge 01: Rising edge 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the External Event Count Function.

Therefore, these registers must be set to 0.

Table 29.56 Simultaneous Rewrite Settings for External Event Count Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.4.5 Operating Procedure for External Event Count Function

Table 29.57 Operating Procedure for External Event Count Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in Table 29.54, Contents of the TAUBnCMORm Register for External Event Count Function and Table 29.55, Contents of the TAUBnCMURm Register for External Event Count Function Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value and waits for detection of the TAUBTTINm input edge.
During operation	Detection of TAUBTTINm edges. The value of TAUBnCDRm can be changed at any time. The TAUBnCNTm register can be read at any time.	TAUBnCNTm performs count-down operation each time a TAUBTTINm input edge is detected. When the effective edges are detected (TAUBnCDRm + 1) times: • TAUBnCNTm loads the TAUBnCDRm value and continues count operation • INTTAUBnIm is generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

29.12.4.6 Specific Timing Diagrams

(1) TAUBnCDRm = 0000_H

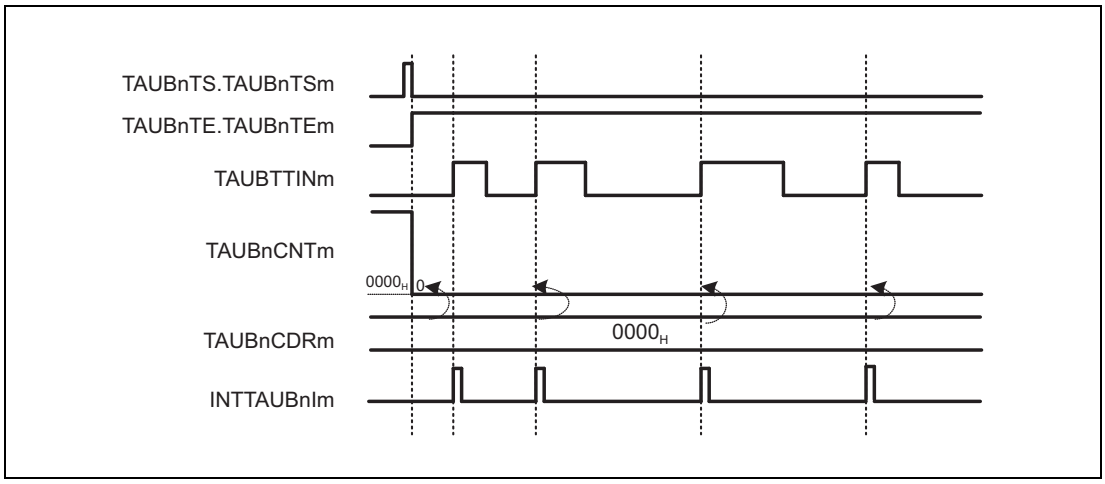


Figure 29.42 TAUBnCDRm = 0000_H, TAUBnCMURm.TAUBnTIS[1:0] = 01_B

- If 0000_H = TAUBnCDRm, 0000_H is loaded to TAUBnCNTm every time a valid TAUBTTINm input edge is detected.

  This means, INTTAUBnIm is generated every time a valid TAUBTTINm input edge is detected.

(2) Operation stop and restart

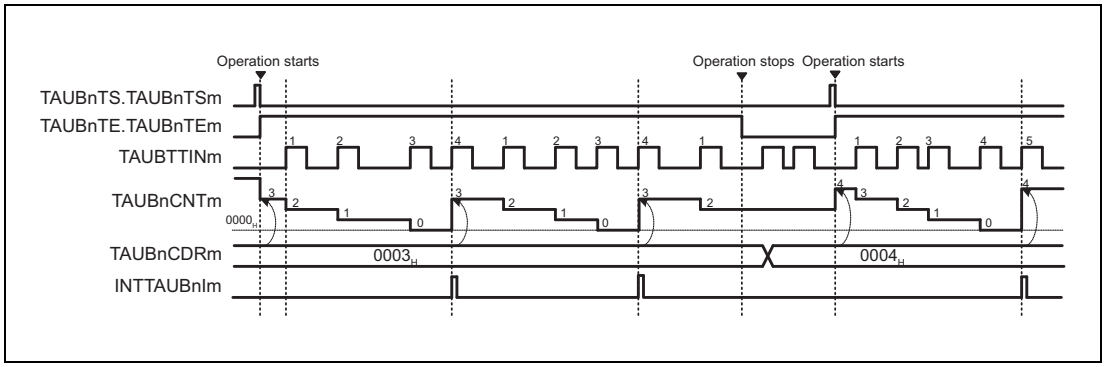


Figure 29.43 Operation Stop and Restart, (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm stops and the current value is retained. TAUBTTINm continues and TAUBnCNTm ignores the valid edge.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. TAUBnCNTm loads the TAUBnCDRm value and restarts count operation.

(3) Forced restart

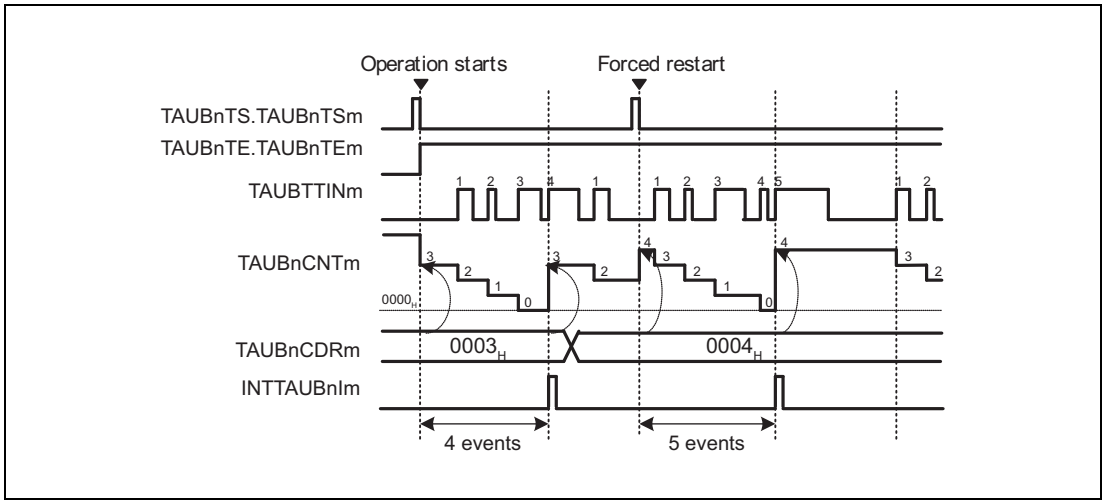


Figure 29.44 Forced Restart, (TAUBnCMURm.TAUBnTIS[1:0] = 01_B)

A forced restart applies the new TAUBnCDRm value to TAUBnCNTm immediately.

- The counter can be restarted (without stopping it first), by setting TAUBnTS.TAUBnTSm to 1 during operation.
- The value of TAUBnCDRm is loaded to TAUBnCNTm and the counter awaits the next valid TAUBTTINm input edge.

29.12.5 One-Pulse Output Function

29.12.5.1 Overview

Summary

This function generates an interrupt (INTTAUBnIm) when a valid TAUBTTINm input edge is detected and continues to generate interrupts at the specified interval.

TAUBTTINm input signal pulses that occur within the defined interval are ignored. When an interrupt is generated, the TAUBTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode must be set to pulse one-count mode. (See Table 29.58, Contents of the TAUBnCMORm Register for One-Pulse Output Function).
- The channel output mode must be set to independent channel output mode 2. (See **Table 29.60, Control Bit Settings for Independent Channel Output Mode 2**.)
- Trigger detection must be disabled during counting (TAUBnCMORm.TAUBnMD0 = 0).

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.
This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The counter starts when a valid TAUBTTINm input edge is detected. The value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from the TAUBnCDRm value. An interrupt is generated and TAUBTTOUTm is set to the active level.

When the counter reaches 0001_H an interrupt is generated and TAUBTTOUTm is set to the active level. The counter stops at 0000_H and awaits the next valid TAUBTTINm input edge.

When the counter is counting down, further TAUBTTINm input signals are ignored, i.e. the counter does not reset.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the counter starts to count down.

Conditions

The type of edge used as the trigger is specified by the TAUBnCMURm.TIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

29.12.5.2 Equations

Interval between TAUBTTINm and INTTAUBnIm = TAUBTTOUTm (timer output) width = count clock cycle × TAUBnCDRm

29.12.5.3 Block Diagram and General Timing Diagram

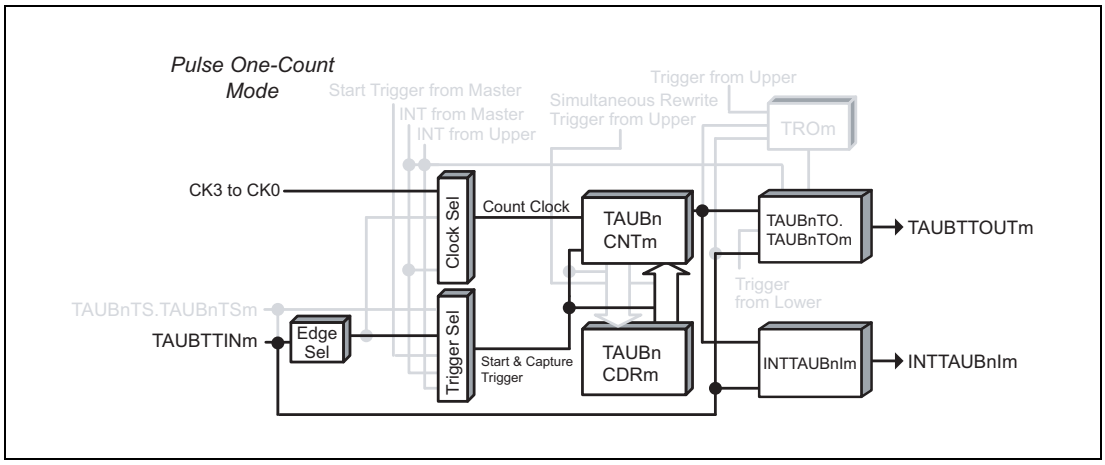


Figure 29.45 Block Diagram for One-Pulse Output Function

The following settings apply to the general timing diagram.

- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

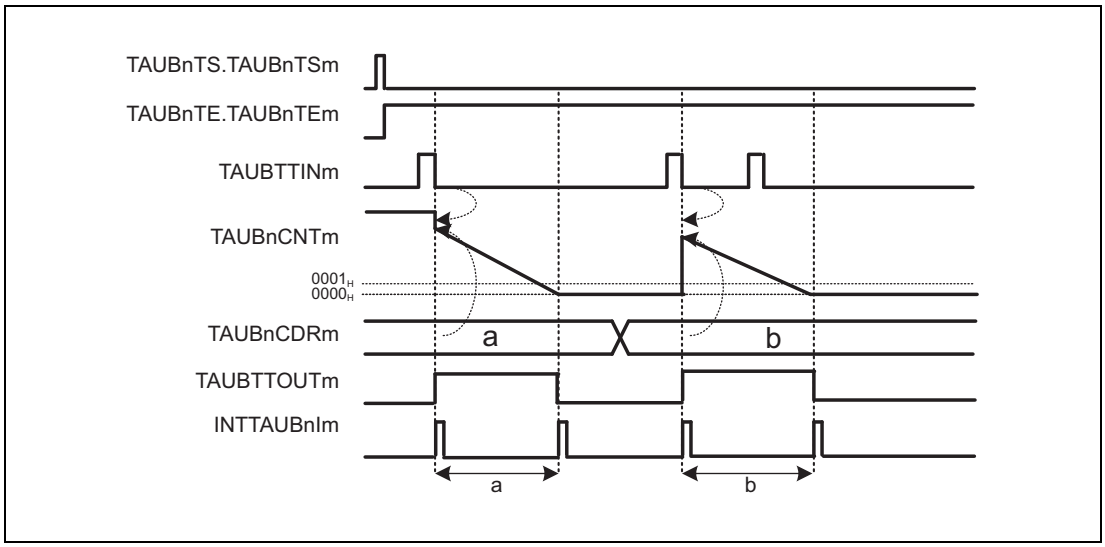


Figure 29.46 General Timing Diagram for One-Pulse Output Function

29.12.5.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.58 Contents of the TAUBnCMORm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1010_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.59 Contents of the TAUBnCMURm Register for One-Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

Table 29.60 Control Bit Settings for Independent Channel Output Mode 2

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUBnTOE.TAUBnTOEm = 0. TAUBTTOUTm can then be controlled independently of the interrupts.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the One-Pulse Output Function.
Therefore, these registers must be set to 0.

Table 29.61 Simultaneous Rewrite Settings for One-Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.5.5 Operating Procedure for One-Pulse Output Function

Table 29.62 Operating Procedure for One-Pulse Output Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in Table 29.58, Contents of the TAUBnCMORm Register for One-Pulse Output Function and Table 29.59, Contents of the TAUBnCMURm Register for One-Pulse Output Function Set the value of the TAUBnCDRm register Set the channel output mode by setting the control bits as described in Table 29.60, Control Bit Settings for Independent Channel Output Mode 2	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUBTTINm start edge	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge. When a start edge is detected, TAUBnCNTm loads the TAUBnCDRm value.
During operation	The value of TAUBnCDRm can be changed at any time. The TAUBnCNTm register can be read at all times.	INTTAUBnIm is generated when TAUBnCNTm starts and TAUBTTOUTm is set to its active level. TAUBnCNTm counts down. When the counter reaches 0001_H: • INTTAUBnIm is generated • TAUBTTOUTm is set to its inactive level. TAUBnCNTm stops counting and waits for a trigger. If a trigger occurs while TAUBnCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

29.12.6 TAUBTTINm Input Pulse Interval Measurement Function

29.12.6.1 Overview

Summary

This function captures the count value and uses this value and the overflow bit TAUBnCSRm.TAUBnOVF to measure the interval of the TAUBTTINm input signal.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is started by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.
This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The counter TAUBnCNTm starts counting up from 0000_H. When a valid TAUBTTINm edge is detected, the value of TAUBnCNTm is captured, transferred to TAUBnCDRm, and an interrupt INTTAUBnIm is generated. The counter resets to 0000_H and subsequently continues operation.

If the counter reaches $FFFF_H$ before a valid TAUBTTINm edge is detected, it overflows to 0000_H. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUBnCDRm and TAUBnCSRm.TAUBnOVF respectively depend on the values of bits TAUBnCMORm.TAUBnCOS[1:0].

Table 29.63 Effects of an Overflow

TAUBnCMORm.COS[1:0]	When Overflow Occurs		When a Valid TAUBTTINm Input is then Detected	
	TAUBnCDRm	TAUBnCSRm.TAUBnOVF	TAUBnCDRm, TAUBnCNTm	TAUBnCSRm.TAUBnOVF
00	Unchanged	0	TAUBnCNTm loaded to TAUBnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUBnCNTm set to 0, TAUBnCDRm unchanged	Unchanged
11		1		

If TAUBnCMORm.TAUBnCOS[0] is 1, the overflow bit TAUBnCSRm.TAUBnOVF can only be cleared by a CPU command that sets TAUBnCSCm.TAUBnCLOV = 1.

The combination of the value of TAUBnCDRm and TAUBnCSRm.TAUBnOVF can be used to deduce the interval of the TAUBTTINm signal. However, if an overflow occurs multiple times before a valid TAUBTTINm input is detected, the overflow bit TAUBnCSRm.TAUBnOVF cannot indicate this.

The function can be stopped by setting TAUBnTT.TAUBnTTm = 1, which in turn sets TAUBnTE.TAUBnTEm = 0. TAUBnCNTm stops but retains its value. While the function is stopped, TAUBTTINm input valid edge detection and TAUBnCNTm capture are not performed.

The counter is reset to 0000_H and subsequently continues operation.

Conditions

If the TAUBnCMORm.TAUBnMD0 bit is set to 0, the interrupt at start or restart is not generated.

NOTE

When TAUBnCMORm.TAUBnCOS[1:0] = 10_B or 11_B, the value of TAUBnCNTm is not written to TAUBnCDRm when the first valid TAUBTTINm input edge occurs after an overflow. However, an interrupt is generated.

29.12.6.2 Equations

TAUBTTINm input pulse interval = count clock cycle ×

[(TAUBnCSRm.TAUBnOVF × ($FFFF_H$ + 1)) + TAUBnCDRm capture value + 1]

29.12.6.3 Block Diagram and General Timing Diagram

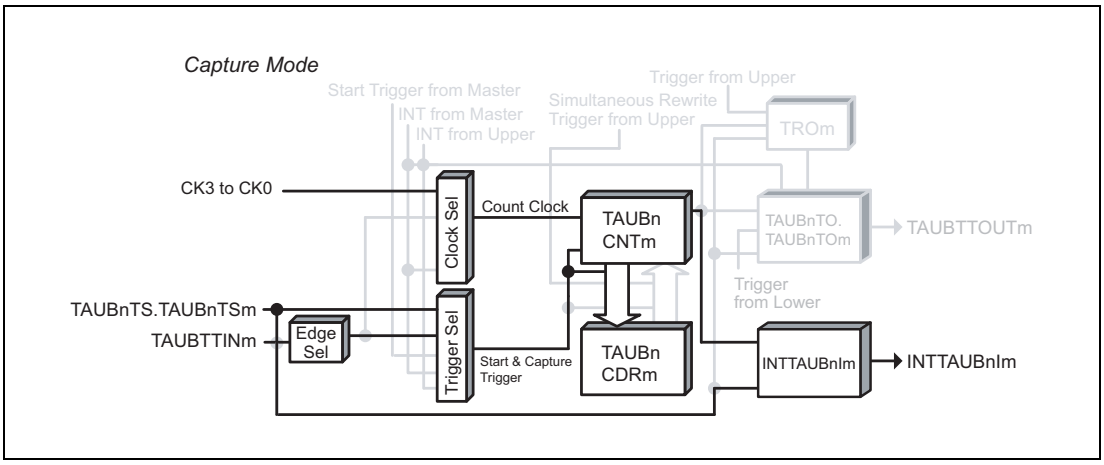


Figure 29.47 Block Diagram for TAUBTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is not generated at operation start (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)
- When a valid TAUBTTINm input is detected after an overflow TAUBnCDRm is changed and TAUBnCSRm.TAUBnOVF is set to 1 (TAUBnCMORm.TAUBnCOS[1:0] = 00_B)

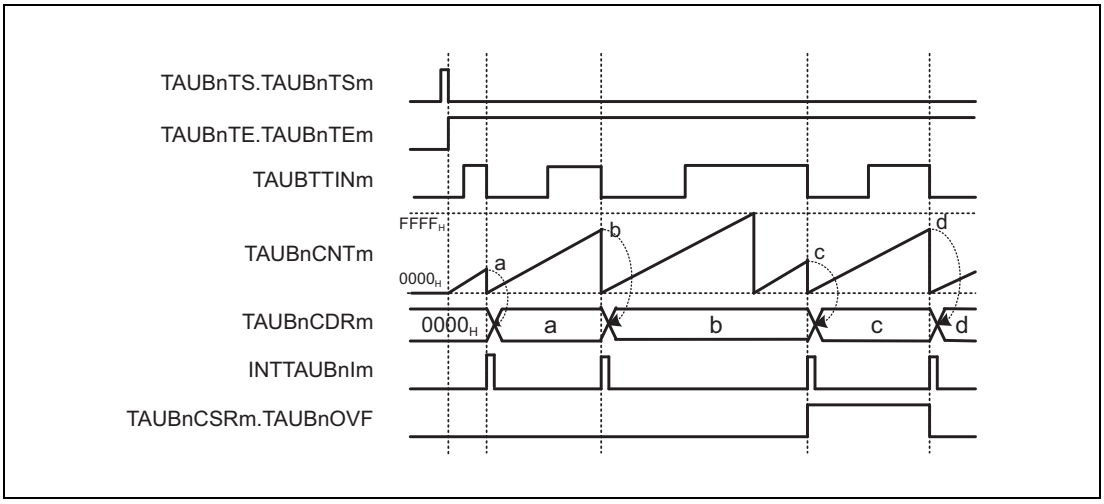


Figure 29.48 General Timing Diagram for TAUBTTINm Input Pulse Interval Measurement Function

29.12.6.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.64 Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	See **Table 29.63, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0010_B.
0	TAUBnMD0	0: INTTAUBnIm not generated at operation start 1: Generates INTTAUBnIm at operation start

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.65 Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the TAUBTTINm Input Pulse Interval Measurement Function. Therefore, these registers must be set to 0.

Table 29.66 Simultaneous Rewrite Settings for TAUBTTINm Input Pulse Interval Measurement Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.6.5 Operating Procedure for TAUBTTINm Input Pulse Interval Measurement Function

Table 29.67 Operating Procedure for TAUBTTINm Input Pulse Interval Measurement Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 29.64, Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Measurement Function** and **Table 29.65, Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Measurement Function** The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm is cleared to 0000_H. INTTAUBnIm is generated when TAUBnCMORm.TAUBnMD0 is set to 1.
During operation	Detection of TAUBTTINm edges. The values of TAUBnCMURm.TAUBnTIS[1:0] bits can be changed at any time. The TAUBnCDRm and TAUBnCSRm registers can be read at any time. The TAUBnCSCm.TAUBnCLOV bit can be set to 1. (The TAUBnCSRm.TAUBnOVF bit can be cleared to 0.	TAUBnCNTm starts to count up from 0000_H. When a TAUBTTINm valid edge is detected: • TAUBnCNTm transfers (captures) its value to TAUBnCDRm, and returns to 0000_H • INTTAUBnIm is then generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and both it and TAUBnCSRm.TAUBnOVF retain their current values.

Restart operation

29.12.6.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUBnCMORm.TAUBnCOS[1:0] = 00_B

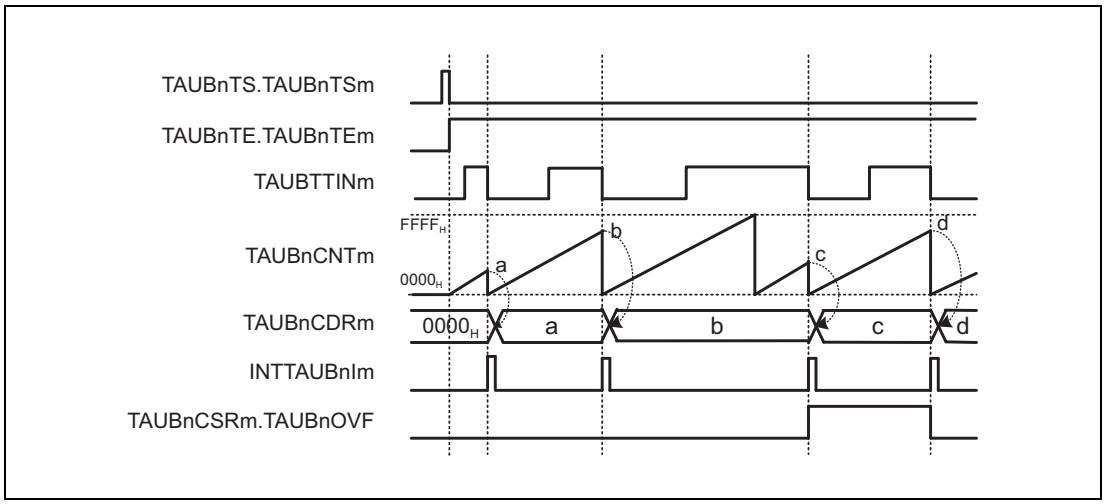


Figure 29.49 TAUBnCMORm.TAUBnCOS[1:0] = 00_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is loaded to TAUBnCDRm and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, while any overflow has not occurred, TAUBnCSRm.TAUBnOVF is cleared to 0.

(2) TAUBnCMORm.TAUBnCOS[1:0] = 01_B

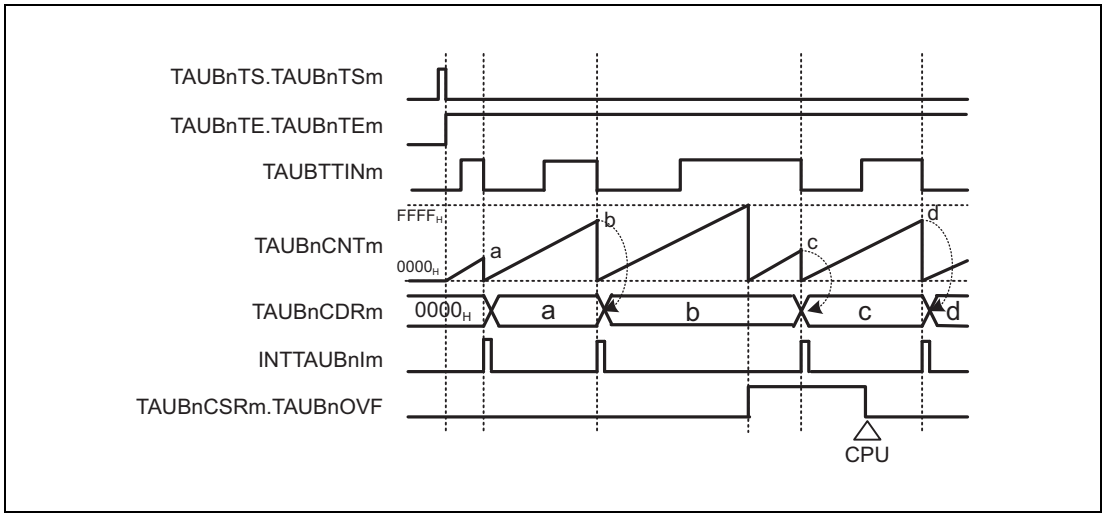


Figure 29.50 TAUBnCMORm.TAUBnCOS[1:0] = 01_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is written to TAUBnCDRm.
- TAUBnCSRm.TAUBnOVF is only cleared by a CPU command. (TAUBnCSCm.TAUBnCLOV bit = 1)

(3) TAUBnCMORm.TAUBnCOS[1:0] = 10_B

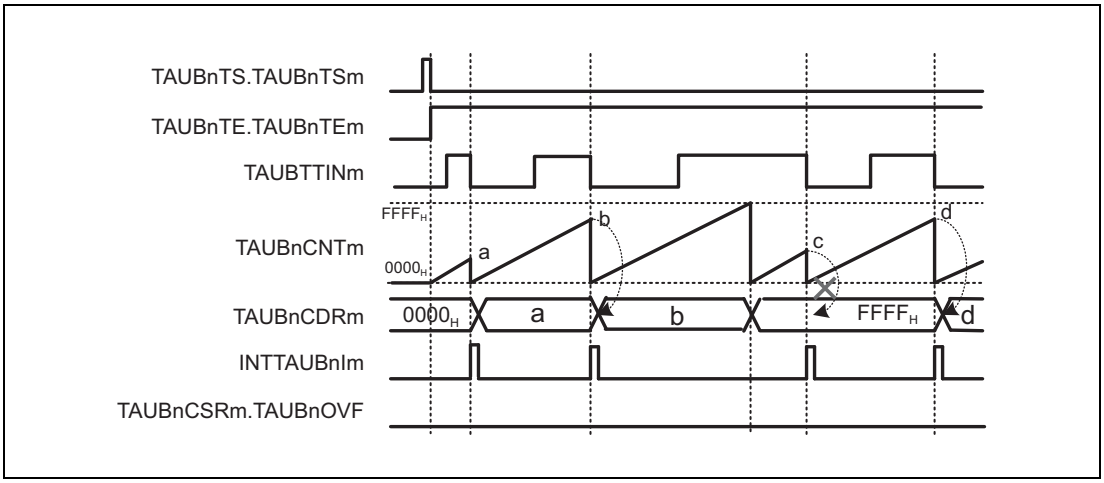


Figure 29.51 TAUBnCMORm.TAUBnCOS[1:0] = 10_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- When an overflow occurs, TAUBnCDRm is set to $FFFF_H$ and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm is reset to 0, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.

(4) TAUBnCMORm.TAUBnCOS[1:0] = 11_B

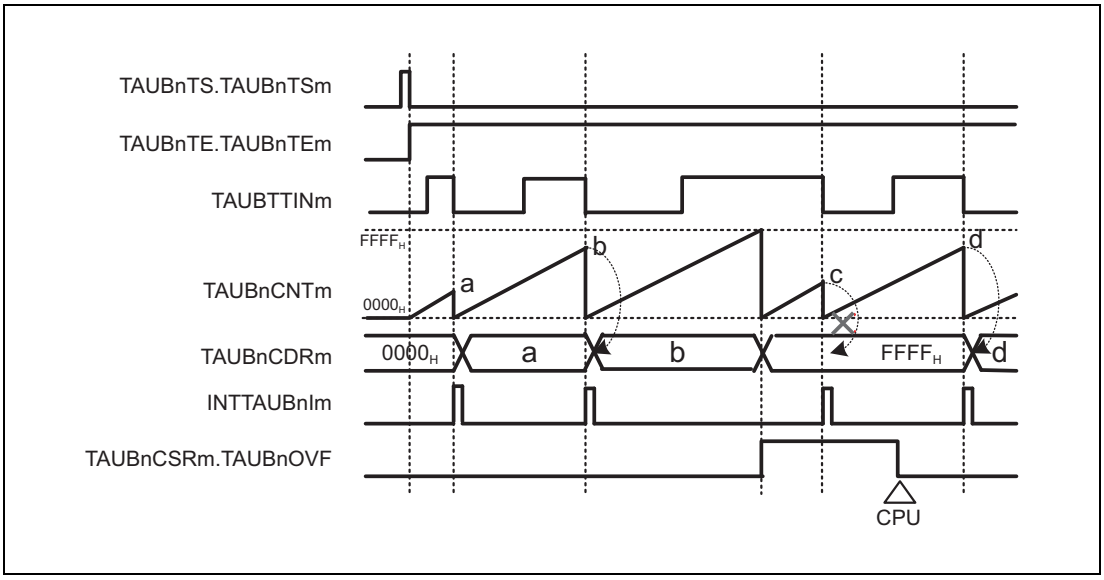


Figure 29.52 TAUBnCMORm.TAUBnCOS[1:0] = 11_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B

- When an overflow occurs, TAUBnCDRm is set to $FFFF_H$, and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm is reset to 0, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.
- TAUBnCSRm.TAUBnOVF is cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

(5) When rising and falling edge detection are selected (TAUBnCMORm.TAUBnMD0 = 1)

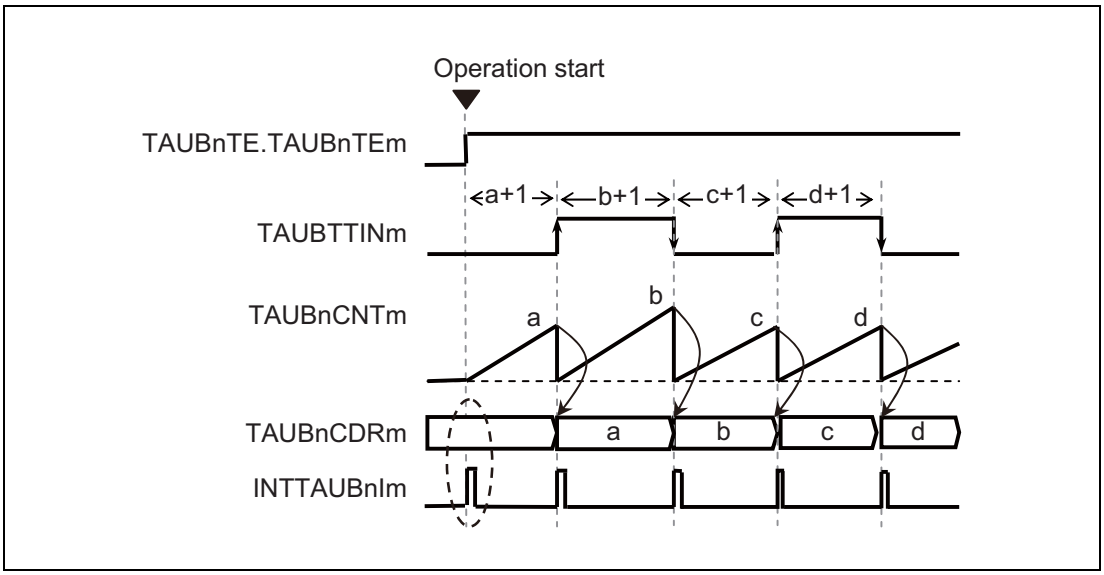


Figure 29.53 TAUBnCMORm.TAUBnMD0 = 1

Setting TAUBnCMURm.TAUBnTIS[1:0] to 10_B (detection of both edges selected) measures the TAUBTTINm rising and falling edge intervals.

(6) Operation stop and operation restart (TAUBnCMORm.TAUBnMD0 = 0)

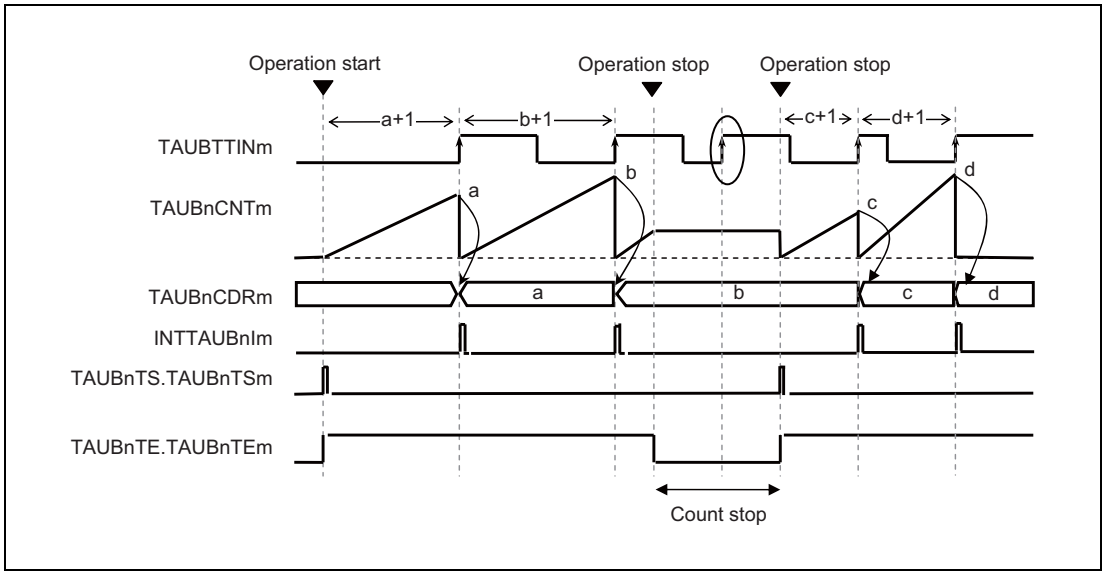


Figure 29.54 Operation Stop and Operation Restart (TAUBnCMORm.TAUBnMD0 = 0)

Setting TAUBnTT.TAUBnTTm to 1 clears TAUBnTE.TAUBnTEm to 0, which stops the count operation. At this time, TAUBnCNTm retains the status and stops.

When TAUBnTE.TAUBnTEm retains 0 (operation stopped), TAUBTTINm input is ignored (edge detection is ignored and capture operation is not performed).

Setting TAUBnTS.TAUBnTSm to 1 clears the counter to 0000_H and restarts count-up operation.

29.12.7 TAUBTTINm Input Signal Width Measurement Function

29.12.7.1 Overview

Summary

This function measures the width of a TAUBTTINm signal by starting counting on one edge of the TAUBTTINm signal and capturing the counter value on the opposite edge.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is started by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.

This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. When a valid TAUBTTINm start edge is detected, the counter TAUBnCNTm starts counting up from 0000_H. When a valid TAUBTTINm stop edge is detected, the value of TAUBnCNTm is captured, transferred to TAUBnCDRm, and an interrupt INTTAUBnIm is generated. The counter retains its value (TAUBnCDRm + 1) and awaits the next valid TAUBTTINm input start edge.

If the counter reaches $FFFF_H$ before a valid TAUBTTINm stop edge is detected, it overflows. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUBnCDRm and TAUBnCSRm.TAUBnOVF respectively depend on the values of bits TAUBnCMORm.TAUBnCOS[1:0].

Table 29.68 Effects of an Overflow

	When Overflow Occurs		When a Valid TAUBTTINm Input Stop Edge is Detected	
TAUBnCMORm.COS[1:0]	TAUBnCDRm	TAUBnCSRm.TAUBnOVF	TAUBnCDRm, TAUBnCNTm	TAUBnCSRm.TAUBnOVF
00	Unchanged	0	TAUBnCNTm written to TAUBnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUBnCNTm stops counting TAUBnCDRm unchanged	Unchanged
11		1		

If TAUBnCMORm.TAUBnCOS[0] = 1, the overflow bit TAUBnCSRm.TAUBnOVF can only be cleared by a CPU command that sets TAUBnCSCm.TAUBnCLOV = 1.

The combination of the value of TAUBnCDRm and TAUBnCSRm.TAUBnOVF can be used to deduce the width of the TAUBTTINm signal. However, if an overflow occurs multiple times before a valid TAUBTTINm input is detected, the overflow bit TAUBnCSRm.TAUBnOVF cannot indicate this.

This function cannot be forcibly restarted.

NOTE

When TAUBnCMORm.TAUBnCOS[1] = 1, the value of TAUBnCNTm is not written to TAUBnCDRm when the first valid TAUBTTINm input edge occurs after an overflow. However, an interrupt is generated.

29.12.7.2 Equations

TAUBTTINm input signal width = count clock cycle ×

[(TAUBnCSRm.OVF × ($FFFF_H$ + 1)) + TAUBnCDRm capture value + 1]

29.12.7.3 Block Diagram and General Timing Diagram

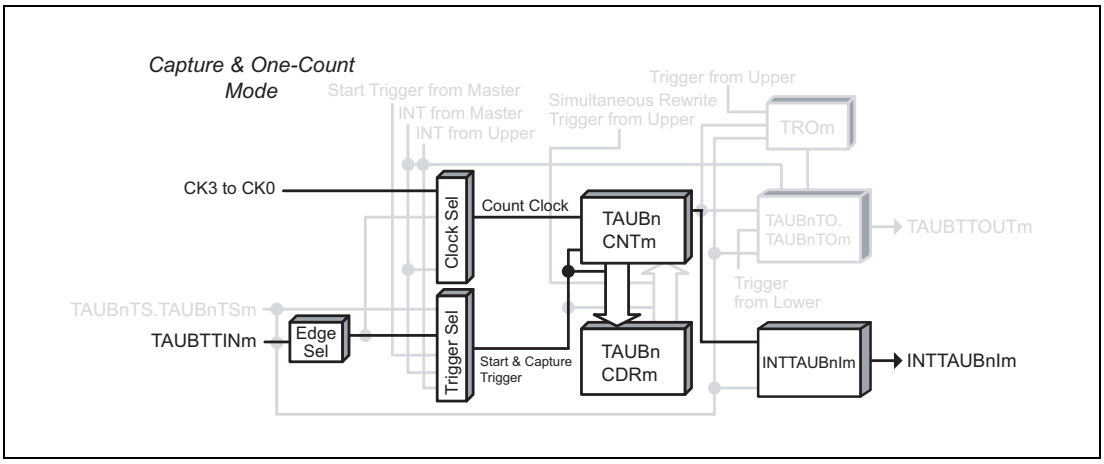


Figure 29.55 Block Diagram for TAUBTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)
- When a valid TAUBTTINm input is detected after an overflow TAUBnCDRm is changed and TAUBnCSRm.TAUBnOVF is set to 1 (TAUBnCMORm.TAUBnCOS[1:0] = 00_B)

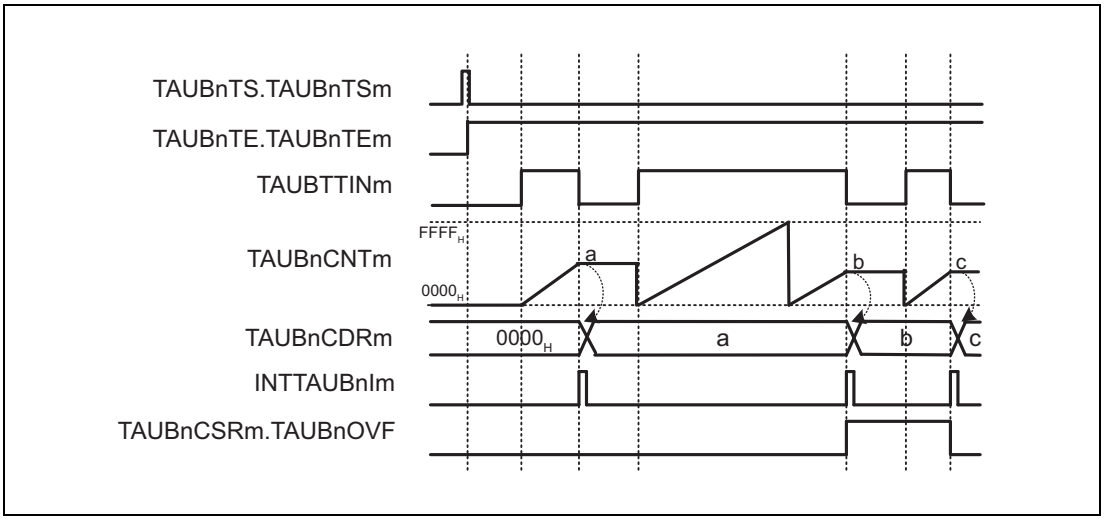


Figure 29.56 General Timing Diagram for TAUBTTINm Input Signal Width Measurement Function

29.12.7.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.69 Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	See **Table 29.68, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0110_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.70 Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Signal Width Measurement Function. Therefore, these registers must be set to 0.

Table 29.71 Simultaneous Rewrite Settings for TAUBTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.7.5 Operating Procedure for TAUBTTINm Input Signal Width Measurement Function

Table 29.72 Operating Procedure for TAUBTTINm Input Signal Width Measurement Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 29.69, Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Measurement Function** and **Table 29.70, Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Measurement Function** The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge. When a TAUBTTINm start edge is detected, TAUBnCNTm start edge to count up.
During operation	The TAUBnCDRm, TAUBnCNTm, and TAUBnCSRm registers can be read at any time. The TAUBnCSC.TAUBnCLOV bit can be set to 1.	TAUBnCNTm starts to count up from 0000_H. When a TAUBTTINm valid edge is detected: • TAUBnCNTm transfers (captures) its value to TAUBnCDRm, and retains its value • INTTAUBnIm is then generated. • The count stops at the value transferred to TAUBnCDRm + 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TEm is cleared to 0 and the counter stops. TAUBnCNTm stops and both it and TAUBnCSRm.TAUBnOVF retain their current values.

Restart operation

29.12.7.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUBnCMORm.TAUBnCOS[1:0] = 00_B

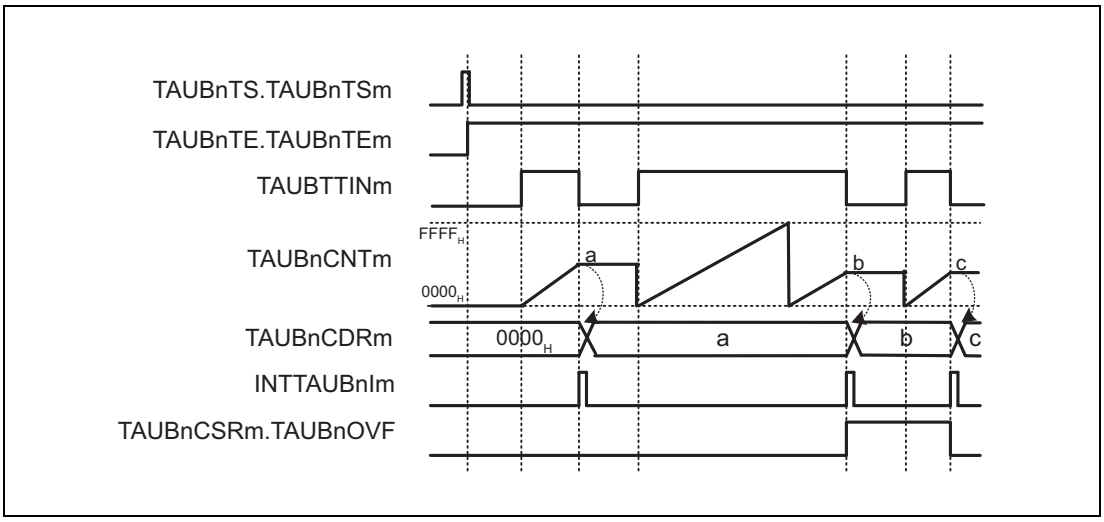


Figure 29.57 TAUBnCMORm.TAUBnCOS[1:0] = 00_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is written to TAUBnCDRm and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, while any overflow has not occurred, TAUBnCSRm.TAUBnOVF is cleared to 0.

(2) TAUBnCMORm.TAUBnCOS[1:0] = 01_B

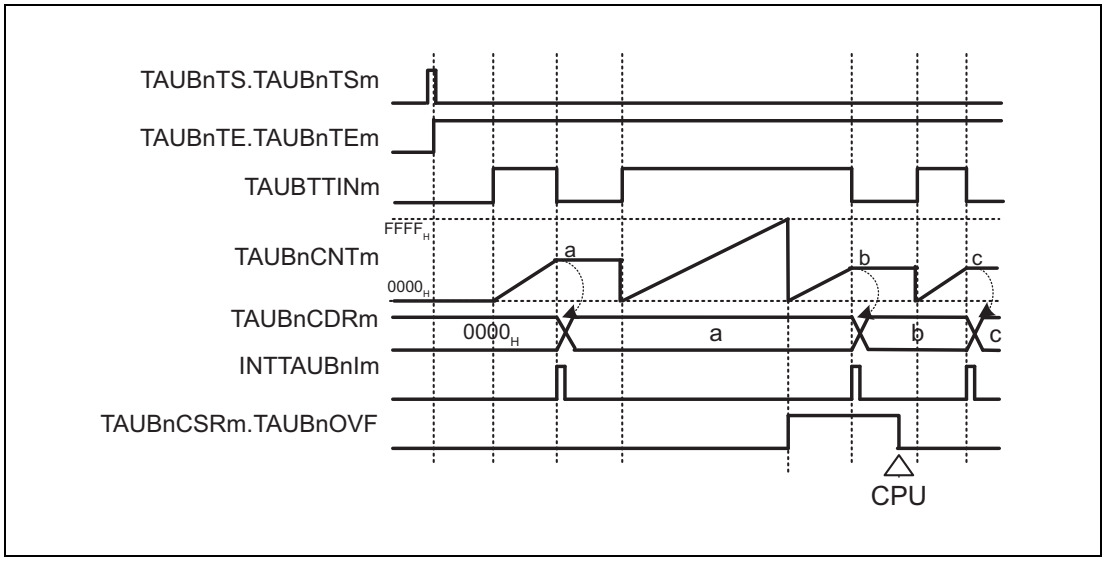


Figure 29.58 TAUBnCMORm.TAUBnCOS[1:0] = 01_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUBnCDRm remains unchanged and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, the value of TAUBnCNTm is written to TAUBnCDRm.
- TAUBnCSRm.TAUBnOVF is only cleared by a CPU command (The TAUBnCSCm.TAUBnCLOV bit = 1).

(3) TAUBnCMORm.TAUBnCOS[1:0] = 10_B

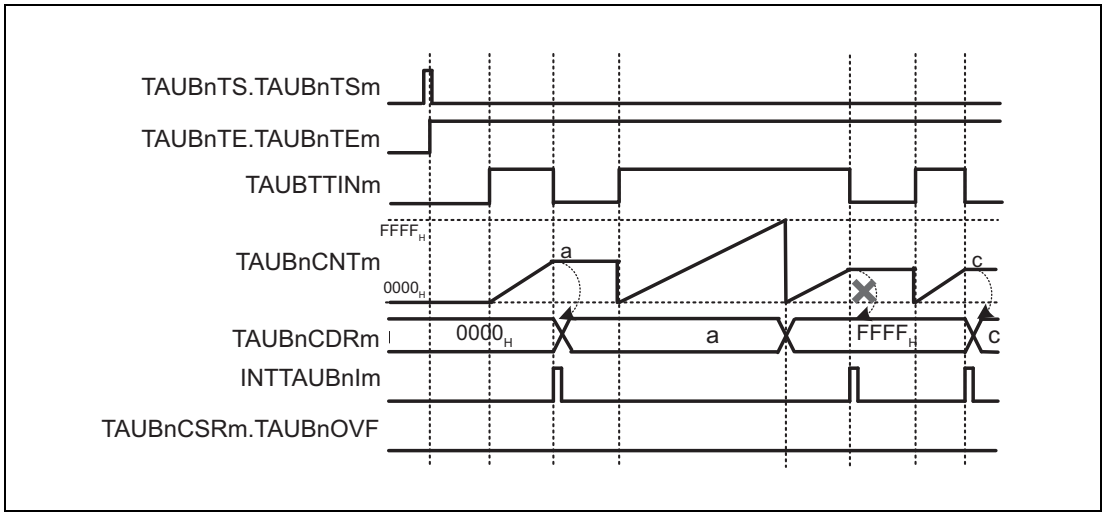


Figure 29.59 TAUBnCMORm.TAUBnCOS[1:0] = 10_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- When an overflow occurs, TAUBnCDRm is set to $FFFF_H$ and TAUBnCSRm.TAUBnOVF remains 0.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm stops counting, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.

(4) TAUBnCMORm.TAUBnCOS[1:0] = 11_B

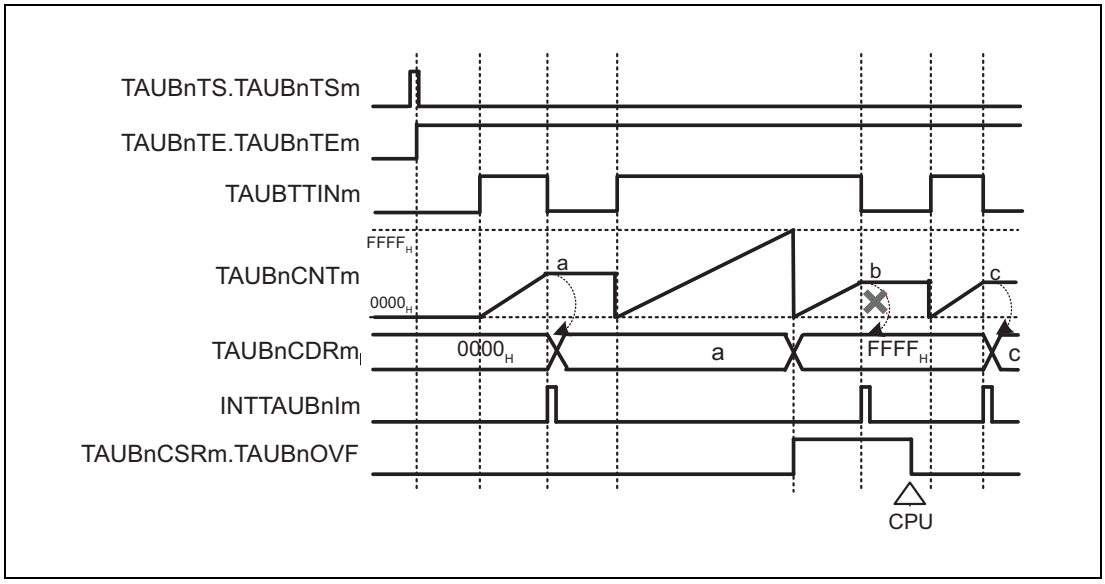


Figure 29.60 TAUBnCMORm.TAUBnCOS[1:0] = 11_B, TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 11_B

- When an overflow occurs, TAUBnCDRm is set to $FFFF_H$, and TAUBnCSRm.TAUBnOVF is set to 1.
- Upon detection of the next valid TAUBTTINm input edge, TAUBnCNTm stops counting, but TAUBnCDRm and TAUBnCSRm.TAUBnOVF remain unchanged.
- Thus, the next TAUBTTINm input valid edge after the overflow is ignored.
- TAUBnCSRm.TAUBnOVF is cleared by setting TAUBnCSCm.TAUBnCLOV = 1.

(5) When an overflow occurs (high width measurement)

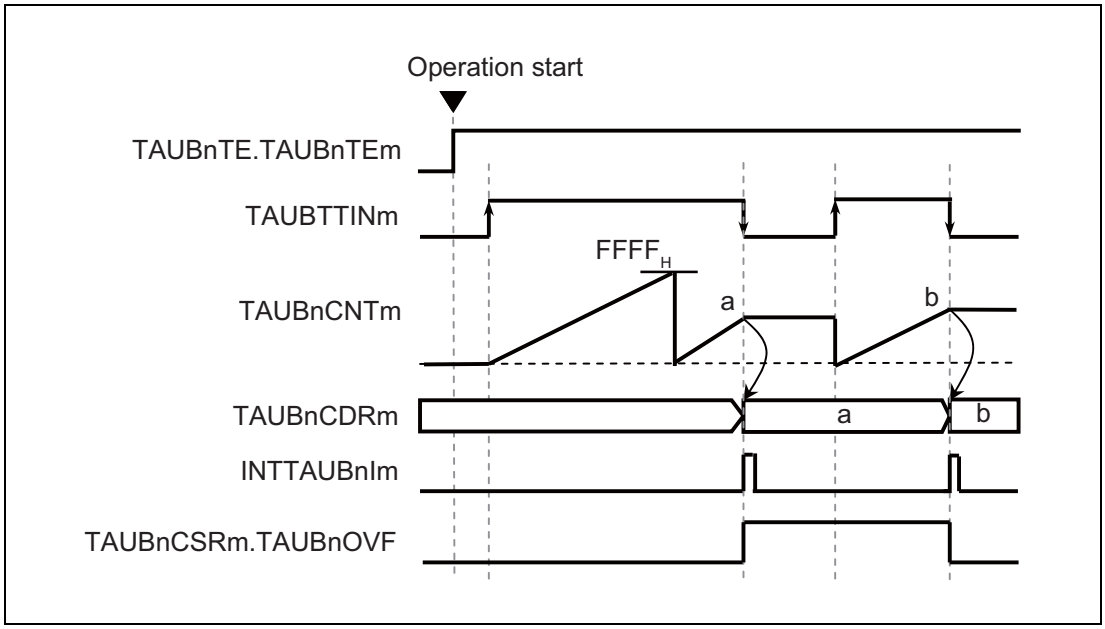


Figure 29.61 When an Overflow Occurs

When a capture trigger is input after the counter value has overflowed, the counter value is transferred to TAUBnCDRm and at the same time TAUBnCSRm.TAUBnOVF is set to 1.

TAUBnCSRm.TAUBnOVF is kept at 1 until the next capture trigger occurs.

If the next capture trigger is not accompanied by an overflow, TAUBnCSRm.TAUBnOVF is cleared to 0.

TAUBTTINm input signal width (example when TAUBnCSRm.TAUBnOVF is 1 and TAUBnCDRm is a)
= count clock cycle × ((10000_H × TAUBnCSRm.TAUBnOVF) + (TAUBnCDRm capture value + 1))
= count clock cycle × ((10000_H × 1) + (a+1))
= count clock cycle × (10000_H + a+1)

29.12.8 TAUBTTINm Input Position Detection Function

29.12.8.1 Overview

Summary

This function measures the interval of an input signal by capturing the counter value on a valid edge of the TAUBTTINm signal.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1. The counter starts to count from 0000_H. When a valid TAUBTTINm input stop edge is detected, the current TAUBnCNTm value is written to TAUBnCDRm and an interrupt (INTTAUBnIm) is generated. The counter continues to count from the current value until the next valid TAUBTTINm input edge is detected.

When the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTE

The TAUBTTINm input signal is sampled at the frequency of the operation clock, specified by the TAUBnCMORm.TAUBnCKS[1:0] bits. As a result, the output cycle of TAUBTTOUTm has an error of ± 1 operation clock cycle.

Conditions

If the TAUBnCMORm.MD0 bit is set to 0, the first interrupt after a start or restart is not generated.

29.12.8.2 Equations

Function duration at a TAUBTTINm input pulse =

count clock cycle × (TAUBnCDRm capture value + 1)

29.12.8.3 Block Diagram and General Timing Diagram

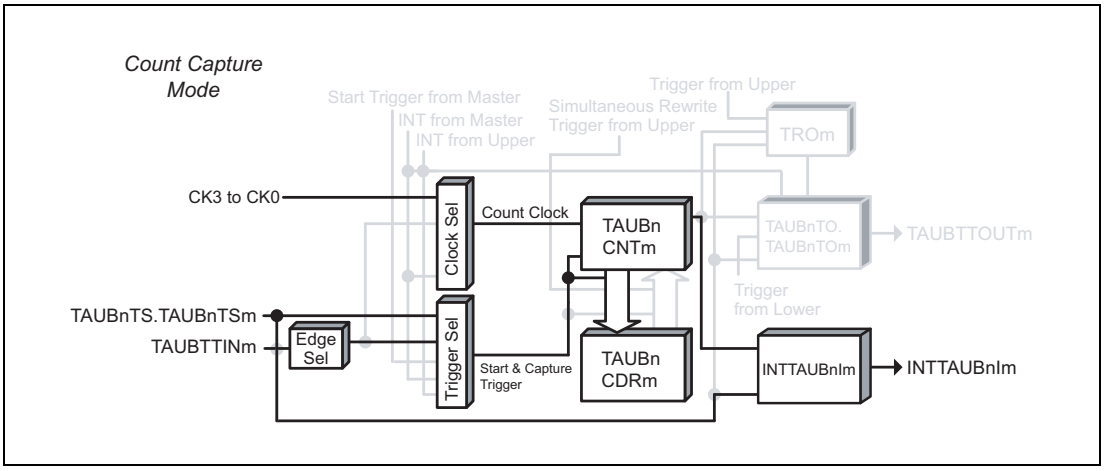


Figure 29.62 Block Diagram for TAUBTTINm Input Position Detection Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is not generated at operation start (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

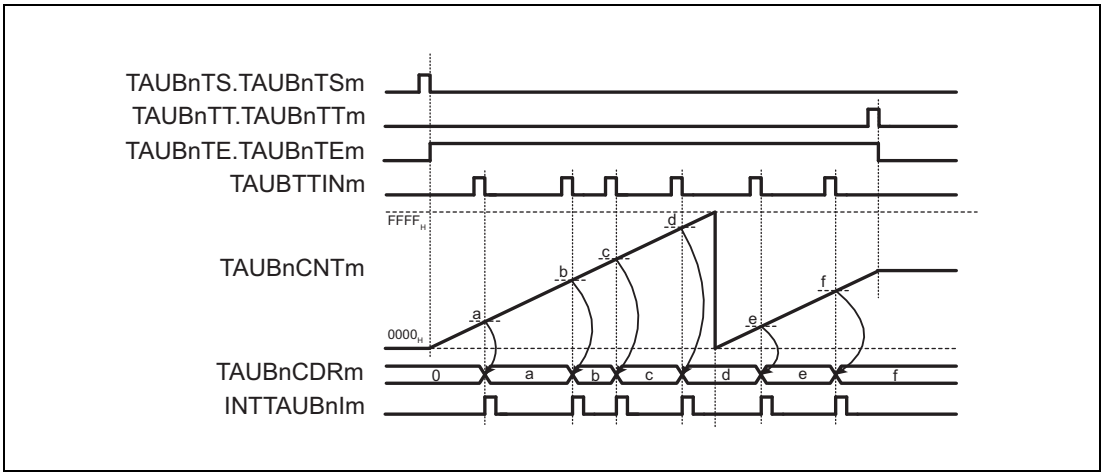


Figure 29.63 General Timing Diagram for TAUBTTINm Input Position Detection Function

29.12.8.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.73 Contents of the TAUBnCMORm Register for TAUBTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1011_B.
0	TAUBnMD0	0: INTTAUBnIm not generated at operation start 1: Generates INTTAUBnIm at operation start

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.74 Contents of the TAUBnCMURm Register for TAUBTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the TAUBTTINm Input Position Detection Function. Therefore, these registers must be set to 0.

Table 29.75 Simultaneous Rewrite Settings for TAUBTTINm Input Position Detection Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.8.5 Operating Procedure for TAUBTTINm Input Position Detection Function

Table 29.76 Operating Procedure for TAUBTTINm Input Position Detection Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in Table 29.73, Contents of the TAUBnCMORm Register for TAUBTTINm Input Position Detection Function and Table 29.74, Contents of the TAUBnCMURm Register for TAUBTTINm Input Position Detection Function The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. INTTAUBnIm is generated when TAUBnCMORm.TAUBnMD0 is set to 1.
During operation	The values of TAUBnCMURm.TAUBnTIS[1:0] bits can be changed at any time. The TAUBnCDRm and TAUBnCSRm registers can be read at any time.	TAUBnCNTm starts to count up from 0000_H. When a TAUBTTINm valid edge is detected: • TAUBnCNTm transfers (captures) its value to TAUBnCDRm • INTTAUBnIm is output. • The counter value is not cleared to 0000_H and TAUBnCNTm continues count operation. Afterwards, this procedure is repeated. If TAUBnCNTm reaches $FFFF_H$, the counter restarts from 0000_H.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

29.12.8.6 Specific Timing Diagrams

(1) Operation stop and restart

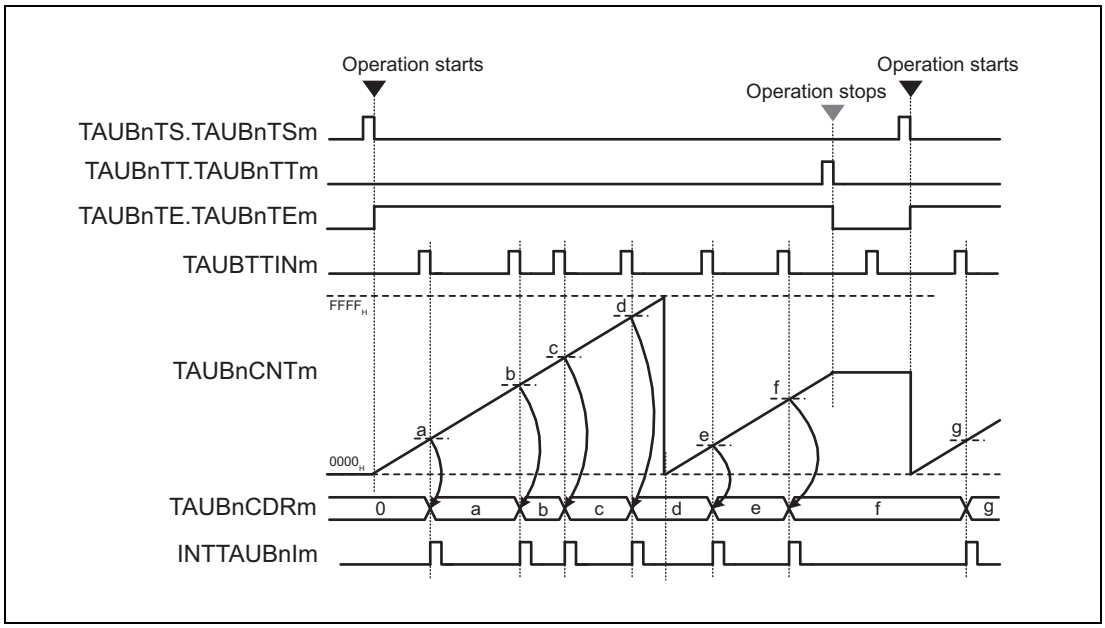


Figure 29.64 Operation Stop and Restart, (TAUBnCMORm.TAUBnMD0 = 0, TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TEm to 0.
- TAUBnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUBTTINm input edges are ignored.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. TAUBnCNTm restarts to count from 0000_H.

29.12.9 TAUBTTINm Input Period Count Detection Function

29.12.9.1 Overview

Summary

This function measures the cumulative width of a TAUBTTINm input signal.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.

This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The counter awaits a valid TAUBTTINm input edge.

When a valid TAUBTTINm input start edge is detected, the counter starts to count from 0000_H.

When a valid TAUBTTINm input stop edge is detected, the current TAUBnCNTm value is written to TAUBnCDRm and an interrupt (INTTAUBnIm) is generated. The counter stops and retains its value (TAUBnCDRm + 1) until the next valid TAUBTTINm input start edge is detected.

When a next valid TAUBTTINm input start edge is detected, the counter restarts from the value retained while stopping.

If the counter reaches $FFFF_H$, the counter restarts from 0000_H.

NOTES

1. The TAUBTTINm input signal is sampled at the frequency of the operation clock, specified by the TAUBnCMORm.TAUBnCKS[1:0] bits.
2. As this function is to measure the TAUBTTINm input signal width, setting TAUBnTS.TAUBnTSm to 1 is disabled while TAUBnTE.TAUBnTEm = 1.

Conditions

The valid start and stop edges are specified by the TAUBnCMURm.TIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, the TAUBTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUBnCMURm.TAUBnTIS[1:0] = 11_B, the TAUBTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

29.12.9.2 Equations

Cumulative TAUBTTINm input width =

count clock cycle × (TAUBnCDRm capture value + 1)

29.12.9.3 Block Diagram and General Timing Diagram

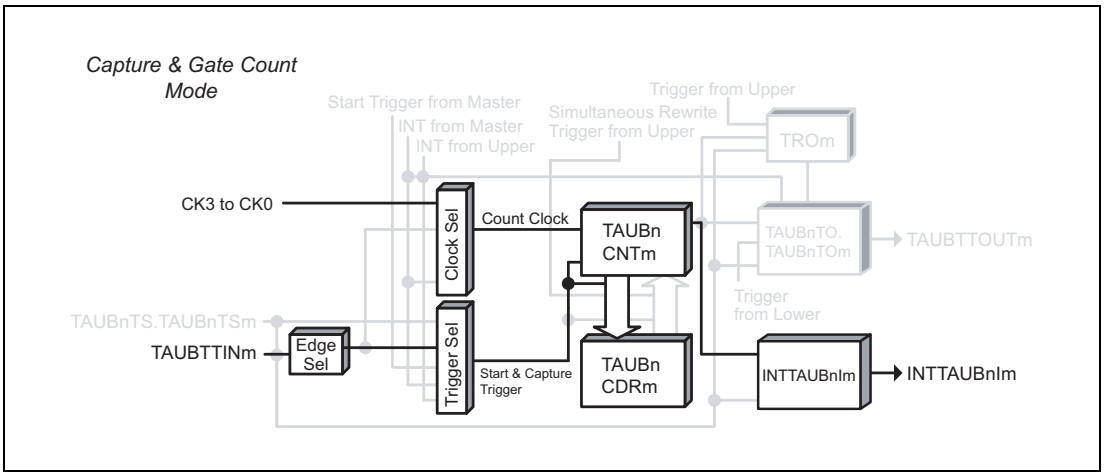


Figure 29.65 Block Diagram for TAUBTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

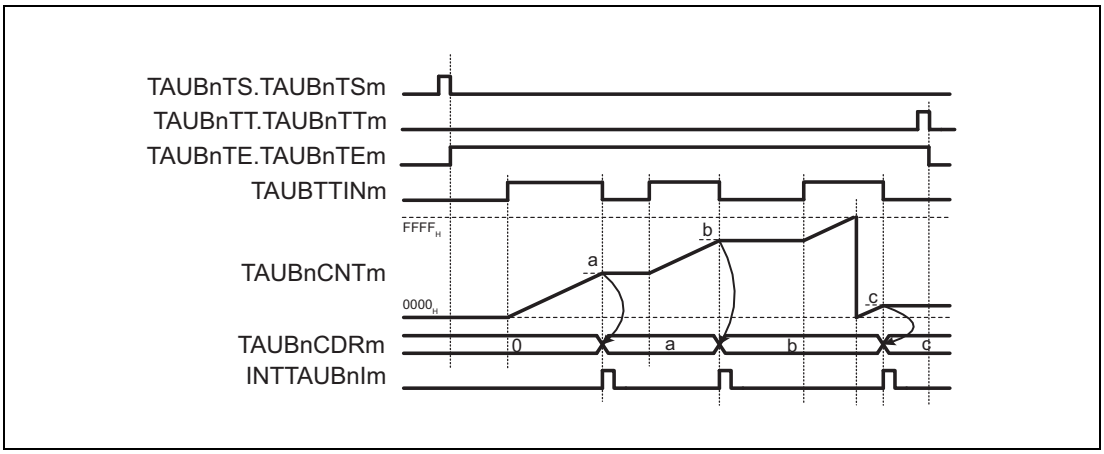


Figure 29.66 General Timing Diagram for TAUBTTINm Input Period Count Detection Function

29.12.9.4 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.77 Contents of the TAUBnCMORm Register for TAUBTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1101_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.78 Contents of the TAUBnCMURm Register for the TAUBTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the TAUBTTINm Input Period Count Detection Function. Therefore, these registers must be set to 0.

Table 29.79 Simultaneous Rewrite Settings for TAUBTTINm Input Period Count Detection Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.9.5 Operating Procedure for TAUBTTINm Input Period Count Detection Function

Table 29.80 Operating Procedure for TAUBTTINm Input Period Count Detection Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 29.77, Contents of the TAUBnCMORm Register for TAUBTTINm Input Period Count Detection Function** and **Table 29.78, Contents of the TAUBnCMURm Register for the TAUBTTINm Input Period Count Detection Function** The TAUBnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge.
During operation	Detection of TAUBTTINm edges. The TAUBnCDRm, TAUBnCNTm, and TAUBnCSRm registers can be read at any time.	When a TAUBTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUBnCNTm starts to count up from the stop value. When TAUBnCNTm detects a capture edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUBnCDRm and INTTAUBnIm is generated. Counting stops at the “value transferred to TAUBnCDRm + 1” value and TAUBnCNTm waits for detection of the TAUBTTINm start edge. When TAUBnCNTm reaches $FFFF_H$, the counter restarts from 0000_H. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

29.12.9.6 Specific Timing Diagrams

(1) Operation stop and restart

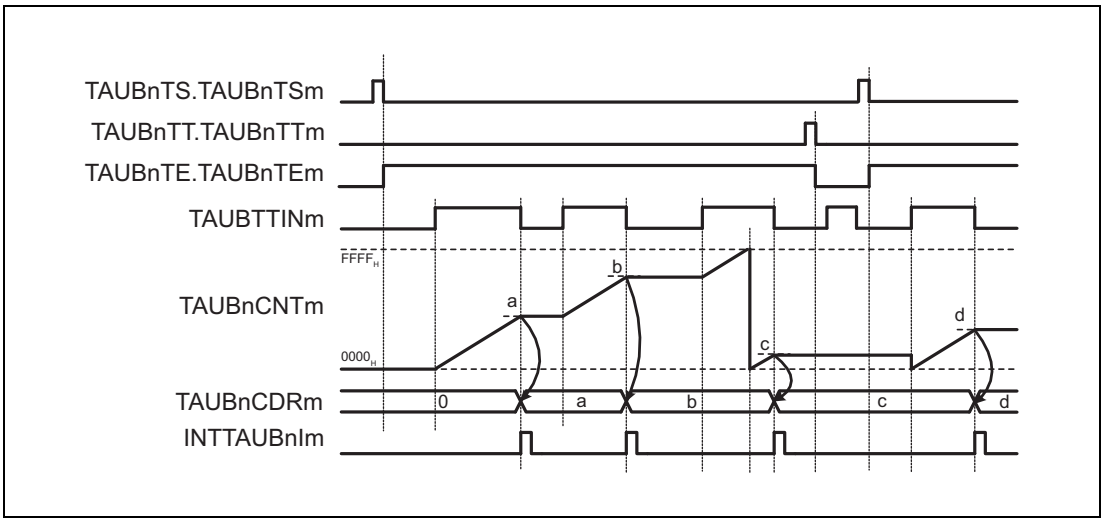


Figure 29.67 Operation Stop and Restart, (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUBTTINm input edges are ignored.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm to 1. TAUBnCNTm restarts to count from 0000_H.

29.12.10 TAUBTTINm Input Pulse Interval Judgment Function

29.12.10.1 Overview

Summary

This function outputs the result of a comparison between the count value (TAUBnCNTm) and the value in the channel data register (TAUBnCDRm) when a TAUBTTINm input pulse occurs. An interrupt request signal INTTAUBnIm is generated if the result of the comparison is true.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is started by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1.
This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value.

When a TAUBTTINm valid edge is detected or TAUBnTS.TAUBnTSm is set to 1, the function compares the current values of TAUBnCNTm and TAUBnCDRm. An interrupt request signal INTTAUBnIm is generated if the result of the comparison is true. TAUBnCNTm reloads the value of TAUBnCDRm and subsequently continues operation, regardless of the result of the comparison.

If the counter reaches 0000_H before a TAUBTTINm valid edge is detected, TAUBnCNTm overflows and is set to $FFFF_H$. It then continues to count down.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the function starts to count down.

Conditions

The TAUBnCMORm.TAUBnMD0 bit specifies the type of comparison:

- If TAUBnCMORm.TAUBnMD0 = 0, INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm.
- If TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm.

29.12.10.2 Block Diagram and General Timing Diagram

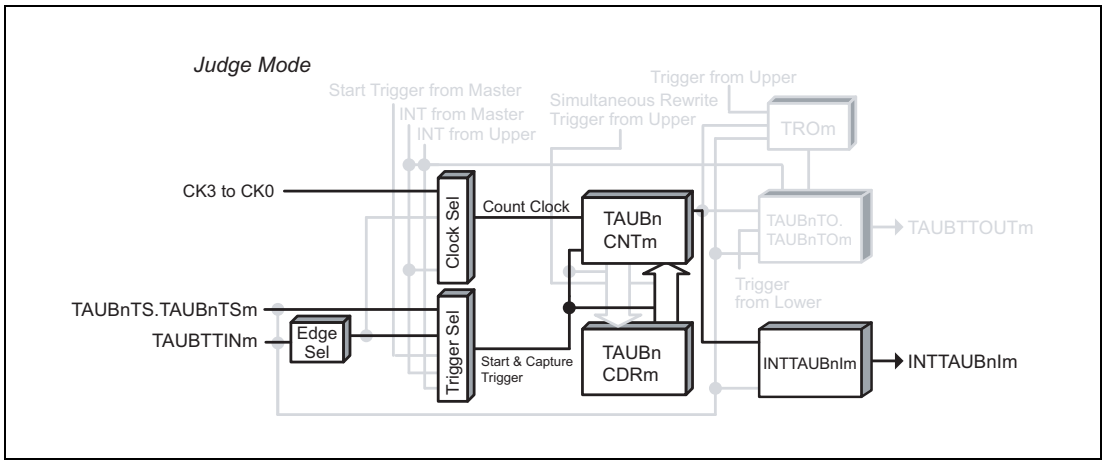


Figure 29.68 Block Diagram for TAUBTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram.

- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

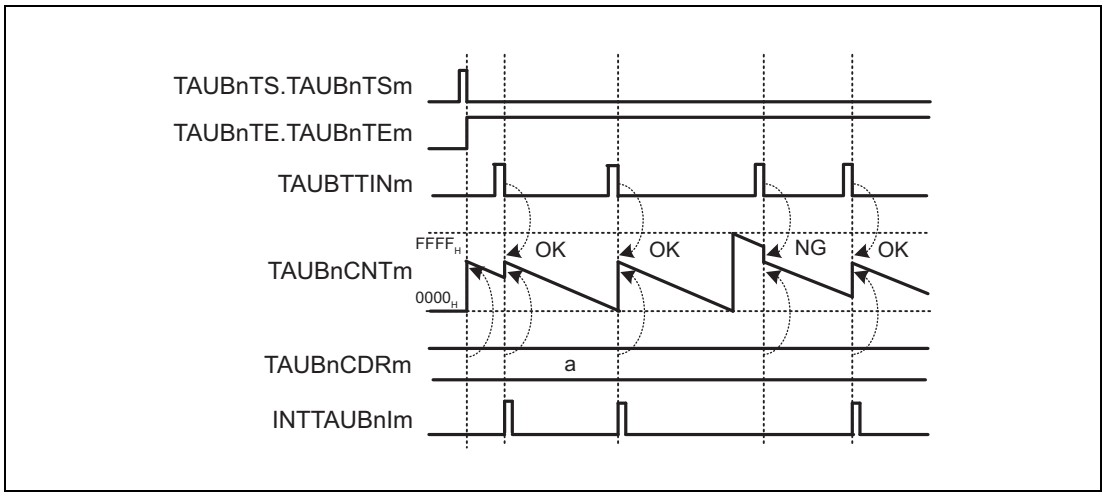


Figure 29.69 General Timing Diagram for TAUBTTINm Input Pulse Interval Judgment Function

29.12.10.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.81 Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0001_B.
0	TAUBnMD0	0: INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm 1: INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.82 Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Pulse Interval Judgment Function. Therefore, these registers must be set to 0.

Table 29.83 Simultaneous Rewrite Settings for TAUBTTINm Input Pulse Interval Judgment Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), RDM.RDMm set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.10.4 Operating Procedure for TAUBTTINm Input Pulse Interval Judgment Function

Table 29.84 Operating Procedure for TAUBTTINm Input Pulse Interval Judgment Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 29.81, Contents of the TAUBnCMORm Register for TAUBTTINm Input Pulse Interval Judgment Function** and **Table 29.82, Contents of the TAUBnCMURm Register for TAUBTTINm Input Pulse Interval Judgment Function** Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value.
During operation	The following register can be changed at any time: • TAUBnCDRm register	When TAUBnCMORm.TAUBnMD0 = 0 If TAUBnCNTm ≤ TAUBnCDRm when a TAUBTTINm input edge is detected, INTTAUBnIm is generated. When TAUBnCMORm.TAUBnMD0 = 1 If TAUBnCNTm > TAUBnCDRm when a TAUBTTINm input edge is detected, INTTAUBnIm is generated. If a TAUBTTINm input edge is detected, then TAUBnCNTm starts to count down from the value of TAUBnCDRm. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

29.12.11 TAUBTTINm Input Signal Width Judgment Function

29.12.11.1 Overview

Summary

This function compares the count value (TAUBnCNTm) for the high or low level width of a TAUBTTINm input signal and the TAUBnCDRm value, and outputs the judgment result from the interrupt request signal INTTAUBnIm.

Prerequisites

TAUBTTOUTm is not used for this function

Description

The counter is started by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. When a valid TAUBTTINm input start edge is detected, the current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value.

When a TAUBTTINm valid stop edge is detected, the function compares the current values of TAUBnCNTm and TAUBnCDRm. An interrupt request signal INTTAUBnIm is generated if the result of the comparison is true. The counter TAUBnCNTm retains its value until the next TAUBTTINm valid start edge is detected, regardless of the result of the comparison.

If the counter reaches 0000_H before a valid TAUBTTINm stop edge is detected, TAUBnCNTm overflows and is set to $FFFF_H$. It then continues to count down.

The value of TAUBnCDRm can be rewritten at any time, and the changed value of TAUBnCDRm is applied the next time the function starts to count down.

Conditions

- The TAUBnCMORm.TAUBnMD0 bit specifies the type of comparison:
  - If TAUBnCMORm.TAUBnMD0 = 0, INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm.
  - If TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm.
- The TAUBnCMURm.TAUBnTIS[1:0] bits specify the type of width measurement:
  - For high width measurement, (When TAUBnCMURm.TAUBnTIS[1:0] = 11_B) the start edge is a rising TAUBTTINm edge and the stop edge is a falling TAUBTTINm edge.
  - For low width measurement, (TAUBnCMURm.TAUBnTIS[1:0] = 10_B) the start edge is a falling TAUBTTINm edge and the stop edge is a rising TAUBTTINm edge.
- Forced restart is not possible for this function.

29.12.11.2 Block Diagram and General Timing Diagram

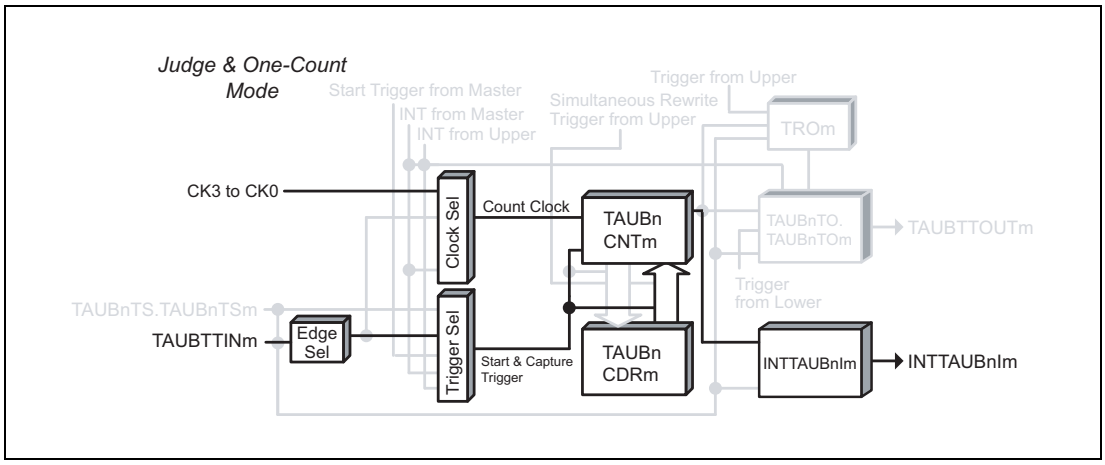


Figure 29.70 Block Diagram for TAUBTTINm Input Signal Width Judgment Function

The following settings apply to the general timing diagram.

- INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm (TAUBnCMORm.TAUBnMD0 = 0)
- TAUBTTINm valid start edge = rising edge, TAUBTTINm valid stop edge = falling edge (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

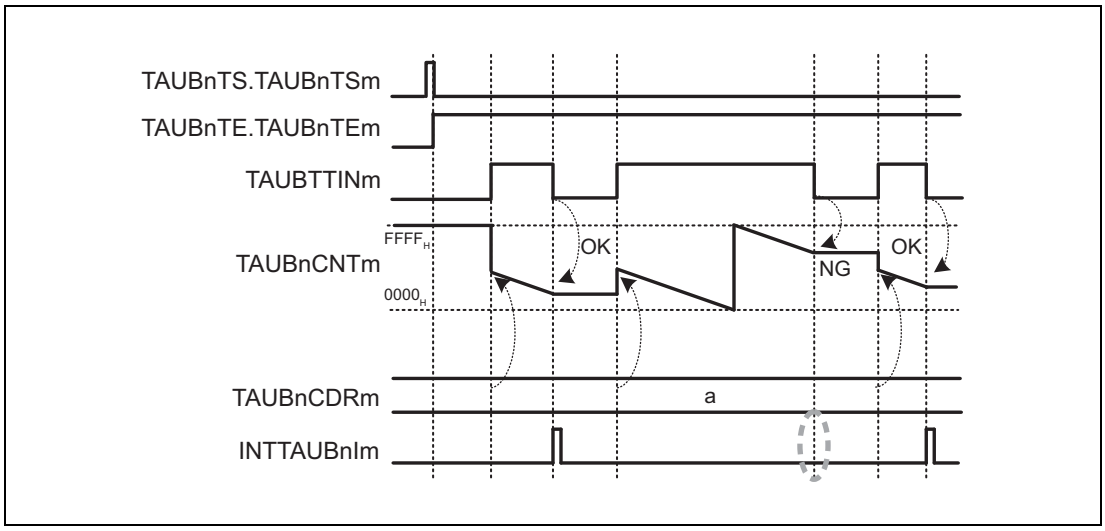


Figure 29.71 General Timing Diagram for TAUBTTINm Input Signal Width Judgment Function

29.12.11.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.85 Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0111_B.
0	TAUBnMD0	0: INTTAUBnIm is generated when TAUBnCNTm ≤ TAUBnCDRm 1: INTTAUBnIm is generated when TAUBnCNTm > TAUBnCDRm

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.86 Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Judgment Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the TAUBTTINm Input Signal Width Judgment Function. Therefore, these registers must be set to 0.

Table 29.87 Simultaneous Rewrite Settings for TAUBTTINm Input Signal Width Judgment Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), RDM.RDMm set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.11.4 Operating Procedure for TAUBTTINm Input Signal Width Judgment Function

Table 29.88 Operating Procedure for TAUBTTINm Input Signal Width Judgment Function

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 29.85, Contents of the TAUBnCMORm Register for TAUBTTINm Input Signal Width Judgment Function** and **Table 29.86, Contents of the TAUBnCMURm Register for TAUBTTINm Input Signal Width Judgment Function** Set the value of the TAUBnCDRm register	Channel operation is stopped.

Table 29.88 Operating Procedure for TAUBTTINm Input Signal Width Judgment Function

	Operation	Status of TAUBn
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the TAUBTTINm start edge.
During operation	The following register can be changed at any time: • TAUBnCDRm register	If a TAUBTTINm start edge is detected, then TAUBnCNTm starts to count down from the value of TAUBnCDRm. When TAUBnCMORm.TAUBnMD0 = 0 If TAUBnCNTm ≤ TAUBnCDRm when a TAUBTTINm input stop edge is detected, INTTAUBnIm is generated. When TAUBnCMORm.TAUBnMD0 = 1 If TAUBnCNTm > TAUBnCDRm when a TAUBTTINm input stop edge is detected, INTTAUBnIm is generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

29.12.12 Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

29.12.12.1 Overview

Summary

This function measures the width of an individual TAUBTTINm input signal. An interrupt is generated if the TAUBTTINm input width is longer than $FFFF_H + 1$.

Prerequisites

- TAUBTTOUTm is not used for this function
- The value of TAUBnCDRm must be set to $FFFF_H$.

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The counter starts when a valid TAUBTTINm input start edge is detected. $FFFF_H$ is written to TAUBnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUBTTINm input start edge is detected, TAUBnCNTm loads $FFFF_H$ and starts to count down.

If the counter reaches 0000_H before a stop edge is detected, an interrupt is generated.

Conditions

The valid start and stop edges are specified by the TAUBnCMURm.TAUBnTIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, the TAUBTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUBnCMURm.TAUBnTIS[1:0] = 11_B, the TAUBTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

29.12.12.2 Block Diagram and General Timing Diagram

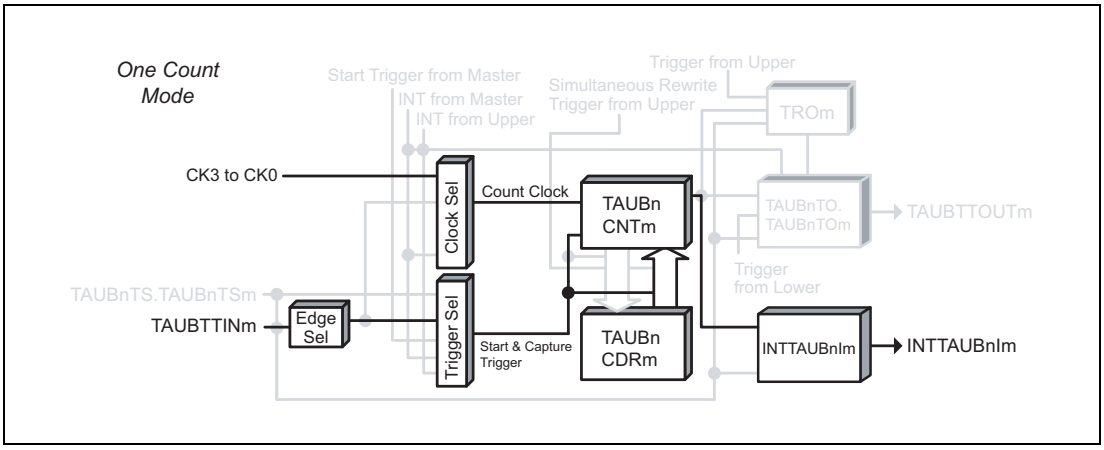


Figure 29.72 Block Diagram for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

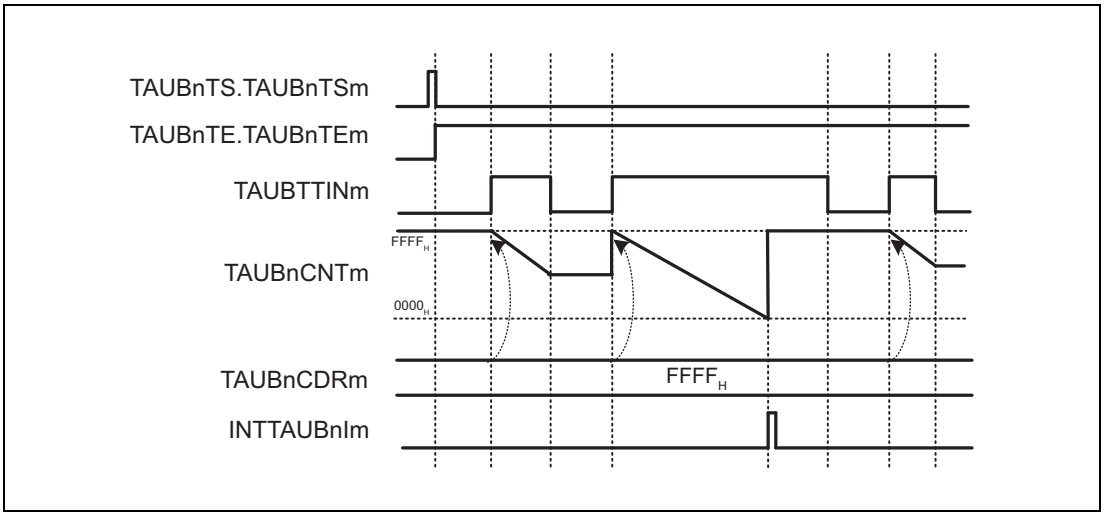


Figure 29.73 General Timing Diagram for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

29.12.12.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.89 Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	Write 0_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.90 Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, and TAUBnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUBTTINm Width Measurement). Therefore, these registers must be set to 0.

Table 29.91 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.12.4 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

Table 29.92 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in **Table 29.89, Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)** and **Table 29.90, Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Width Measurement)** Set the value of the TAUBnCDRm register to $FFFF_H$.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUBTTINm start edge	TAUBnTE.TAUBnTEm is set to 1 and TAUBnCNTm waits for detection of the start edge. When a start edge is detected, TAUBnCNTm loads the TAUBnCDRm value ($FFFF_H$).
During operation	The TAUBnCNTm register can be read at any time.	TAUBnCNTm counts down. When the counter reaches 0000_H: • INTTAUBnIm is generated When a TAUBTTINm input stop edge is detected during count operation: • TAUBnCNTm stops and retains its current value. When a TAUBTTINm input start edge is detected while the counter is stopped: • The TAUBnCDRm value ($FFFF_H$) is loaded to TAUBnCNTm again and the counter starts to count down. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

29.12.13 Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

29.12.13.1 Overview

Summary

This function measures the cumulative width of a TAUBTTINm input signal. An interrupt is generated if the cumulative TAUBTTINm input width is longer than $FFFF_H$.

Prerequisites

- TAUBTTOUTm is not used for this function
- The value of TAUBnCDRm must be set to $FFFF_H$

Description

The counter is enabled by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

The counter starts when a valid TAUBTTINm input start edge is detected. $FFFF_H$ is written to TAUBnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUBTTINm input start edge and then continues to count down from the current value.

When the counter reaches 0000_H an interrupt is generated. $FFFF_H$ is written to TAUBnCNTm and the counter continues to count down until a TAUBTTINm input stop edge is detected.

Conditions

The valid start and stop edges are specified by the TAUBnCMURm.TAUBnTIS[1:0] bits.

- If TAUBnCMURm.TAUBnTIS[1:0] = 10_B, the TAUBTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUBnCMURm.TAUBnTIS[1:0] = 11_B, the TAUBTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

29.12.13.2 Block Diagram and General Timing Diagram

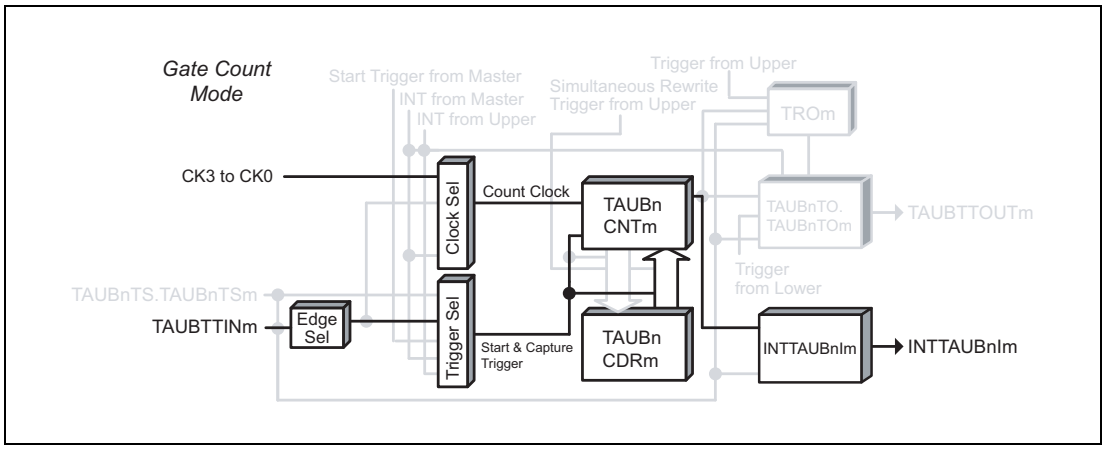


Figure 29.74 Block Diagram for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUBnCMURm.TAUBnTIS[1:0] = 11_B)

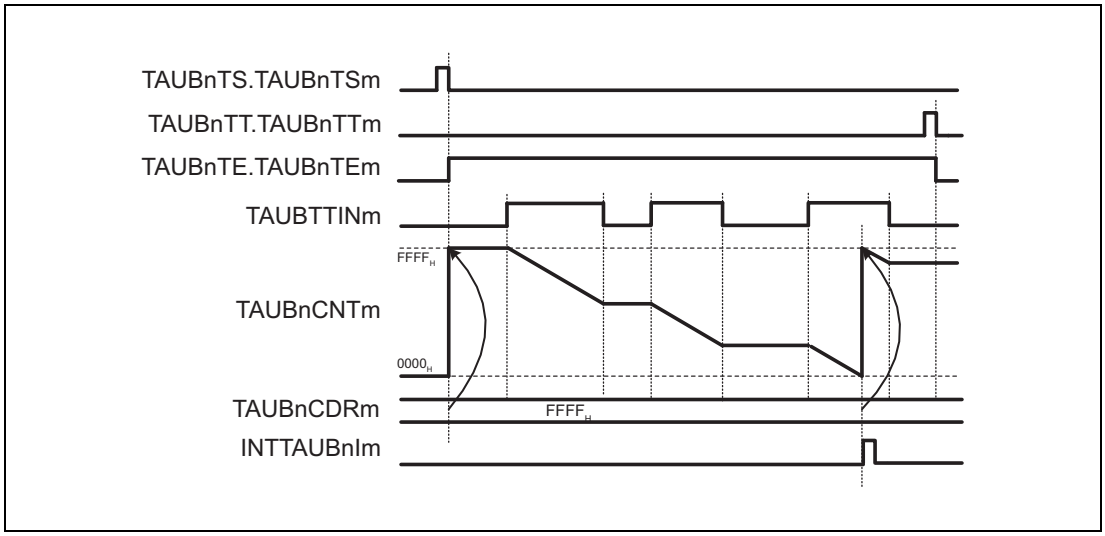


Figure 29.75 General Timing Diagram for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

29.12.13.3 Register Settings

(1) TAUBnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.93 Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 010_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1100_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.94 Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUBnRDE, TAUBnRDS, TAUBnRDM, TAUBnRDC) cannot be used with the Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 29.95 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	0: Disables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: When simultaneous rewrite is disabled (TAUBnRDE.TAUBnRDEm = 0), set these bits to 0
TAUBnRDM.TAUBnRDMm	
TAUBnRDC.TAUBnRDCm	

29.12.13.4 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

Table 29.96 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection)

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers as described in Table 29.93, Contents of the TAUBnCMORm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection) and Table 29.94, Contents of the TAUBnCMURm Register for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection) Set the value of the TAUBnCDRm register to $FFFF_H$.	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1 TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUBTTINm start edge	TAUBnTE.TAUBnTEm is set to 1 TAUBnCNTm waits for detection of the start edge. When a start edge is detected, TAUBnCNTm loads the TAUBnCDRm value ($FFFF_H$).
During operation	The TAUBnCNTm register can be read at all times.	TAUBnCNTm counts down. When the counter reaches 0000_H: • INTTAUBnIm is generated • TAUBnCNTm loads the TAUBnCDRm value ($FFFF_H$) and continues count operation When a TAUBTTINm input stop edge is detected during count operation: • TAUBnCNTm stops and retains its current value. When a TAUBTTINm input start edge is detected while the counter is stopped: • TAUBnCNTm starts to count down from the stop value. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

29.13 Independent Channel Simultaneous Rewrite Functions

The following describes functions that carry out simultaneous rewrite:

29.13.1 Simultaneous Rewrite Trigger Generation Function Type 1

29.13.1.1 Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is generated at regular intervals. The upper channel is for generating the simultaneous rewrite trigger (TAUBnRDC.TAUBnRDCm = 1), and the lower channels are for conducting simultaneous rewrite when triggered from the upper channel (TAUBnRDC.TAUBnRDCm = 0).

Prerequisites

- Two (or more) channels that are lower than the channel used as the upper channel, each with simultaneous rewrite enabled (TAUBnRDE.TAUBnRDEm = 1)
- The operation mode of the upper channel must be set to interval timer mode, see **Table 29.97, Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1**
- For the operation modes that can be set to the lower channels, see **Table 29.37, Simultaneous Rewrite Methods and when They are Triggered**
- In this function, TAUBTTOUTm is not used for all the channels.

Description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) of the upper and lower channel(s) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of the data register buffer of the upper channel (TAUBnCDRm buf) is written to the counter (TAUBnCNTm) and the counter starts to count down from this value.
The counter(s) of the lower channel(s) start to count as specified by their selected operating modes.

When a counter reaches 0000_H, an interrupt is generated from the channel.
The corresponding TAUBnCNTm then reloads the current TAUBnCDRm buffer value and subsequently continues operation.

If the channel where the interrupt occurs is specified as the trigger channel for simultaneous rewrite (TAUBnRDC.TAUBnRDCm = 1) and is an upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUBnRSF.TAUBnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers. Each time a counter starts to count down, it reads the value in the data register buffer and counts down from this value.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Conditions

- The channel which is monitored for INTTAUBnIm is specified by setting TAUBnRDC.TAUBnRDCm = 1 for the corresponding channel. The TAUBnRDC.TAUBnRDCm bit must be 0 for all other channels in which simultaneous rewrite should take place.

29.13.1.2 Equations

Simultaneous rewrite trigger generation cycle = count clock cycle × (TAUBnCDRm + 1)

To control simultaneous rewrite, the following condition must be satisfied:

[For PWM]

TAUBnCDRm = [(value of TAUBnCDRm of master channel subject to simultaneous rewrite + 1) × number of interrupts] – 1

[For triangle PWM]

TAUBnCDRm = [(value of TAUBnCDRm of master channel subject to simultaneous rewrite + 1) × 2 × number of interrupts] – 1

That is, the ratio of TAUBnCDRm + 1 and value of TAUBnCDRm of master channel subject to simultaneous rewrite + 1 must be an integer. This integer corresponds to the number of interrupts.

Note that the cycle for the triangle PWM is twice the cycle for the PWM

29.13.1.3 Block Diagram and General Timing Diagram

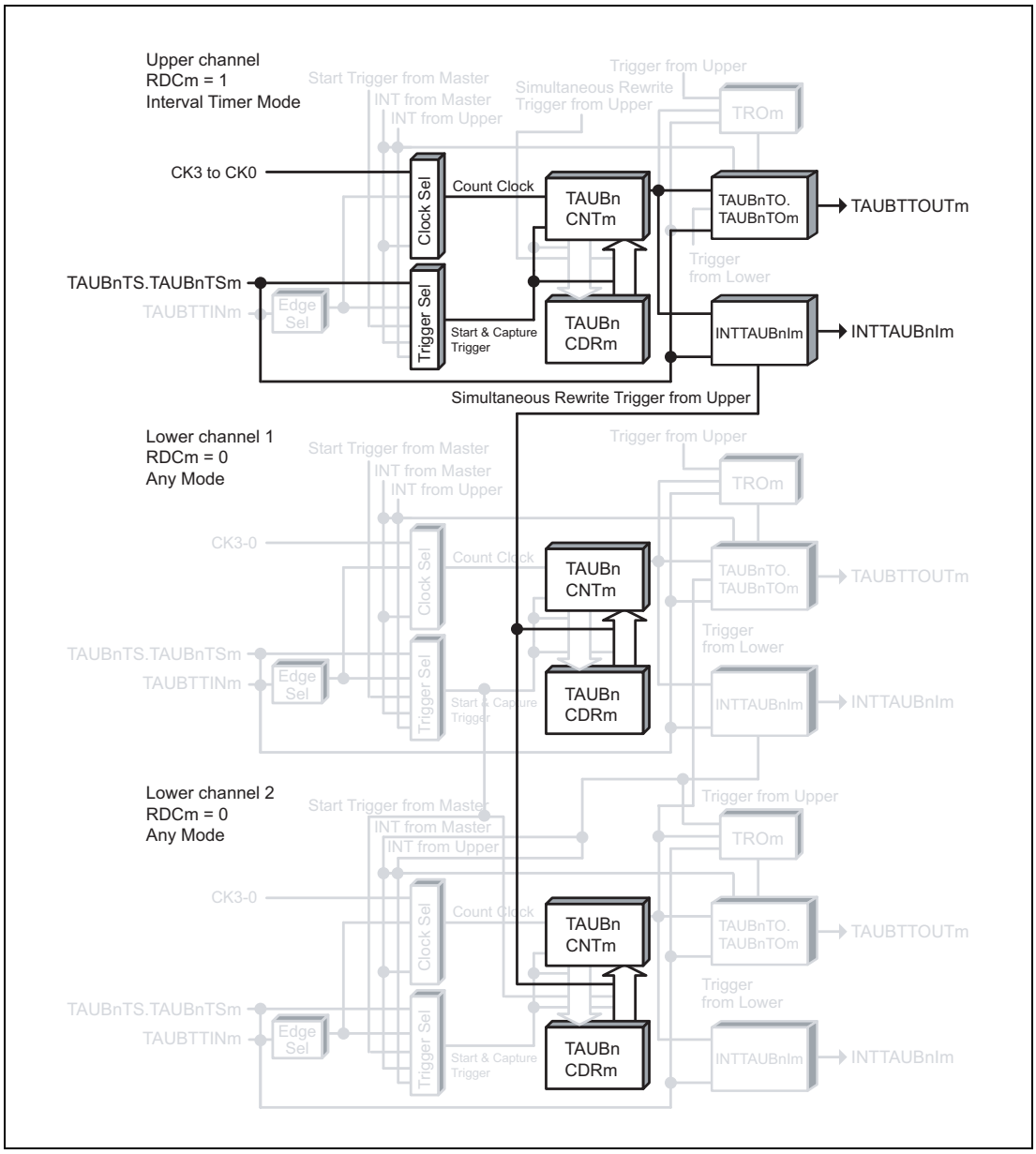


Figure 29.76 Block Diagram for Simultaneous Rewrite Trigger Generation Function Type 1

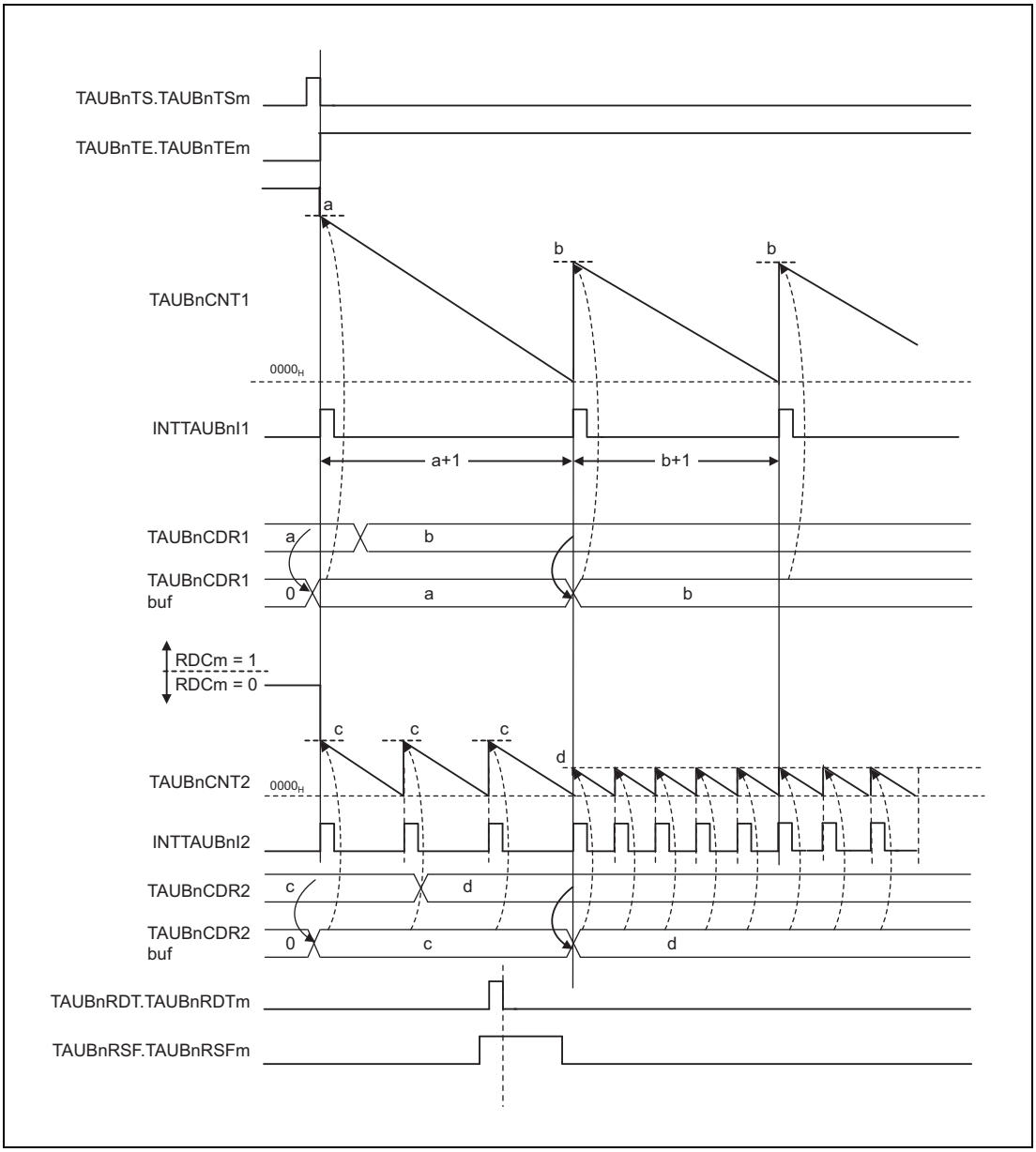


Figure 29.77 General Timing Diagram for Simultaneous Rewrite Trigger Generation Function Type 1

29.13.1.4 Register Settings for The Upper Channel

(1) TAUBnCMORm for the upper channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.97 Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for the upper channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.98 Contents of the TAUBnCMURm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the upper channel

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function. However, it can be used in independent channel output mode controlled by software.

(4) Simultaneous rewrite for the upper channel

Table 29.99 Simultaneous Rewrite Settings for Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	1: Selects an upper channel as the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The signal that controls simultaneous rewrite is loaded when the master channel starts counting
TAUBnRDC.TAUBnRDCm	1: Channel is monitored for an INTTAUBnIm signal that is used as the simultaneous rewrite trigger

29.13.1.5 Register Settings for the Lower Channel(s)

(1) Register settings for the lower channel(s)

For the TAUBnCMORm register of the lower channels, follow the TAUBnCMORm register settings for the operation mode that can be set. (See Table 29.37, Simultaneous Rewrite Methods and when They are Triggered)

(2) TAUBnCMURm for the lower channel(s)

For the TAUBnCMURm register of the lower channels, follow the TAUBnCMURm register settings for the operation mode that can be set. (See Table 29.37, Simultaneous Rewrite Methods and when They are Triggered)

(3) Channel output mode for the lower channel(s)

Output can be made according to the setting for lower channels (master/slave). As for the available function for simultaneous rewrite trigger generation function type 1, see **Table 29.37, Simultaneous Rewrite Methods and when They are Triggered**.

(4) Simultaneous rewrite for the lower channel(s)

Table 29.100 Simultaneous Rewrite Settings for the Lower Channel in Simultaneous Rewrite Trigger Generation Function Type 1

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	1: Selects an upper channel as the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The signal that controls simultaneous rewrite is loaded when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

29.13.1.6 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Table 29.101 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1

	Operation	Status of TAUBn
Initial channel setting	Set the TAUBnCMORm register and TAUBnCMURm registers for the upper channel as described in Table 29.97, Contents of the TAUBnCMORm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1 and **Table 29.98, Contents of the TAUBnCMURm Register for the Upper Channel of the Simultaneous Rewrite Trigger Generation Function Type 1** Set the TAUBnCMORm register and TAUBnCMURm registers for the lower channel as described in **Section 29.13.1.5, Register Settings for the Lower Channel(s)** Set the value of the TAUBnCDRm register	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm to 1. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is set to 1 and the counter starts. TAUBnCNTm loads the TAUBnCDRm value. When TAUBnCMORm.TAUBnMD0 = 1, INTTAUBnIm is generated.
During operation	TAUBnRDT.TAUBnRDTm, TAUBnCDR.TAUBnCDRm can be changed. TAUBnRSF.TAUBnRSFm can be read at all times.	TAUBnCNTm counts down. When the counter reaches 0000_H: • TAUBnCNTm reloads the TAUBnCDRm value and continues count operation • INTTAUBnIm is generated Simultaneous rewrite is controlled when INTTAUBnIm is generated from the channel where TAUBnRDC.TAUBnRDCm is set to 1. Afterwards, this procedure is repeated.
Stop operation	Set TAUBnTT.TAUBnTTm to 1. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm stops and retains its current value.

Restart operation

29.14 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the Timer Array Unit B. For a general overview of synchronous channel operation, see Section 29.2, Overview.

29.14.1 PWM Output Function

29.14.1.1 Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the pulse width (duration) of the TAUBTTOUTm to be set. The pulse cycle is set in the master channel. The pulse width is set in the slave channel.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode, see **Table 29.102, Contents of the TAUBnCMORm Register for the Master Channel of the PWM Output Function**
- The operation mode of the slave channel(s) must be set to one-count mode, see **Table 29.105, Contents of the TAUBnCMORm Register for the Slave Channel of the PWM Output Function**
- TAUBTTOUTm is not used for the master channel of this function
- The channel output mode of the slave channel(s) must be set to synchronous channel output mode 1.

Description

The counters are started by setting the channel trigger bits (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation. The current value of TAUBnCDRm is written to TAUBnCNTm and the counters start to count down from these values. INTTAUBnIm is generated on the master channel and TAUBTTOUTm (slave) toggles, which realizes a PWM output.

- Master channel:

  When the counter of the master channel reaches 0000_H, pulse cycle time has elapsed and INTTAUBnIm is generated. The counter loads the TAUBnCDRm value and counts down.

- Slave channel:

  The INTTAUBnIm of the master channel triggers the counter of the slave channel(s). The current value of TAUBnCDRm (slave) is written to TAUBnCNTm (slave) and the counter starts to count down from this value. The TAUBTTOUTm signal is set to the active level.

  When the counter reaches 0000_H, i.e. duty time has elapsed, INTTAUBnIm is generated and the TAUBTTOUTm signal is reset to the inactive level. The counter returns to $FFFF_H$ and awaits the next INTTAUBnIm of the master channel, and thus the start of the next pulse cycle.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel(s), which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel(s) stop but retain their values. The counters can be restarted by setting TAUBnTS.TAUBnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. Please see Section 29.6, Simultaneous Rewrite.

29.14.1.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUBnCDRm (slave) / (TAUBnCDRm (master) + 1)) × 100

– Duty cycle = 0%
  TAUBnCDRm (slave) = 0000_H
– Duty cycle = 100%
  TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1

29.14.1.3 Block Diagram and General Timing Diagram

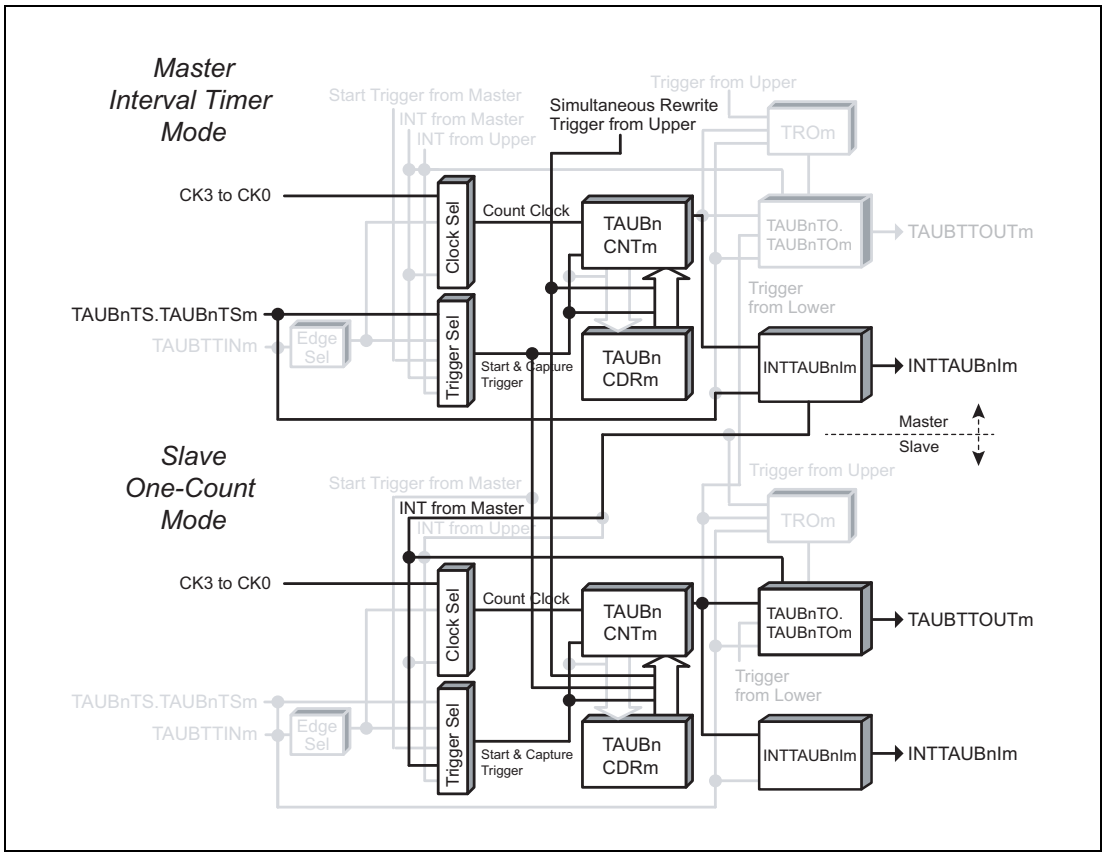


Figure 29.78 Block Diagram for PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUBnTOL.TAUBnTOLm = 0)

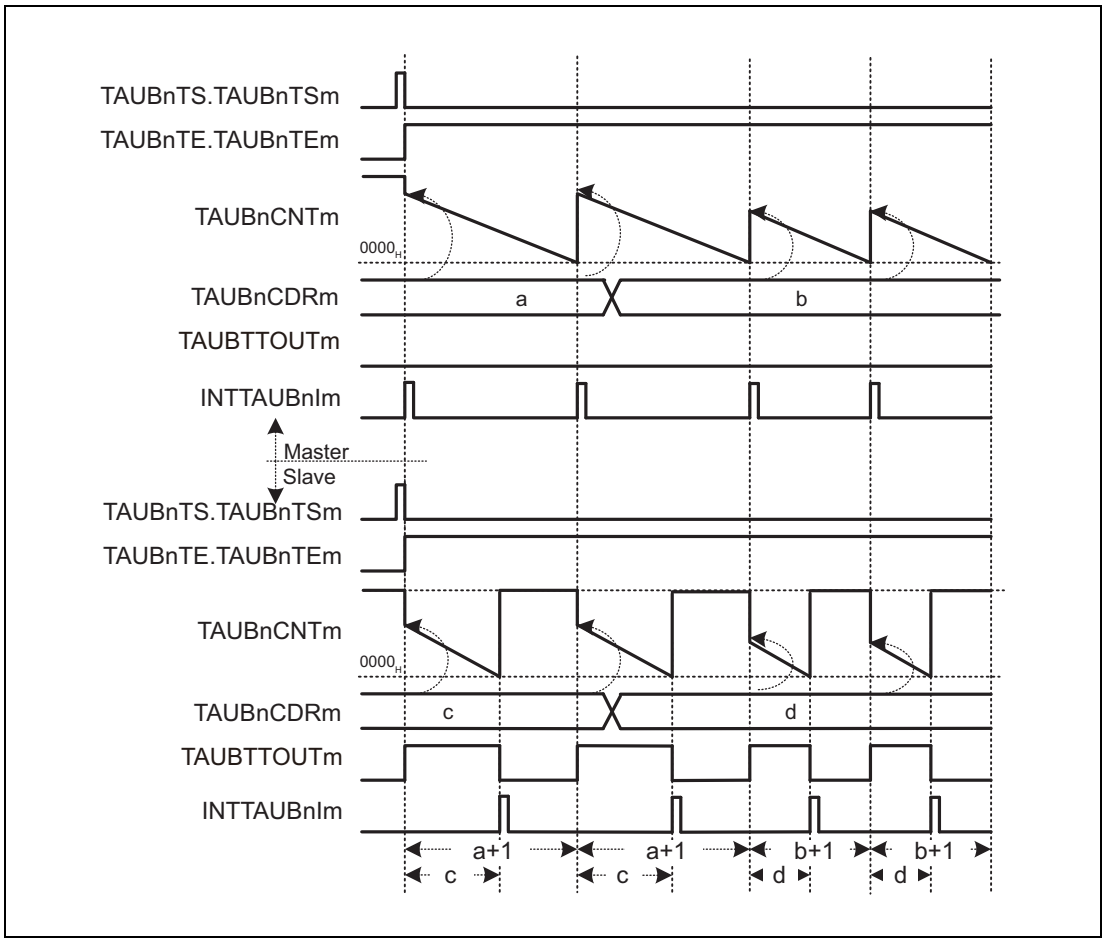


Figure 29.79 General Timing Diagram for PWM Output Function

NOTE

The interval between the slave channel starting to count and an interrupt being generated is the value of corresponding TAUBnCDRm + 1.

29.14.1.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.102 Contents of the TAUBnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.103 Contents of the TAUBnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

The channel output mode is not used by this function.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 29.104 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

NOTE

When used in TAUBnRDS.TAUBnRDSm = 1, the master channel requires an upper channel operating in Section 29.13.1, Simultaneous Rewrite Trigger Generation Function Type 1.

Configure the operation following the conditions below.

- The channel set to Simultaneous Rewrite Trigger Output Function Type 1: TAUBnRDCm = 1, TAUBnRDSm = 1
  The setting value of TAUBnCDRm to this channel is as follows.
  = ((setting value of TAUBnCDRm of the master channel subject to simultaneous rewrite + 1) × number of interrupts) – 1
- Master channel: TAUBnRDCm = 0, TAUBnRDSm = 1
- Slave channel: TAUBnRDCm = 0, TAUBnRDSm = 1

Although the value of duty exceeds 100% when the setting value of TAUBnCDRm (slave) > the setting value of TAUBnCDRm (master) + 1, the output will be aggregated to 100%.

29.14.1.5 Register Settings for the Slave Channel(s)

(1) TAUBnCMORm for the slave channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.105 Contents of the TAUBnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 100_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for the slave channel(s)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.106 Contents of the TAUBnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the slave channel(s)

Table 29.107 Control Bit Settings for Synchronous Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 29.108 Simultaneous Rewrite Settings for the Slave Channel of the PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

29.14.1.6 Operating Procedure for PWM Output Function

Table 29.109 Operating Procedure for PWM Output Function

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.1.4, Register Settings for the Master Channel**. Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.1.5, Register Settings for the Slave Channel(s)**. Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUBnIm is generated on the master channel and TAUBTTOUTm (slave) is set.
During operation	TAUBnCDRm can be changed at any time. TAUBnTOL.TAUBnTOLm can be changed. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	TAUBnCNTm of the master channel loads TAUBnCDRm and counts down. When the counter reaches 0000_H: • INTTAUBnIm (master) is generated • TAUBnCNTm (master) loads the TAUBnCDRm value and continues count operation • TAUBnCNTm (slave) loads the TAUBnCDRm value and counts down • TAUBTTOUTm (slave) is set to the active level When TAUBnCNTm (slave) reaches 0000_H: • The counter of TAUBnCNTm (slave) stops. • INTTAUBnIm (slave) is generated • TAUBTTOUTm (slave) is set to the inactive level
Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

29.14.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

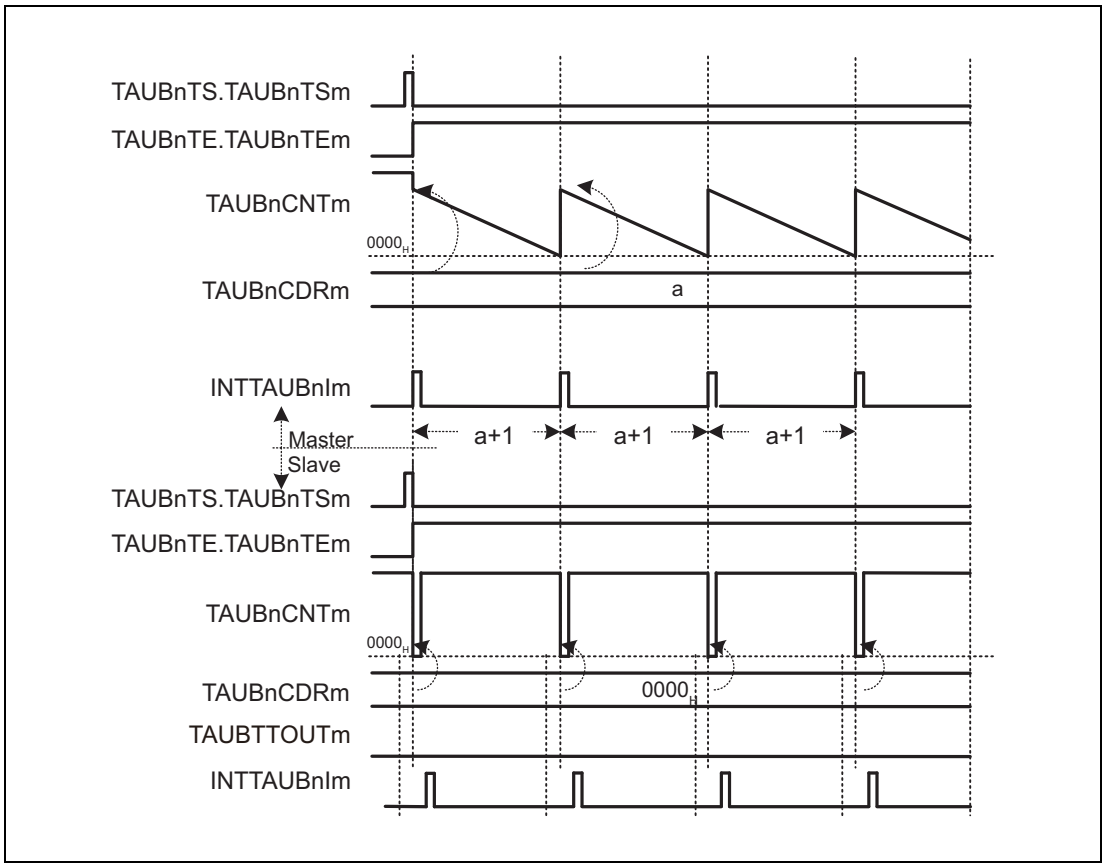


Figure 29.80 TAUBnCDRm (slave) = 0000_H, Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

- Every time the master channel generates an interrupt (INTTAUBnIm), 0000_H is written to TAUBnCNTm (slave). As a result, a slave channel interrupt (INTTAUBnIm) is generated at the same time and TAUBTTOUTm remains inactive.
- TAUBnCNTm (slave) generates an interrupt every time the value of TAUBnCDRm is loaded.

(2) Duty cycle = 100%

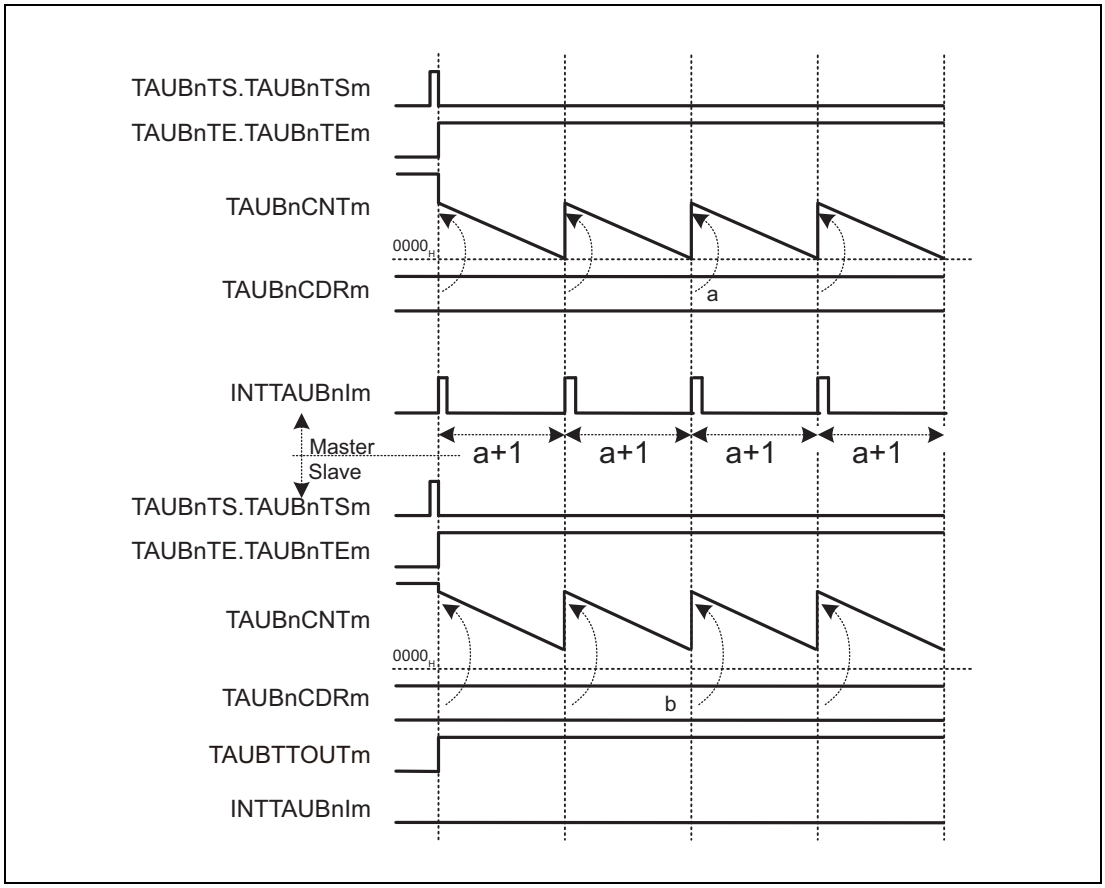


Figure 29.81 TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1, Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

If the value TAUBnCDRm (slave) is higher than the value TAUBnCDRm (master), the counter of the slave channel cannot reach 0000_H and cannot generate interrupts. The TAUBTTOUTm remains at active state.

(3) Operation stop and restart

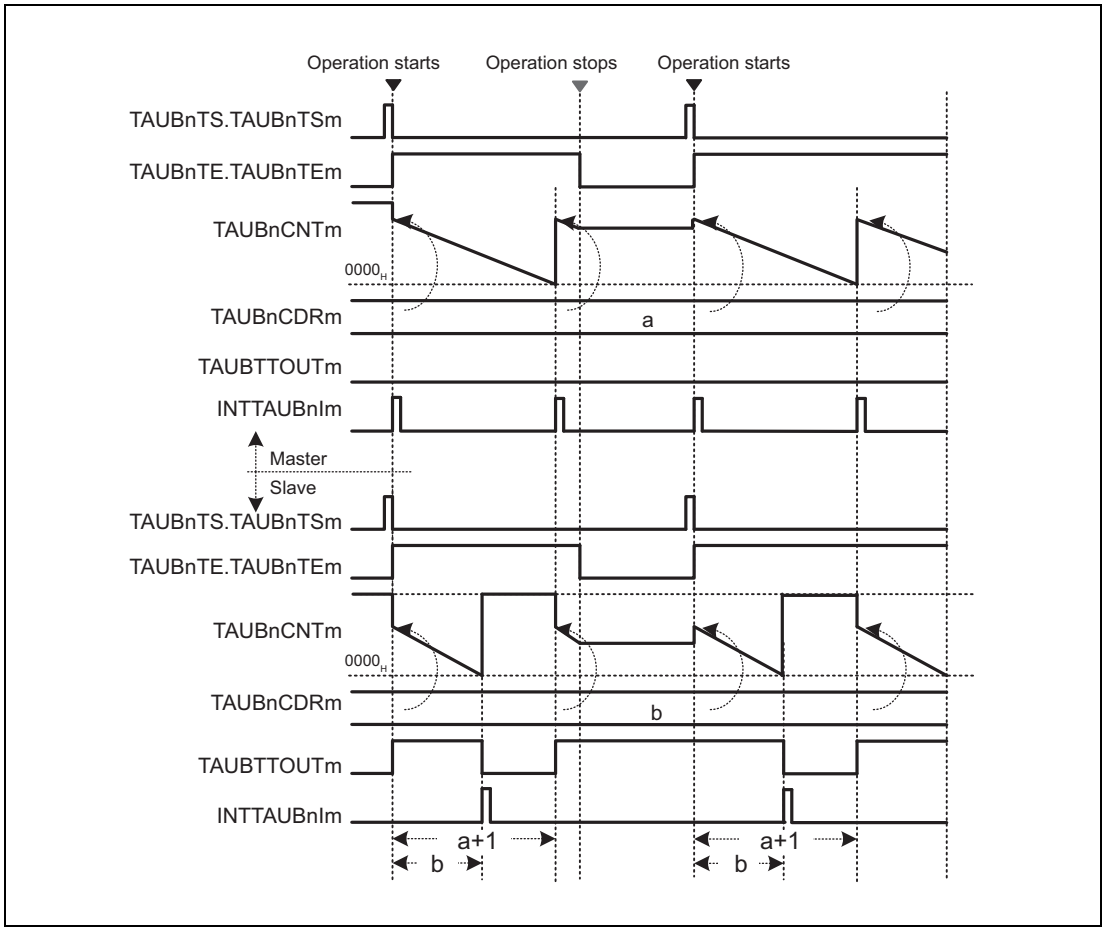


Figure 29.82 Stop and Restart Operation, Positive Logic (TAUBnTOL.TAUBnTOLm (slave) = 0)

- The counter can be stopped by setting TAUBnTT.TAUBnTTm of the master and slave channel(s) to 1, which in turn sets TAUBnTE.TAUBnTEm to 0.
- TAUBnCNTm and TAUBTTOUTm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUBnTS.TAUBnTSm of master and slave channel(s) to 1. TAUBnCNTm of master and slave channel reload the current values of TAUBnCDRm and start to count down from these values.

(4) Operation stop and restart (Slave output, Initialization)

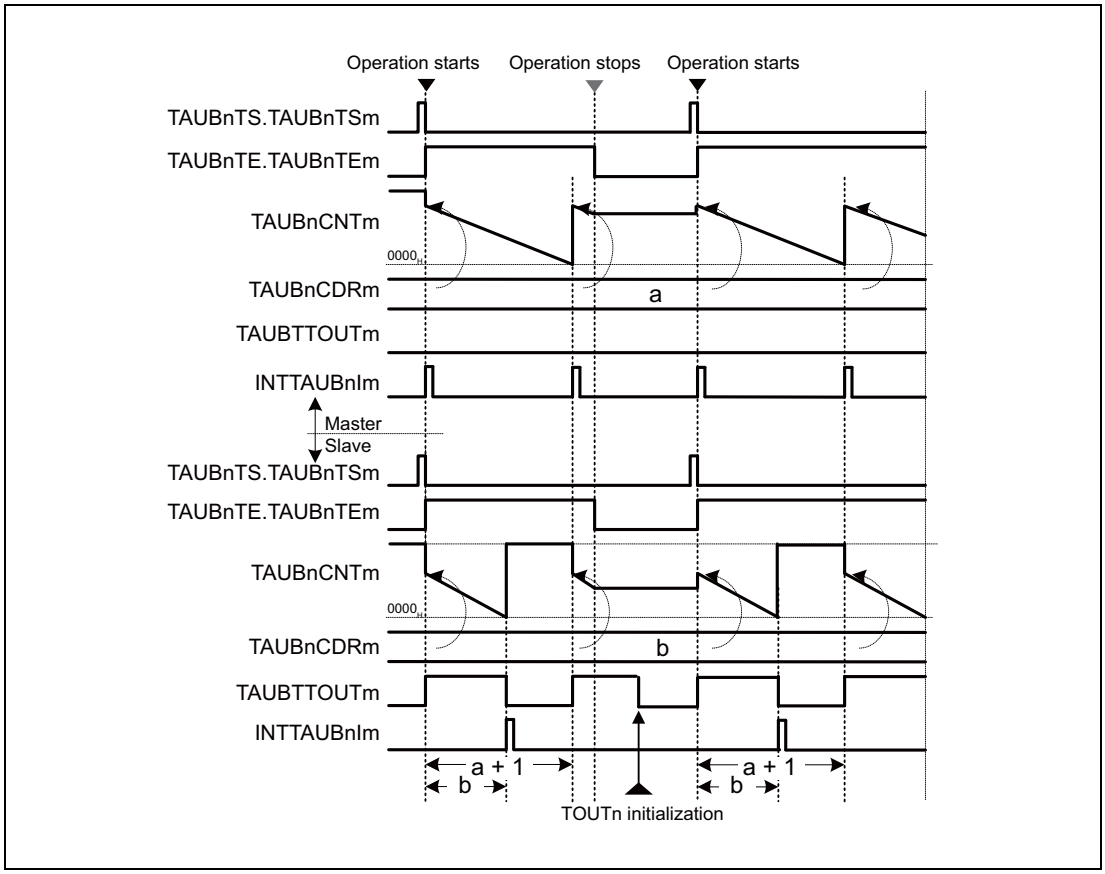


Figure 29.83 Operation Stop and Restart (Slave Output, Initialization)

When TAUBnTOE.TAUBnTOEm of the slave channel is set to 0 while TAUBnTE.TAUBnTEm = 0 and the inactive level of TAUBTTOUTm is written in the TAUBnTO.TAUBnTOm, the output level of TAUBTTOUTm (slave channel) becomes active when INTTAUBnIm is issued when the count operation is started.

29.14.2 One-Shot Pulse Output Function

29.14.2.1 Overview

Summary

This function outputs a signal pulse with a defined pulse width and a specific delay time compared to an external input signal pulse by using a master and a slave channel. The delay time is specified using the master channel. The pulse width is specified using the slave channel.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to one-count mode, see **Table 29.110, Contents of the TAUBnCMORm Register for the Master Channel of the One-Shot Pulse Output Function**
- The operation mode of the slave channel must be set to pulse one-count mode, see **Table 29.113, Contents of the TAUBnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function**
- TAUBTTOUTm is not used for the master channel of this function
- The channel output mode of the slave channel must be set to independent channel output mode 2.
- TAUBTTINm (master) has to be detected while TAUBnCNTm (master) and TAUBnCNTm (slave) await a trigger. Furthermore, the slave is only triggered by an interrupt from the master channel and not by TAUBTTINm (slave).

Description

The counters are enabled by setting the channel trigger bits (TAUBnTS.TAUBnTSm) for master and slave channels to 1. This in turn sets TAUBnTE.TAUBnTEm = 1, enabling count operation.

- Master channel:
  When the next valid TAUBTTINm input edge is detected, the current value of TAUBnCDRm is written to TAUBnCNTm. The counter starts to count down from this value. If TAUBnCMORm.TAUBnMD0 = 0, a trigger (TAUBTTINm) which is detected within the delay time is ignored.
  When the counter of the master channel reaches 0000_H, INTTAUBnIm is generated. The counter returns to $FFFF_H$ and awaits the next valid TAUBTTINm input edge.
- Slave channel:
  The INTTAUBnIm of the master channel triggers the counter of the slave channel. The current value of TAUBnCDRm (slave) is written to TAUBnCNTm (slave) and the counter starts to count down from this value.
  An interrupt is generated and the TAUBTTOUTm signal is set.
  When the counter reaches 0001_H, INTTAUBnIm is generated and the TAUBTTOUTm signal is reset. The counter remains at 0000_H and awaits the next INTTAUBnIm of the master channel.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel stop but retain their values. The counters can be restarted by setting TAUBnTS.TAUBnTSm to 1.

The counter of the master channel can be restarted without stopping it first (forced restart) by setting TAUBnTS.TAUBnTSm to 1 during operation.

NOTES

1. If a forced restart of the counter is executed during operation, the width of the output signal does not correspond to the value of TAUBnCDRm (slave).
2. The TAUBTTINm input signal is sampled at the frequency of the operating clock, specified by TAUBnCMORm.TAUBnCKS[1:0] bits. As a result, the output cycle of TAUBTTOUTm has an error of ± 1 operation clock cycle.

Conditions

- If TAUBnCMORm.TAUBnMD0 of the master channel is set to 0, during counting detected TAUBTTINm input edges are ignored.
- Simultaneous rewrite can be used with this function. Please see Section 29.6, Simultaneous Rewrite.

29.14.2.2 Equations

Delay from trigger input to pulse output
= (TAUBnCDRm (master) + 1) × count clock cycle

Pulse width = (TAUBnCDRm (slave)) × count clock cycle

29.14.2.3 Block Diagram and General Timing Diagram

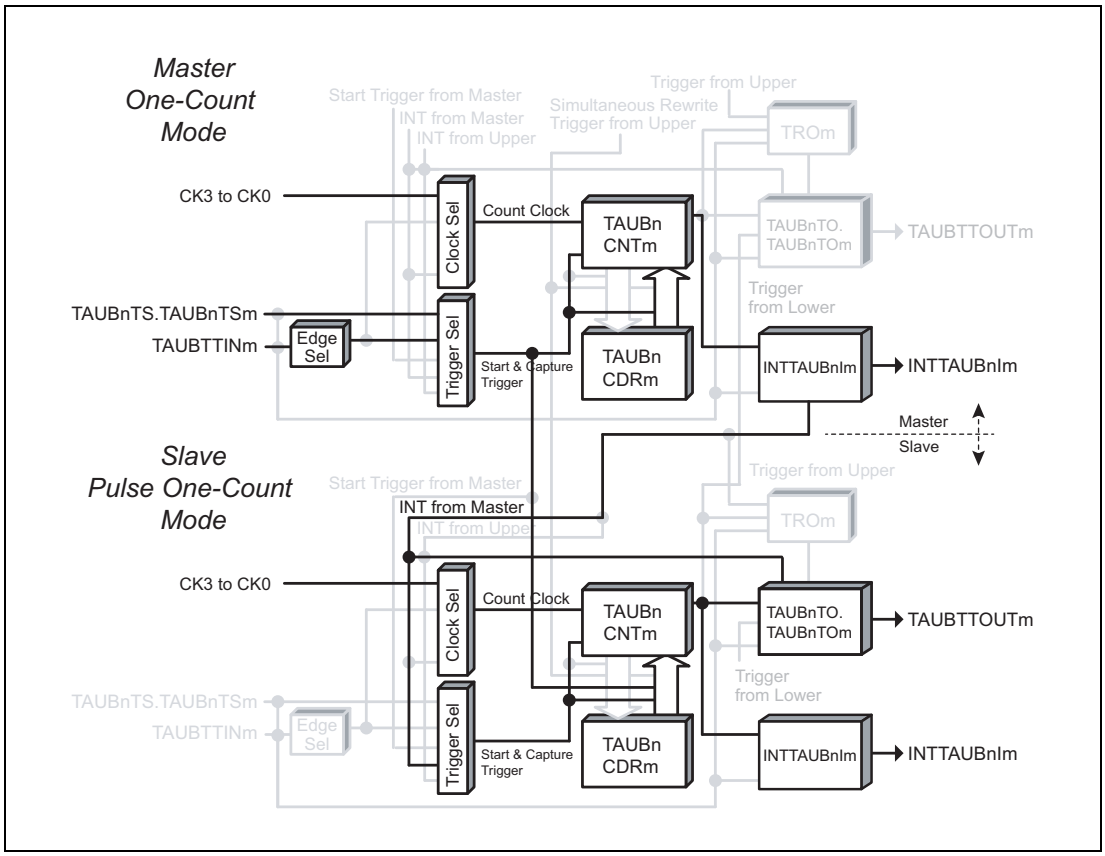


Figure 29.84 Block Diagram for One-Shot Pulse Output Function

The following settings apply to the general basic diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

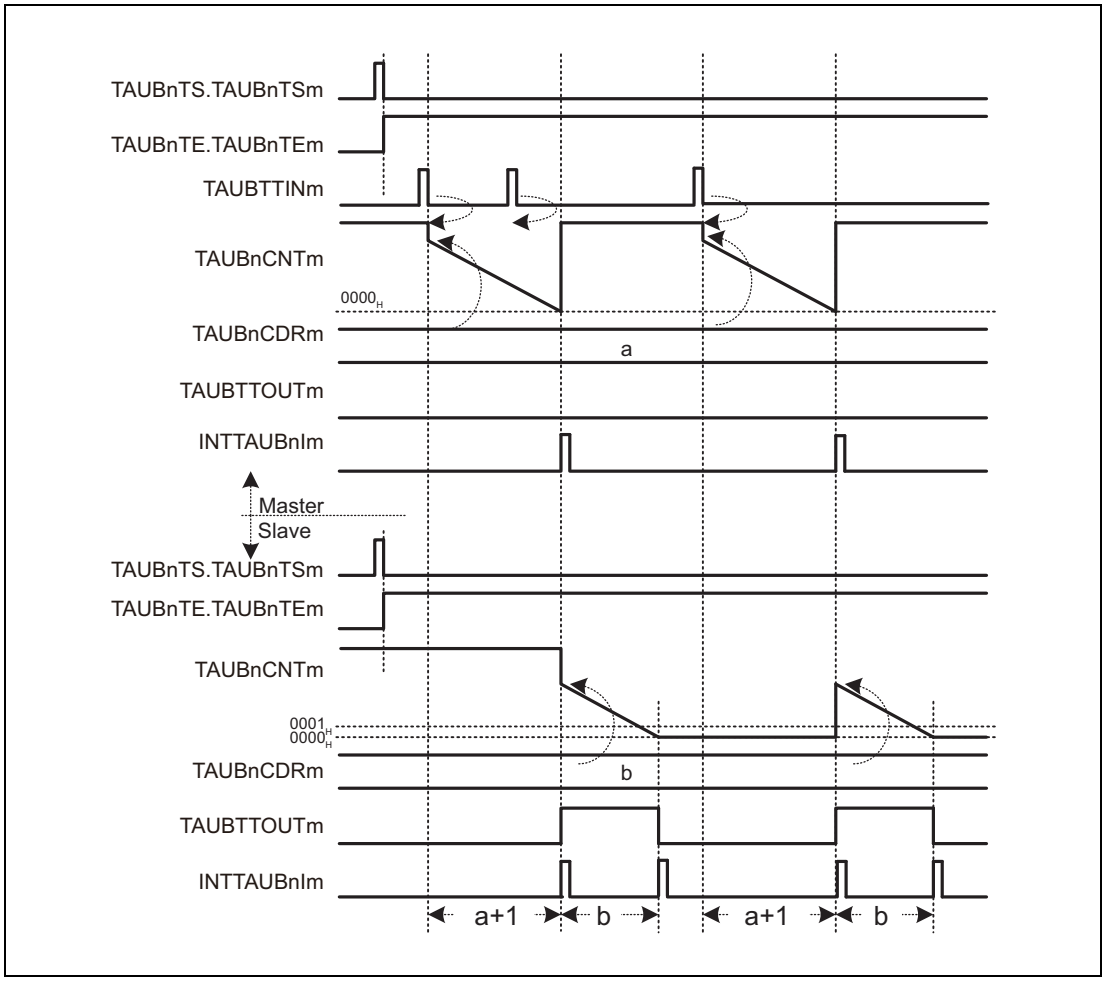


Figure 29.85 General Timing Diagram for One-Shot Pulse Output Function

29.14.2.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.110 Contents of the TAUBnCMORm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 001_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	0: Disables start trigger detection during counting 1: Enables start trigger detection during counting

(2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.111 Contents of the TAUBnCMURm Register for the Master Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode for the master channel

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 29.112 Simultaneous Rewrite Settings for the Master Channel of the One-Shot Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

29.14.2.5 Register Settings for the Slave Channel

(1) TAUBnCMORm for the slave channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.113 Contents of the TAUBnCMORm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 100_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1010_B.
0	TAUBnMD0	0: Disables start trigger detection during counting 1: Enables start trigger detection during counting The value of the MD0 bit of the master and slave channel must be identical.

(2) TAUBnCMURm for the slave channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.114 Contents of the TAUBnCMURm Register for the Slave Channel of the One-Shot Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the slave channel

Table 29.115 Control Bit Settings for Independent Channel Output Mode 2

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	1: Enables independent channel output mode
TAUBnTOM.TAUBnTOMm	0: Independent channel output
TAUBnTOC.TAUBnTOCm	1: Operating mode 2
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	0: Disables dead time operation
TAUBnTDL.TAUBnTDLm	0: When dead time operation is disabled (TAUBnTDE.TAUBnTDEm = 0), set these bits to 0

(4) Simultaneous rewrite for the slave channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 29.116 Simultaneous Rewrite Settings for the Slave Channel of the One-Shot Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

29.14.2.6 Operating Procedure for One-Shot Pulse Output Function

Table 29.117 Operating Procedure for One-Shot Pulse Output Function

		Operation	Status of TAUBn
	Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.2.4, Register Settings for the Master Channel**. Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.2.5, Register Settings for the Slave Channel**. Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.
Restart operation	Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the master channel awaits a TAUBTTINm input.
	During operation	TAUBnCDRm can be changed at any time. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	When a valid TAUBTTINm input edge is detected, TAUBnCNTm of the master channel loads the TAUBnCDRm value and counts down. When the counter reaches 0000_H: • INTTAUBnIm (master) is generated • TAUBnCNTm (master) is reset to $FFFF_H$ and waits for the next valid TAUBTTINm input edge. • TAUBnCNTm (slave) loads the TAUBnCDRm value and starts to count down • INTTAUBnIm (slave) is generated • TAUBTTOUTm (slave) is set to the active level. When TAUBnCNTm (slave) reaches 0001_H: • The counter of TAUBnCNTm (slave) stops. • INTTAUBnIm (slave) is generated • TAUBTTOUTm (slave) is set to an inactive level.
	Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

29.14.2.7 Specific Timing Diagrams

(1) TAUBnCDRm (master) = 0000_H

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

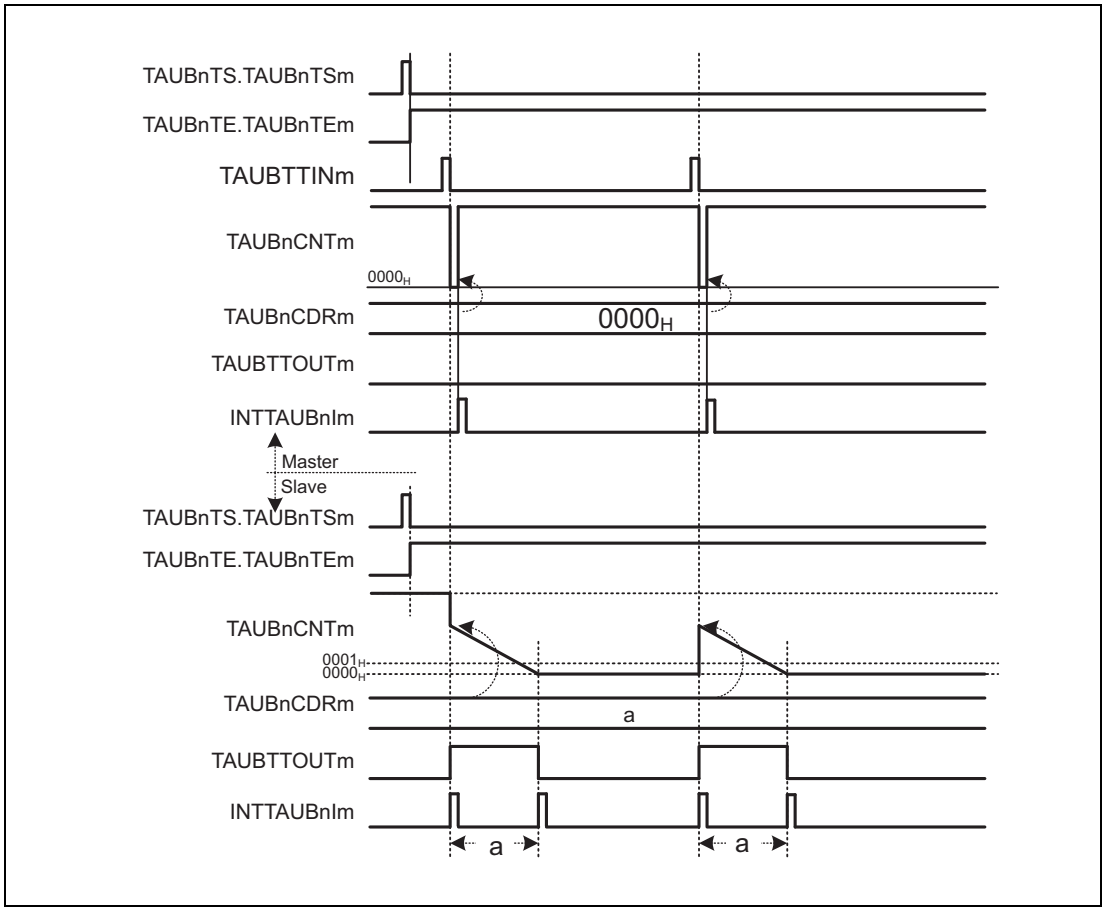


Figure 29.86 TAUBnCDRm (master) = 0000_H

- When a valid TAUBTTINm input edge is detected, the value 0000_H is written to TAUBnCNTm (master). The counter is set to 0000_H for one count and returns to $FFFF_H$.
  Thus, the slave channel starts to count down one count clock later to TAUBTTINm (master).

(2) TAUBnCDRm (slave) = 0000_H

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

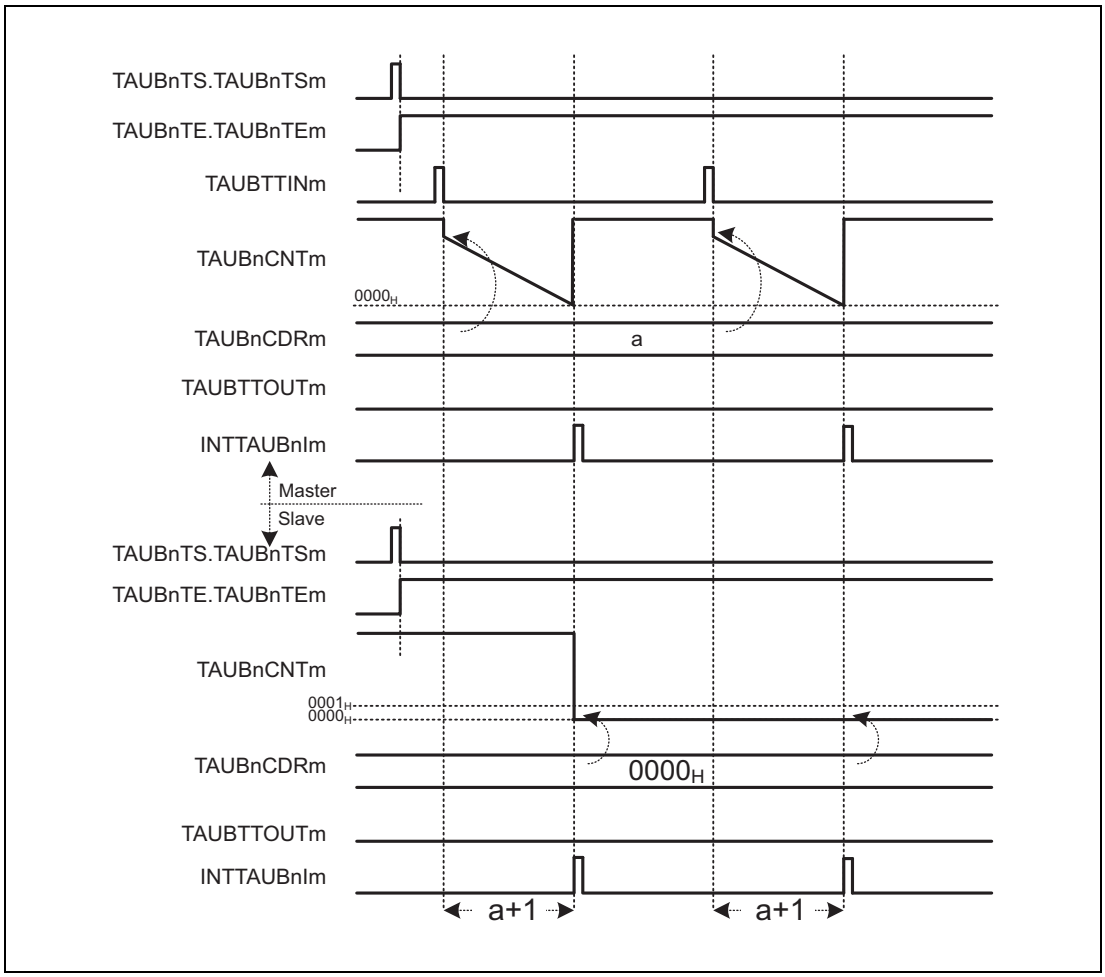


Figure 29.87 TAUBnCDRm (slave) = 0000_H

- TAUBTTOUTm remains at not active state, because the pulse width is zero.

(3) TAUBnCMORm.TAUBnMD0 = 0 (disables start trigger during count operation)

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

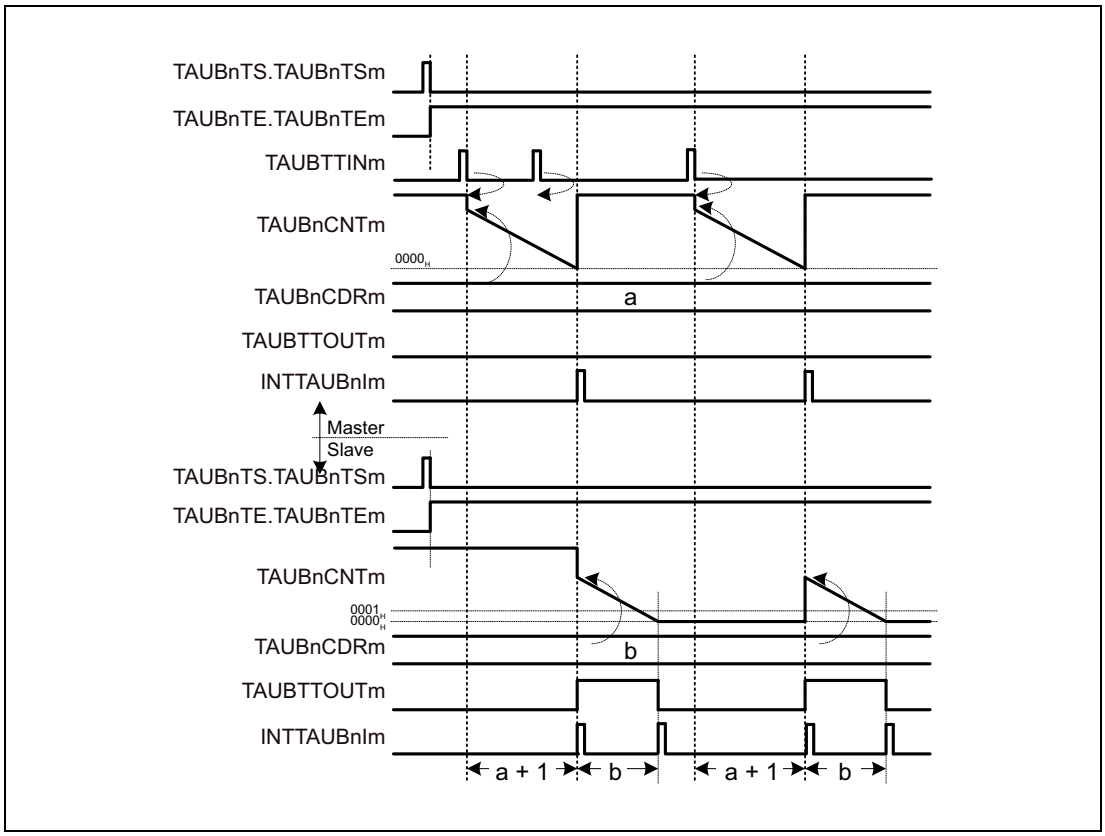


Figure 29.88 TAUBnCMORm.TAUBnMD0 = 0

- Even when an effective edge is input to TAUBTTINm while the counter of the master channel counts down, the counter continues counting down.

(4) TAUBnCMORm.TAUBnMD0 = 1

The following settings apply to this diagram.

- Start trigger detection enabled during counting (TAUBnCMORm.TAUBnMD0 = 1)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

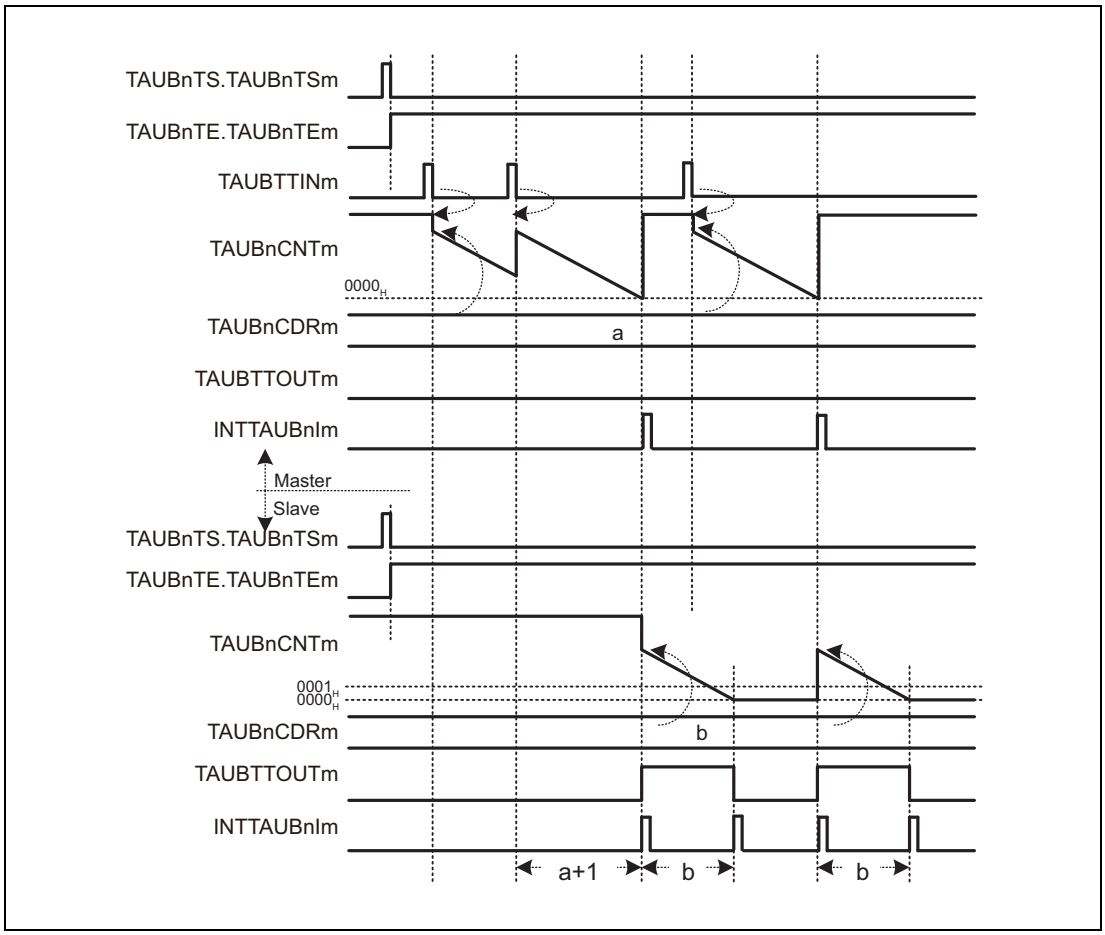


Figure 29.89 TAUBnCMORm.TAUBnMD0 = 1

- If a valid TAUBTTINm input edge is detected while the counter of the master channel counts down, TAUBnCNTm reloads the value of TAUBnCDRm. The counter restarts to count down.

This means the delay for INTTAUBnIm generation interval is extended by the value of TAUBnCNTm at the time when a valid TAUBTTINm input edge is detected.

(5) Stopping and restarting the operation

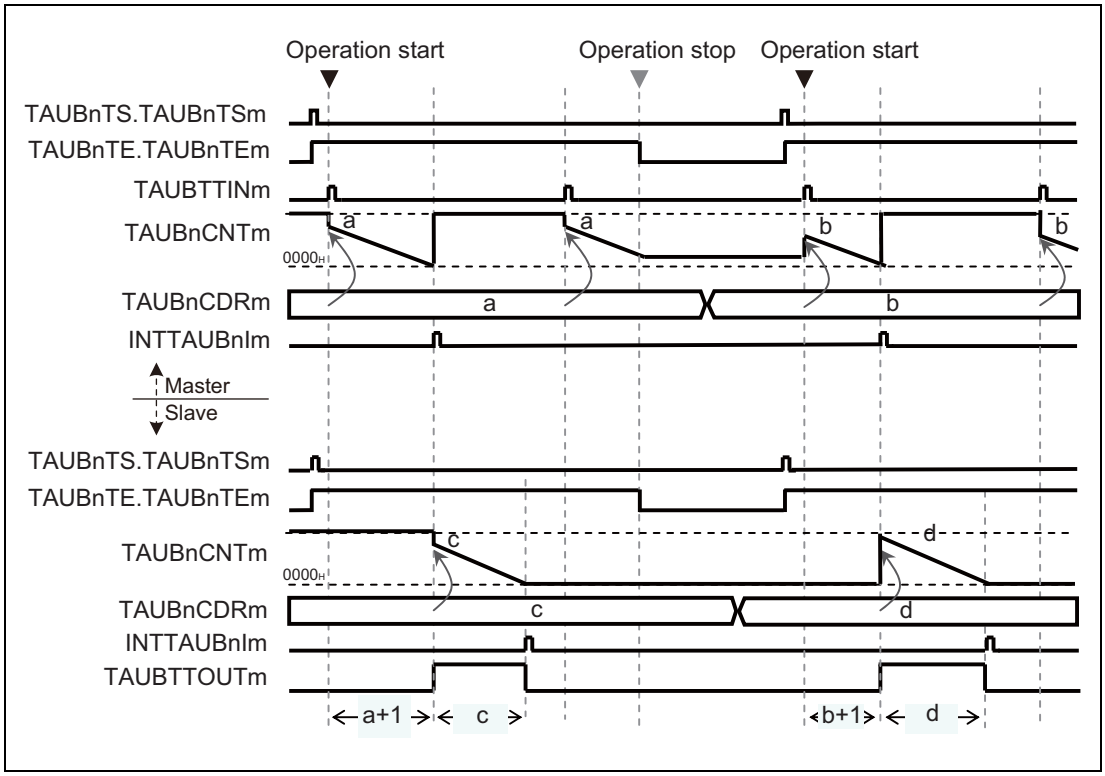


Figure 29.90 Stopping and Restarting the Operation

Setting TTm of the master and slave channels to 1 clears TAUBnTE.TAUBnTEm to 0, thereby stopping the count operation. If this happens, TAUBnCNTm and TAUBTTOUTm stop operation with the values retained.

Setting TAUBnTS.TAUBnTSm of the master and slave channels to 1 concurrently sets TAUBnTE.TAUBnTEm to 1.

When the start trigger is detected while the TAUBnTE.TAUBnTEm is set to 1, the TAUBnCDRm value is transferred to TAUBnCNTm and the operation restarts.

(6) Restarting the master channel while the slave channel is counting

The following settings apply to this diagram.

- Start trigger detection disabled during counting (TAUBnCMORm.TAUBnMD0 = 0)
- Falling edge detection (TAUBnCMURm.TAUBnTIS[1:0] = 00_B)

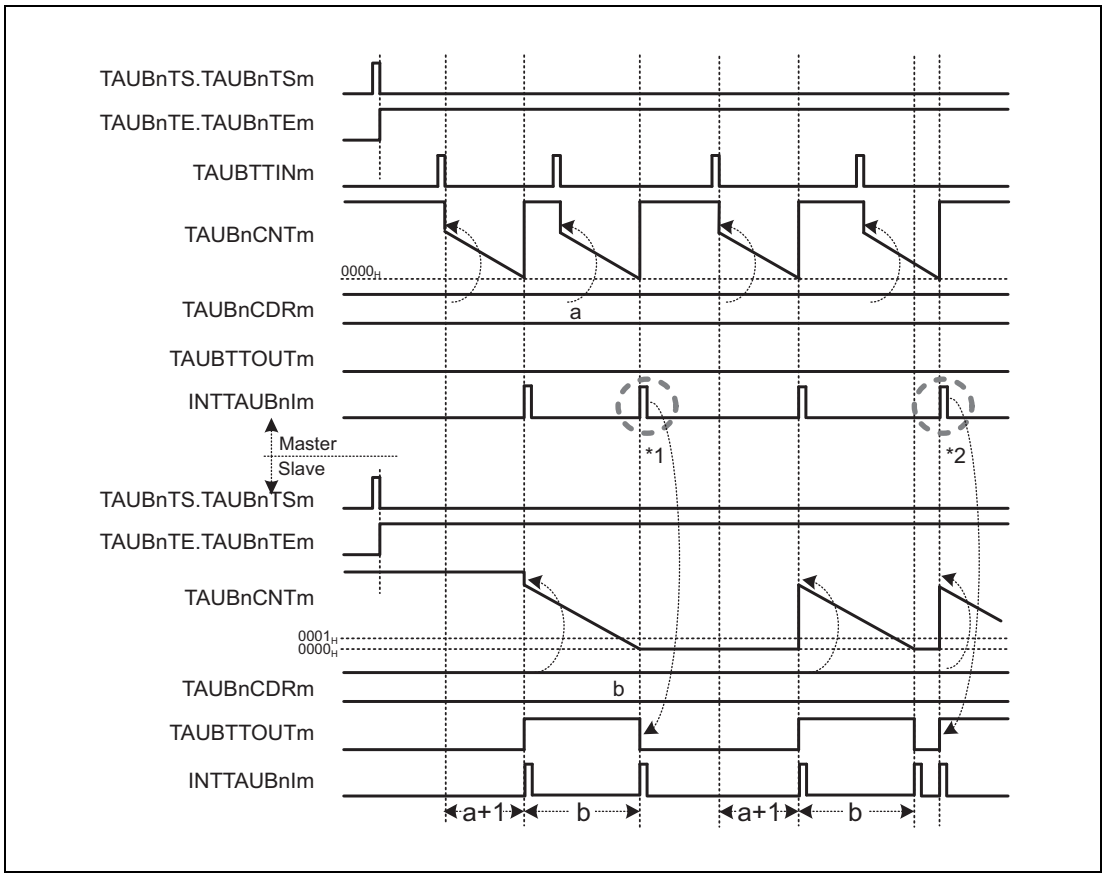


Figure 29.91 Interval of TAUBTTINm ≤ Delay Time + Pulse Width + 1

- If the master channel generates an interrupt before the counter of the slave channel has reached 0001_H or exactly when 0001_H is reached (*1), the interrupt (master) is ignored.
- If an interrupt of the master channel occurs when the counter of the slave channel awaits the next trigger, the value of TAUBnCDRm (slave) is reloaded. An interrupt is generated and TAUBTTOUTm toggles. If TAUBnCNTm (master) has started to count down while the TAUBnCNTm (slave) is still counting (*2), TAUBTTOUTm is not output with the expected delay time.
- To generate the correct one-shot pulse, the start trigger for the master channel must be detected while the master and slave channels are waiting for the start trigger, and not while they are counting.

29.14.3 Delay Pulse Output Function

29.14.3.1 Overview

Summary

This function outputs two signals. The reference signal has a defined pulse width and pulse cycle specified using the master channel and slave channel 1.
Slave channels 2 and 3 output the reference signal with a specified delay. The delay signal is identical to the reference signal, but delayed by amount specified in slave channel 2.

The signal values are specified in the following way:

- The pulse cycle is specified using the master channel.
- The duty cycle of the reference signal is specified using slave channel 1.
  The duty cycle of the delay signal is specified using slave channel 3.
- The delay is specified in slave channel 2.

Prerequisites

- Four channels
- The operation mode of the master channel must be set to interval timer mode, see **Table 29.118, Contents of the TAUBnCMORm Register for the Master Channel of the Delay Pulse Output Function**.
- The operation mode of slave channels 1 and 2 must be set to one-count mode, see **Table 29.121, Contents of the TAUBnCMORm Register for the Slave Channel 1 of the Delay Pulse Output Function** and Table 29.125, Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Delay Pulse Output Function.
- The operation mode of slave channel 3 must be set to pulse one-count mode, see **Table 29.128, Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Delay Pulse Output Function**
- TAUBTTOUTm is not used for the master channel and slave channel 2
- The channel output mode of slave channel 1 must be set to synchronous channel output mode 1.
- The channel output mode of slave channel 3 must be set to independent channel output mode 1.

Description

The counters of the channel group are started by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm to 1, enabling count operation.

- Master channel:

  The current value of TAUBnCDRm is written to TAUBnCNTm and the counter starts to count down from this value. INTTAUBnIm is generated on the master channel.
  When the counter of the master channel reaches 0000_H, pulse cycle time has elapsed and INTTAUBnIm is generated. The counter reloads the TAUBnCDRm value and counts down.

- Slave channels 1 and 2:

When the slave channels 1 and 2 detect an interrupt from the master channel, they start to count down from the current value of TAUBnCDRm. The TAUBTTOUTm signal (slave 1) is set.

– Slave channel 1:
When the counter of slave channel 1 reaches 0000_H (duty time has elapsed) INTTAUBnIm is generated and the TAUBTTOUTm signal is reset. The counter returns to $FFFF_H$ and awaits the next INTTAUBnIm of the master channel.

– Slave channel 2:
When the counter of slave channel 2 reaches 0000_H, delay time has elapsed and INTTAUBnIm is generated. The counter returns to $FFFF_H$ and awaits the next INTTAUBnIm of the master channel.
INTTAUBnIm (slave channel 2) triggers the counter of slave channel 3

- Slave channel 3:
When slave channel 3 detects an interrupt from slave channel 2, it starts to count down from the current value of TAUBnCDRm. INTTAUBnIm is generated and the TAUBTTOUTm signal (slave 3) is set.
When the counter of slave channel 3 reaches 0001_H, INTTAUBnIm is generated and the TAUBTTOUTm signal is reset.
The output from slave channel 3 is the delayed PWM pulse.

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channels, which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUBnTS.TAUBnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. Please see Section 29.6, Simultaneous Rewrite.

29.14.3.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

Duty width 1 = (TAUBnCDRm (slave 1)) × count clock cycle

Delay = (TAUBnCDRm (slave 2) + 1) × count clock cycle

Duty width 2 = (TAUBnCDRm (slave 3)) × count clock cycle

Where the setting of the delay is within the following range:

$0000_H \leq$ TAUBnCDRm (slave 2) < TAUBnCDRm (master)

NOTES

1. The output waveform of TAUBTTOUTm (slave 3) is the output waveform of TAUBTTOUTm (slave 1) delayed for the delay generated by slave 2. It cannot be delayed for more than the pulse cycle.
2. When INTTAUBnIm of slave 2 occurs while slave 3 is counting, slave 3 restarts the operation. Therefore, the output waveform of TAUBTTOUTm (slave 3) retains the active level. (In this case, TAUBTTOUTm (Slave-CH-3) cannot output the waveform of the delayed basic pulse of TAUBTTOUTm (Slave-CH-1).)

29.14.3.3 Block Diagram and General Timing Diagram

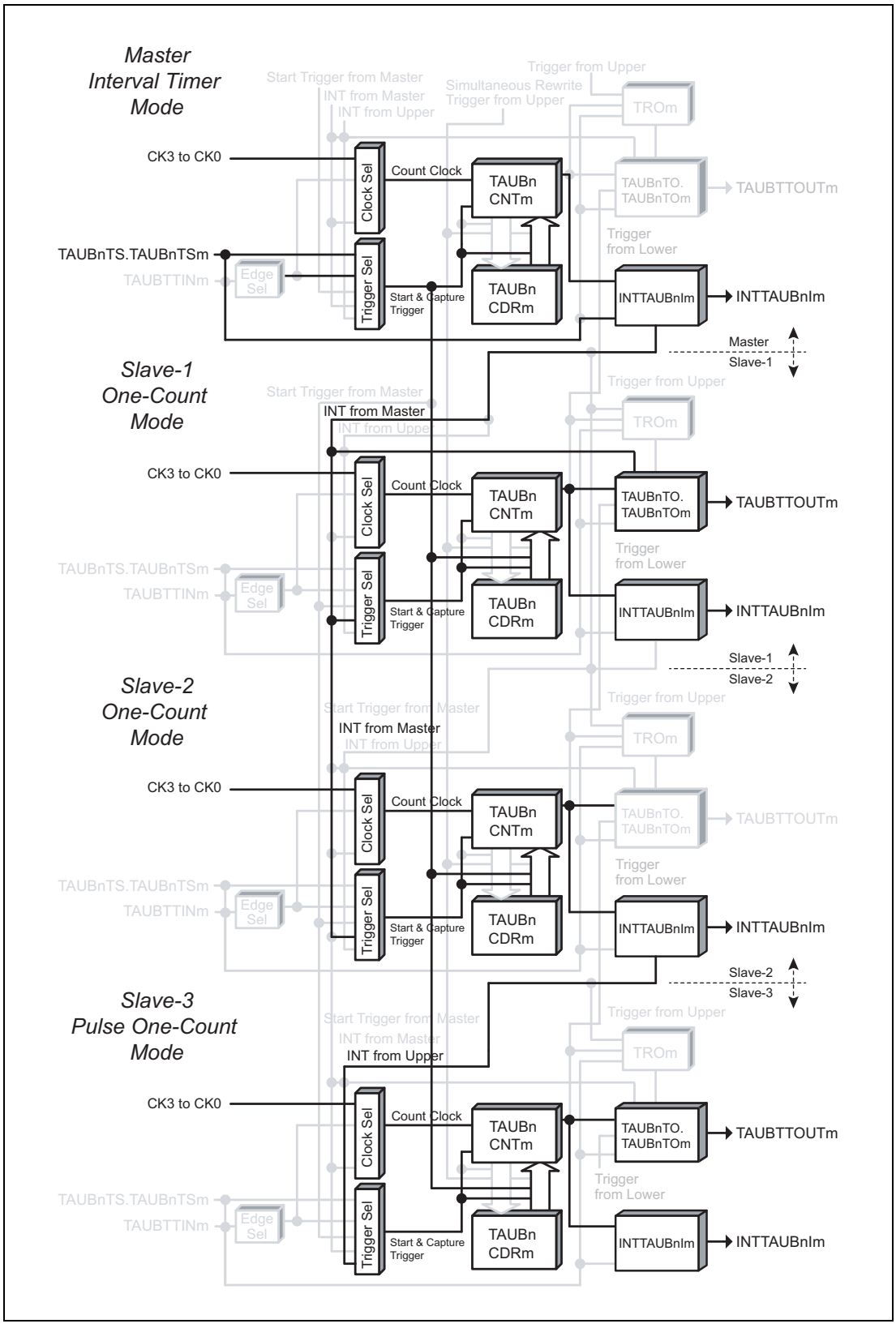


Figure 29.92 Block Diagram for Delay Pulse Output Function

The following settings apply to the general timing diagram.

- Slave channel 1: Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3: Positive logic (TAUBnTOL.TAUBnTOLm = 0)

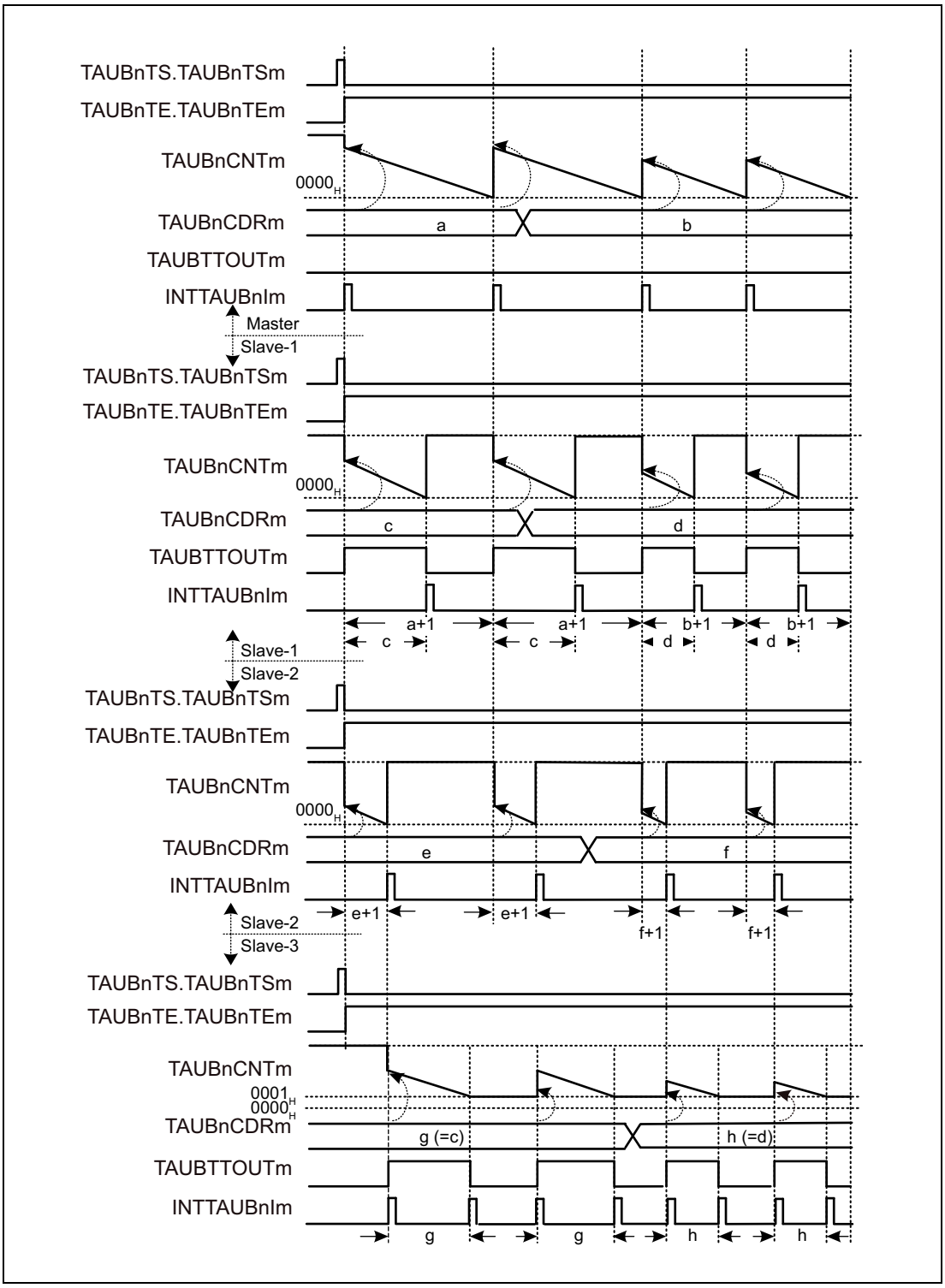


Figure 29.93 General Timing Diagram for Delay Pulse Output Function

29.14.3.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.118 Contents of the TAUBnCMORm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.119 Contents of the TAUBnCMURm Register for the Master Channel of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by the master channel of this function.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 29.120 Simultaneous Rewrite Settings for the Master Channel of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

29.14.3.5 Register Settings for Slave Channel 1

(1) TAUBnCMORm for slave channel 1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.121 Contents of the TAUBnCMORm Register for the Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel must be identical.
13	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 100_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for slave channel 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.122 Contents of the TAUBnCMURm Register for the Slave Channel 1 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 1

Table 29.123 Control Bit Settings for Slave Channel 1 of the Synchronous Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for slave channel 1

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 29.124 Simultaneous Rewrite Settings for Slave Channel 1 of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

29.14.3.6 Register Settings For Slave Channel 2

(1) TAUBnCMORm for slave channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.125 Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 100_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for slave channel 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.126 Contents of the TAUBnCMURm Register for the Slave Channel 2 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 2

Set TAUBnTOE.TAUBnTOEm to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite for slave channel 2

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 29.127 Simultaneous Rewrite Settings for Slave Channel 2 of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

29.14.3.7 Register Settings for Slave Channel 3

(1) TAUBnCMORm for slave channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.128 Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 101_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1010_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for slave channel 3

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.129 Contents of the TAUBnCMURm Register for the Slave Channel 3 of the Delay Pulse Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 3

Table 29.130 Control Bit Settings for Independent Channel Output Mode 2

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for slave channel 3

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 29.131 Simultaneous Rewrite Settings for Slave Channel 3 of the Delay Pulse Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUBnRDM.TAUBnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

29.14.3.8 Operating Procedure for Delay Pulse Output Function

Table 29.132 Operating Procedure for Delay Pulse Output Function (1/2)

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.3.4, Register Settings for the Master Channel**. Slave channel 1: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.3.5, Register Settings for Slave Channel 1**. Slave channel 2: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.3.6, Register Settings For Slave Channel 2**. Slave channel 3: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.3.7, Register Settings for Slave Channel 3**. Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.

Table 29.132 Operating Procedure for Delay Pulse Output Function (2/2)

	Operation	Status of TAUBn
Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master channel and slave channels 1 and 2 start. INTTAUBnIm is generated on the master channel and TAUBTTOUTm (slave 1) is set.
During operation	TAUBnCDRm can be changed at any time. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	TAUBnCNTm of the master channel and slave channels 1 and 2 load TAUBnCDRm and count down. When the counter of the master channel reaches 0000_H: • INTTAUBnIm (master) is generated • TAUBnCNTm (master) loads the TAUBnCDRm value and continues count operation • TAUBnCNTm (slave 1 and slave 2) reload the TAUBnCDRm value and start counting down • TAUBTTOUTm (slave 1) is set When TAUBnCNTm (slave 1) reaches 0000_H: • INTTAUBnIm (slave 1) is generated • TAUBTTOUTm (slave 1) is reset When TAUBnCNTm (slave 2) reaches 0000_H: • INTTAUBnIm (slave 2) is generated • INTTAUBnIm (slave 3) is generated • TAUBTTOUTm (slave 3) is set • TAUBnCNTm (slave 3) reloads the TAUBnCDRm value and starts counting down When TAUBnCNTm (slave 3) reaches 0001_H: • INTTAUBnIm (slave 3) is generated • TAUBTTOUTm (slave 3) is reset
Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

29.14.3.9 Specific Timing Diagrams

(1) Duty cycle (slave 3) = 100%

The following values apply to the figure below.

- TAUBnCDRm (master) = $000A_H$
- TAUBnCDRm (slave 1) = $000B_H$
- TAUBnCDRm (slave 2) = 0000_H
- TAUBnCDRm (slave 3) = $000B_H$

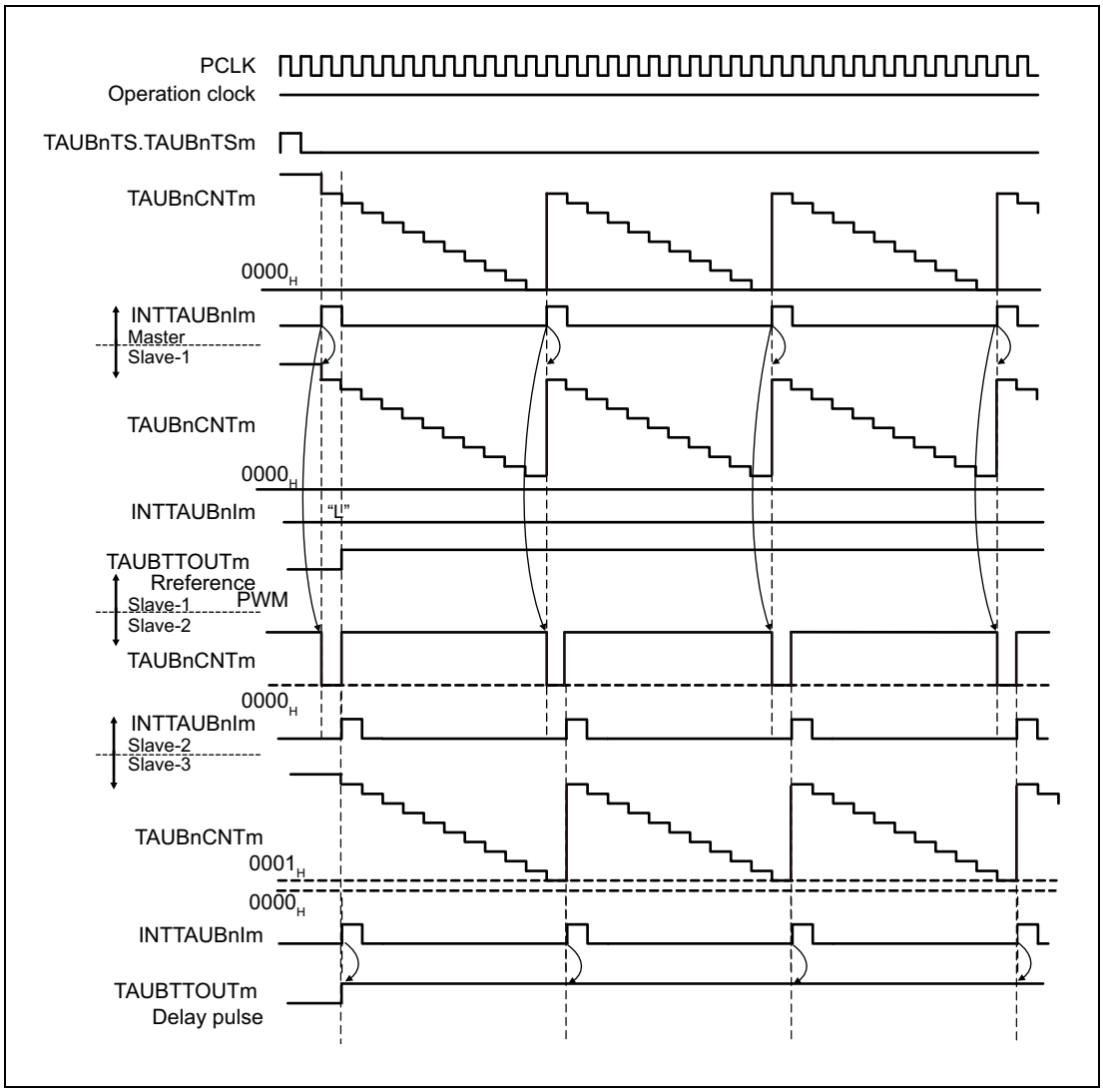


Figure 29.94 Duty Cycle (slave 3) = 100%

- If the value of TAUBnCDRm (slave 1 and 3) is higher than the value of TAUBnCDRm (master), the counter of the slave channel 1 cannot reach 0000_H and cannot generate interrupt request signals. TAUBTTOUTm of channels 1 and 3 remain in the active state.

(2) TAUBTTOUTm (slave 1) = TAUBTTOUTm (slave 3)

The following values apply to the figure below.

- TAUBnCDRm (master) = $000A_H$
- TAUBnCDRm (slave 1) = 0005_H
- TAUBnCDRm (slave 2) = 0000_H
- TAUBnCDRm (slave 3) = 0005_H

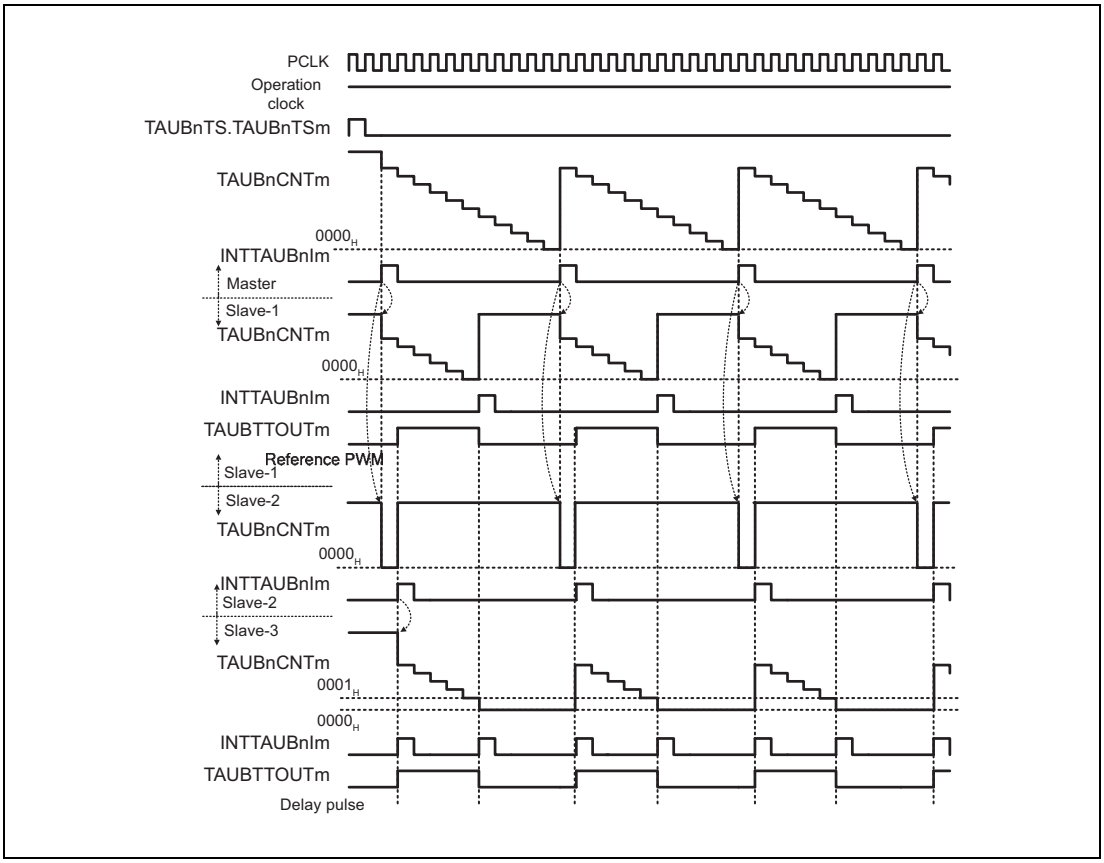


Figure 29.95 TAUBTTOUTm (slave 1) = TAUBTTOUTm (slave 3)

- If TAUBnCDRm (slave 2) = 0000_H, the counter of slave channel 3 starts counting one count clock later than the counter of slave channel 1. The reference pulse and the delay pulse are output with a delay of one clock count.

29.14.4 AD Conversion Trigger Output Function Type 1

29.14.4.1 Overview

Summary

This function is identical to Section 29.14.1, PWM Output Function except that TAUBTTOUTm is not output.

This is achieved by setting the channel output mode of the slave to independent channel output mode controlled by software.

29.14.4.2 Block Diagram and General Timing Diagram

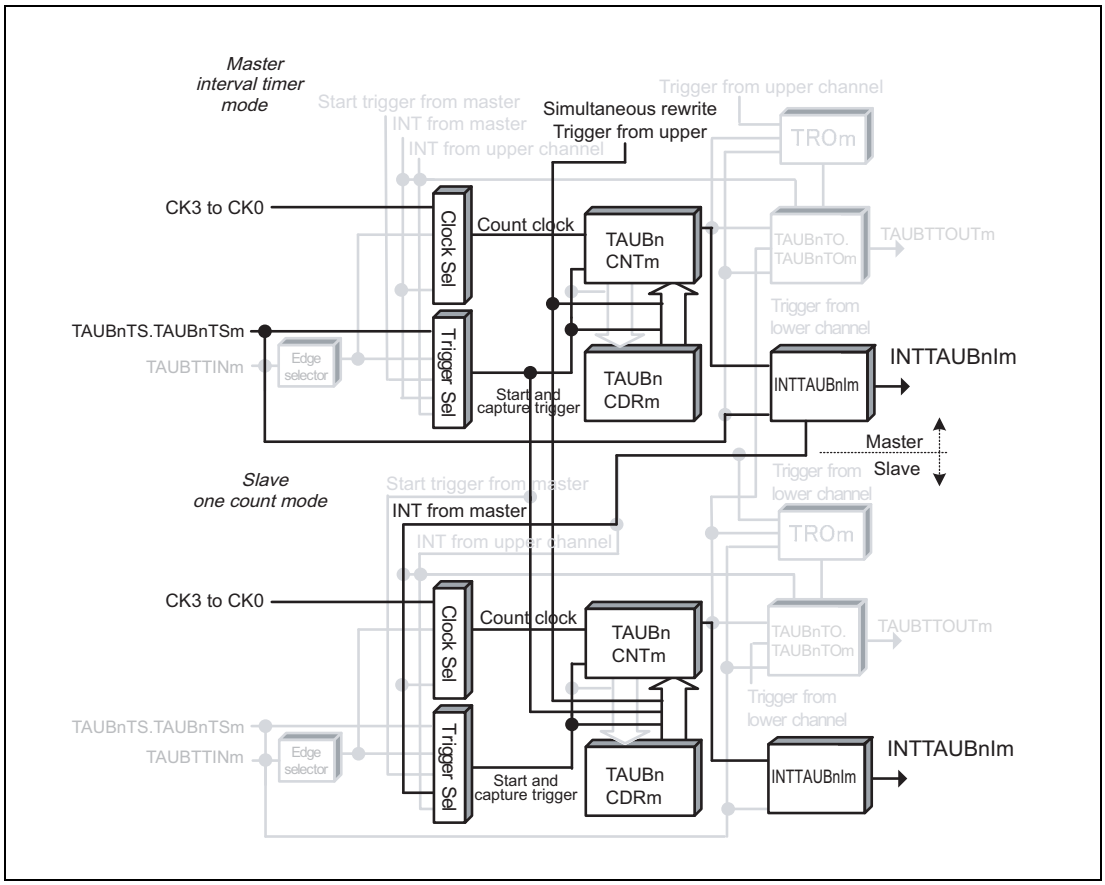


Figure 29.96 Block Diagram for AD Conversion Trigger Output Function Type 1

The following settings apply to the general timing diagram.

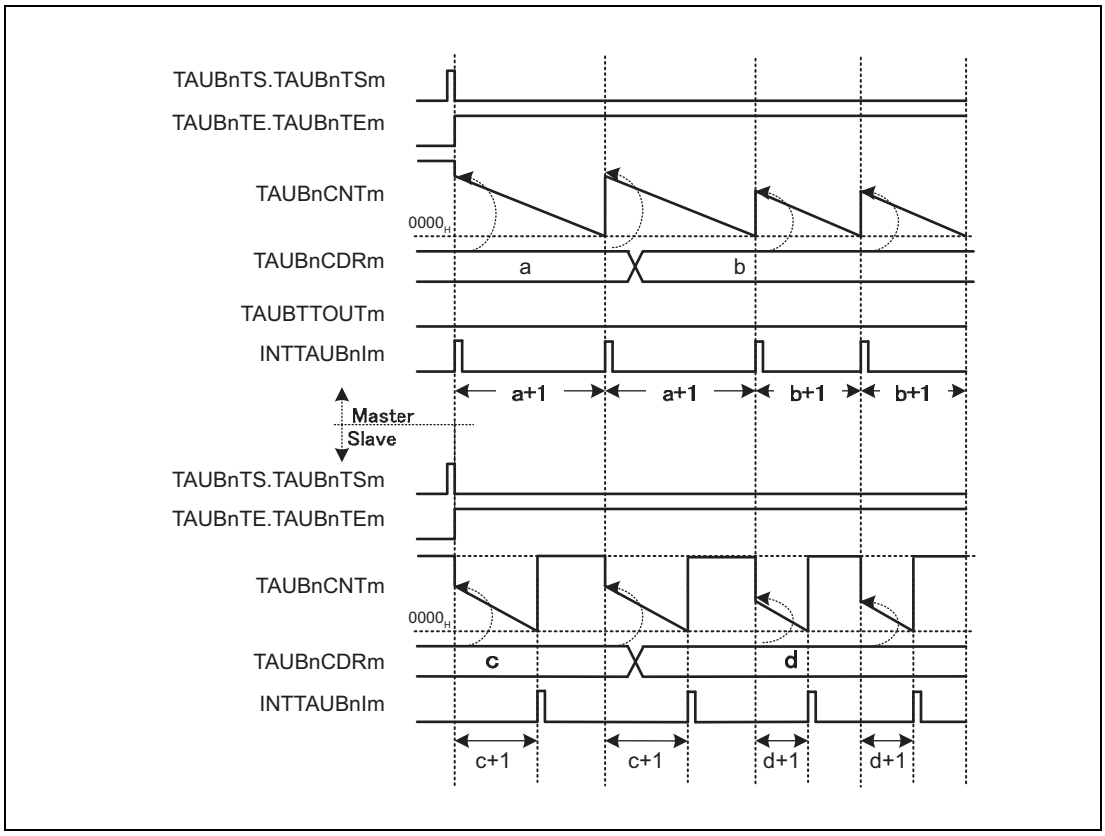


Figure 29.97 General Timing Diagram for AD Conversion Trigger Output Function Type 1

29.14.5 Triangle PWM Output Function

29.14.5.1 Overview

Summary

This function generates multiple triangle PWM outputs by using a master and one or more slave channels. It enables the pulse cycle (frequency) and the duty cycle of TAUBTTOUTm to be set using the master and slave channel(s) respectively.

The master channel generates a carrier cycle from two pulse cycles. The first cycle of the master channel controls the down status and the second cycle controls the up status of the slaves counter.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to interval timer mode, see **Table 29.133, Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function**.
- The operation mode of the slave channel(s) must be set to up down count mode, see Table 29.137, Contents of the TAUBnCMORm Register for the Slave Channel of the Triangle PWM Output Function.
- The channel output mode of the master channel must be set to independent channel output mode 1.
- The channel output mode of the slave channel(s) must be set to synchronous channel output mode 2.
- The following settings establish TAUBTTOUTm at high level for the down status of the carrier cycle.
  - If the TAUBnCMORm.TAUBnMD0 (master) bit is set to 0,TAUBnTO.TAUBnTOm must be set to 1 while TAUBnTOE.TAUBnTOEm is 0. (recommended)
  - If the TAUBnCMORm.TAUBnMD0 (master) bit is set to 1, TAUBnTO.TAUBnTOm must be set to 0 while TAUBnTOE.TAUBnOEm is 0.

Functional description

The counters are started by setting the channel trigger bit (TAUBnTS.TAUBnTSm) to 1 for every channel. This in turn sets TAUBnTE.TAUBnTEm, enabling count operation. The current values of TAUBnCDRm (master and slave) are written to TAUBnCNTm (master and slave) and the counters start to count down from these values. If the master channel TAUBnCMORm.TAUBnMD0 bit is set to 1, an interrupt is generated and TAUBTTOUTm signal of the master toggles.

- Master channel:
  When the counter of the master channel reaches 0000_H (pulse cycle time has elapsed) INTTAUBnIm is generated and the TAUBTTOUTm signal toggles. TAUBnCNTm then reloads the TAUBnCDRm value and counts down.
- Slave channel:
  The INTTAUBnIm of the master channel triggers the counter of the slave channel:

– If the slave counter currently counts down, it changes count direction.
– If the slave counter currently counts up, the value of TAUBnCDRm is reloaded and the counter counts down.

When the counter of the slave channel reaches 0001_H while counting up or down, INTTAUBnIm is generated and the TAUBTTOUTm (slave) signal is set or reset.

The counter continues to count down or up and awaits the next INTTAUBnIm of the master channel.

TAUBTTOUTm can be switched between positive and negative phase setting TAUBnTOL.TAUBnTOLm during operation.

The counters can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel(s), which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel(s) stop but retain their values.

Conditions

Simultaneous rewrite can be used with this function. Please see Section 29.6, Simultaneous Rewrite.

29.14.5.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUBnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUBnCDRm (master) + 1) × 2 × count clock cycle

Duty cycle =

[(TAUBnCDRm (master) + 1 - TAUBnCDRm (slave))/ (TAUBnCDRm (master) + 1)] × 100

– Duty cycle = 100%
  TAUBnCDRm (slave) = 0000_H
– Duty cycle = 0%
  TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1

29.14.5.3 Block Diagram and General Timing Diagram

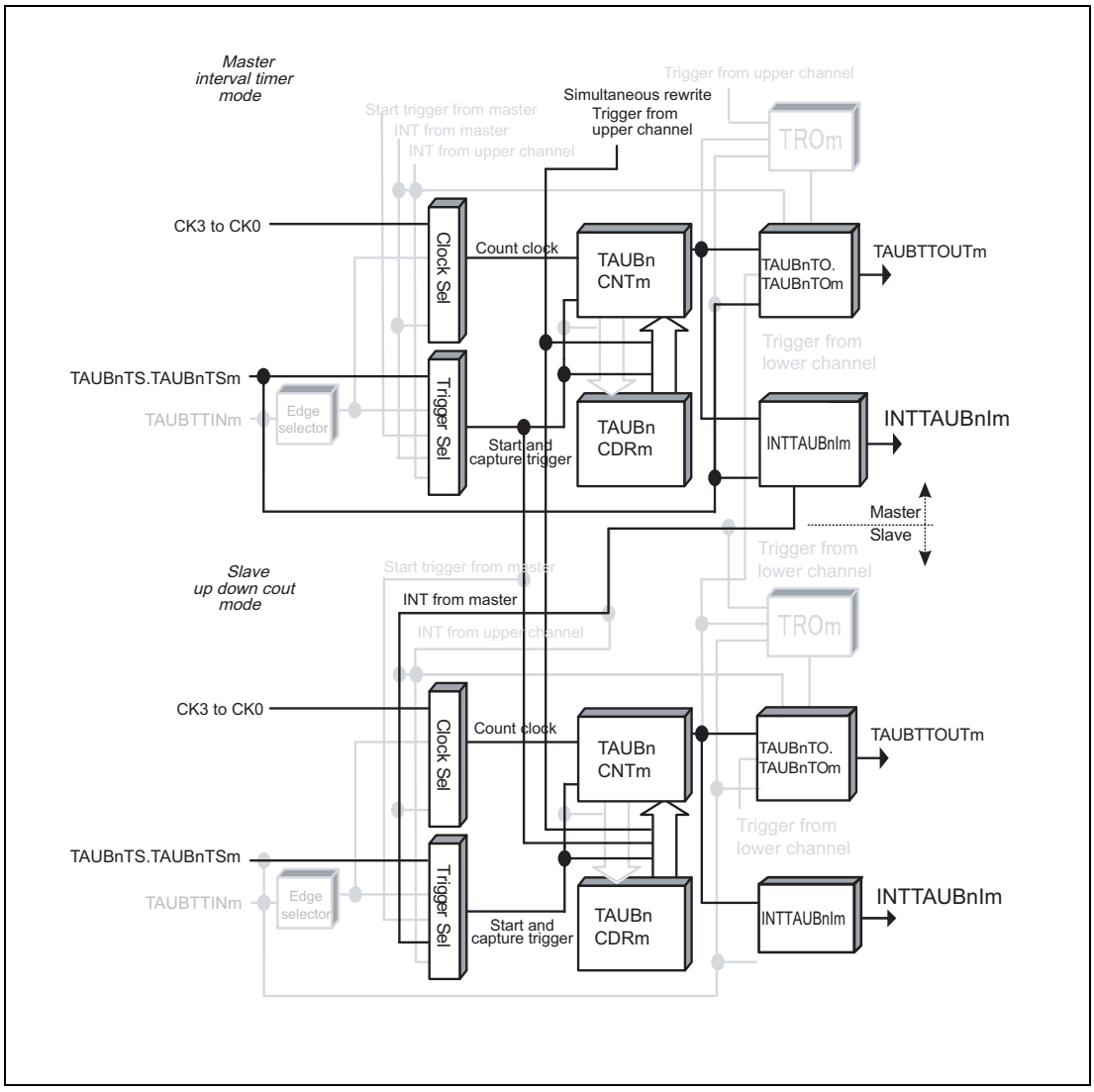


Figure 29.98 Block Diagram for Triangle PWM Output Function

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

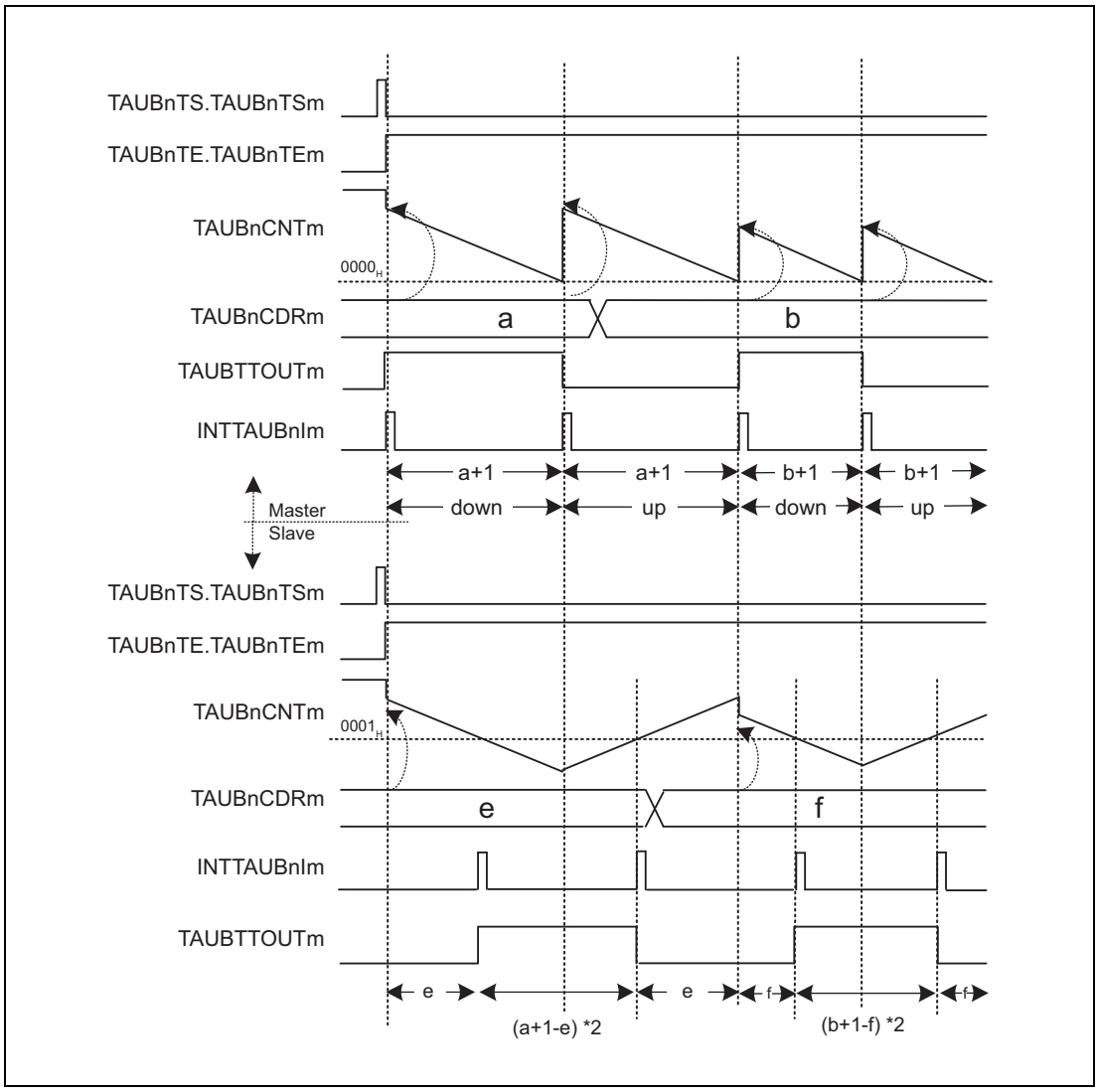


Figure 29.99 General Timing Diagram for Triangle PWM Output Function

29.14.5.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.133 Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm not generated and TAUBTTOUTm does not toggle at operation start 1: Generates INTTAUBnIm and toggles TAUBTTOUTm at operation start

(2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.134 Contents of the TAUBnCMURm Register for the Master Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

Table 29.135 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	Write 0_B.
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 29.136 Simultaneous Rewrite Settings for the Master Channel of the Triangle PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

NOTE

If TAUBnRDS.TAUBnRDSm = 1, the master channel requires an upper channel that generates the simultaneous rewrite trigger signal.

29.14.5.5 Register Settings for the Slave Channel(s)

(1) TAUBnCMORm for the slave channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.137 Contents of the TAUBnCMORm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 111_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1001_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm for the slave channel(s)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.138 Contents of the TAUBnCMURm Register for the Slave Channel of the Triangle PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the slave channel(s)

Table 29.139 Control Bit Settings for Synchronous Channel Output Mode 2

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 29.140 Simultaneous Rewrite Settings for the Slave Channel of the Triangle PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

29.14.5.6 Operating Procedure for Triangle PWM Output Function

Table 29.141 Operating Procedure for Triangle PWM Output Function

		Operation	Status of TAUBn
	Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.5.4, Register Settings for the Master Channel** Slave channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.5.5, Register Settings for the Slave Channel(s)** Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.
Restart operation	Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUBnIm (master) is generated on the master channel when TAUBnCMORm.TAUBnMD0 set to 1.
	During operation	TAUBnCDRm can be changed at any time. TAUBnTOL.TAUBnTOLm can be changed. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	TAUBnCNTm of the master and slave channel loads TAUBnCDRm and counts down. When the counter of the master channel reaches 0000_H: • INTTAUBnIm (master) is generated • TAUBTTOUTm (master) toggles • TAUBnCNTm (master) loads the TAUBnCDRm value and continues count operation. • TAUBnCNTm (slave) loads the TAUBnCDRm value or counts in the reverse direction. When TAUBnCNTm of the slave = 0001_H: • INTTAUBnIm (slave) is generated • TAUBTTOUTm (slave) is set (in count-down status) or reset (in count-up status)
	Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

29.14.5.7 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
  - TAUBnCDRm = a = 5_H
- Slave channel:
  - TAUBnCDRm = 6_H

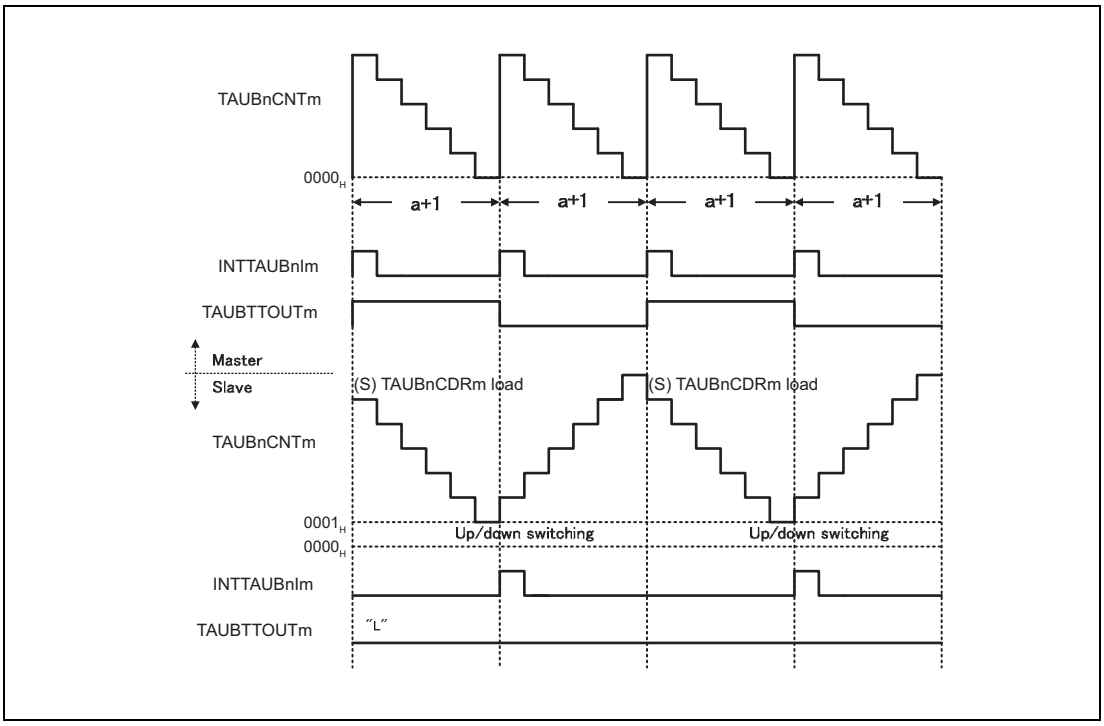


Figure 29.100 TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1

- If TAUBnCDRm (slave) ≥ TAUBnCDRm (master) + 1, INTTAUBnIm of slave channel is not generated during counting down. The set signal is never detected, so TAUBTTOUTm remains at low state.

(2) Duty cycle = 100%

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
  - TAUBnCDRm = a = 5_H
- Slave channel:
  - TAUBnCDRm = 0_B

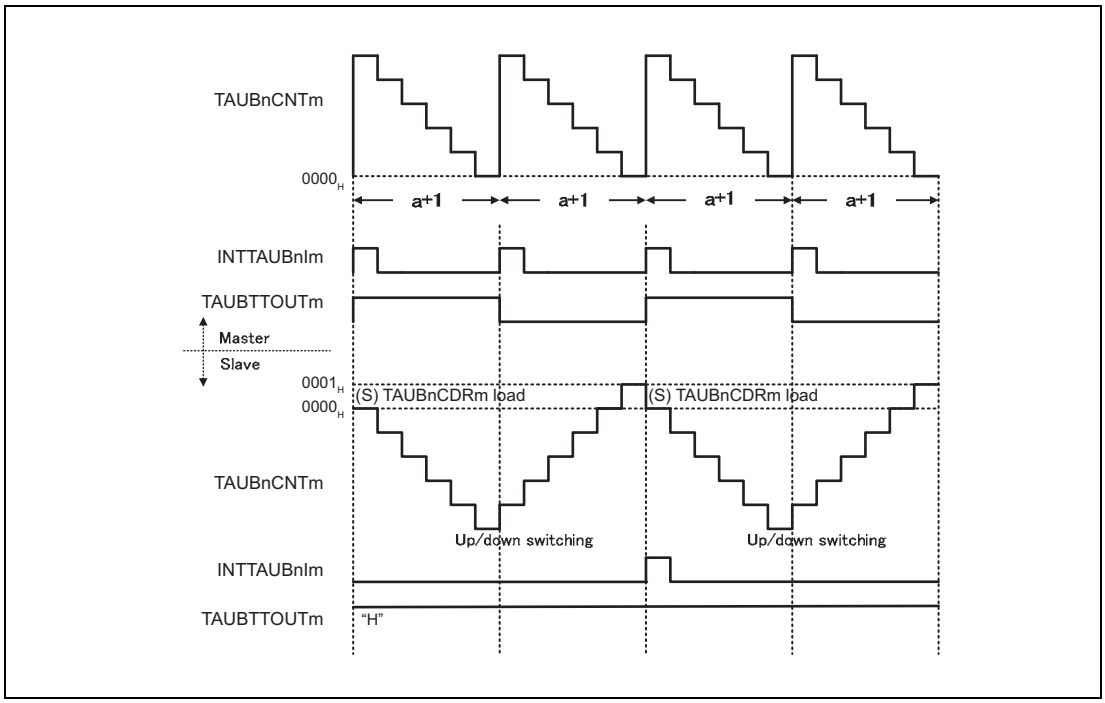


Figure 29.101 TAUBnCDRm (slave) = 0000_H

- If TAUBnCDRm (slave) = 0000_H, INTTAUBnIm of slave channel is not generated during counting up. The reset signal is never detected, so TAUBTTOUTm remains at high state.

29.14.6 Triangle PWM Output Function with Dead Time

29.14.6.1 Overview

Summary

This function generates multiple triangle PWM outputs with a defined dead time by using a master and two or more slave channels. The resulting PWM signals with the dead time are output via TAUBTTOUTm of the slave channels 2 and 3. It enables the pulse cycle (frequency) and the duty cycle of TAUBTTOUTm to be set using the master and slave channel(s) respectively.

The master generates a carrier cycle. The first pulse controls the down status and the second pulse controls the up status of the slaves counter.

An interrupt on slave 2 causes TAUBTTOUTm of the slave channels to be set or reset.
Depending on the settings of TAUBnTDL.TAUBnTDLm, delay time is added to positive or negative logic side of the signal (i.e. whether TAUBTTOUTm is set or reset immediately or after dead time has elapsed). The duration of the dead time is specified by slave channel 3.

Prerequisites

- Three channels. Select an even channel CH (a) and an odd channel CH (a + 1) for slave channel 2 and 3 respectively.
- The operation mode of the master channel must be set to interval timer mode, see **Table 29.143, Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time**
- Slave channel 1 is not used for this function. This ensures that slave channel 2 is an even channel (a), and slave channel 3 is an odd channel (a + 1).
  Slave channel 1 can be used as a separate timer (independent function).
- The operation mode of slave channel 2 must be set to up down mode, see **Table 29.147, Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Triangle PWM Output Function with Dead Time**
  Furthermore, slave channel 2 must be an even channel
- The operation mode of slave channel 3 must be set to one-count mode, see **Table 29.151, Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time**
  Furthermore, slave channel 3 must be an odd channel
- The channel output mode of the master channel must be set to independent channel output mode 1
- The channel output mode of the slave channels 2 and 3 must be set to synchronous channel output mode 2
- The following settings establish TAUBTTOUTm at high level for the down status of the carrier cycle.
  - If the TAUBnCMORm.MD0 (master) bit is set to 0,TAUBnTO.TAUBnTOm must be set to 1 while TAUBnTOE.TAUBnTOEm is 0. (recommended)
  - If the TAUBnCMORm.MD0 (master) bit is set to 1, TAUBnTO.TAUBnTOm must be set to 0 while TAUBnTOE.TAUBnTOEm is 0.

NOTE

Slave channel 1 is not used for Triangle PWM Output Function with Dead Time.
Slave channel 1 can be used as a separate timer (independent function).

Functional description

The counters are started by setting the channel trigger bits (TAUBnTS.TAUBnTSm) to 1. This in turn sets TAUBnTE.TAUBnTEm, enabling count operation. The current values of TAUBnCDRm is written to TAUBnCNTm and the counters start to count down from these values. If the master channel TAUBnCMORm.TAUBnMD0 bit is set to 1, an interrupt is generated and TAUBTTOUTm signal of the master toggles.

- Master channel:
  When the counter of the master channel reaches 0000_H, INTTAUBnIm is generated and the TAUBTTOUTm signal toggles. The counter reloads the TAUBnCDRm value and counts down.
- Slave channel 2:
  The INTTAUBnIm of the master channel triggers the counter of the slave channel 2:
  - If the slave counter currently counts down, it changes count direction.
  - If the slave counter currently counts up, the value of TAUBnCDRm is reloaded and the counter counts down.

  The counter continues to count down or up and awaits the next INTTAUBnIm of the master channel.

  When the counter value of slave channel 2 reaches 0001_H, INTTAUBnIm is generated.
- Slave channel 3:
  INTTAUBnIm of slave channel 2 triggers the counter of slave channel 3. The current value of TAUBnCDRm (slave 3) is written to TAUBnCNTm (slave 3) and the counter starts to count down from this TAUBnCDRm value.
  When the counter reaches 0000_H, INTTAUBnIm is generated. The counter returns to $FFFF_H$ and awaits the next INTTAUBnIm of slave channel 2.

The TAUBnTDL.TAUBnTDLm settings of the corresponding channel specify whether it is set/reset immediately, or after dead time has elapsed, as shown in **Table 29.142, Behavior of TAUBTTOUTm when an Interrupt Occurs on Slave Channel 2**.

The TAUBnTOL.TOLm settings specify whether set corresponds to a high signal (TAUBnTOL.TAUBnTOLm = 0) or a low signal (TAUBnTOL.TAUBnTOLm = 1).

The counter can be stopped by setting TAUBnTT.TAUBnTTm to 1 for the master and slave channel(s), which in turn sets TAUBnTE.TAUBnTEm to 0. TAUBnCNTm and TAUBTTOUTm of master and slave channel(s) stop but retain their values.

TAUBnCDRm value of slave channel 2 can be set to 0000_H to output 100% TAUBTTOUTm.

NOTE

If a forced restart is executed during operation, TAUBTTOUTm is not output as a triangle PWM signal.

Conditions

Simultaneous rewrite can be used with this function. Please see Section 29.6, Simultaneous Rewrite.

TAUBnTOL.TAUBnTOLm and TAUBnTDL.TAUBnTDLm bits should be set before the counter starts, and slave channels 2 and 3 should have opposite TAUBnTOL.TAUBnTOLm settings or opposite TAUBnTDL.TAUBnTDLm settings.

Table 29.142 Behavior of TAUBTTOUTm when an Interrupt Occurs on Slave Channel 2

TAUBnTDL.TAUBnTDLm	Count Direction of Slave Channel 2 when Interrupt is Generated	TAUBTTOUTm Set/Reset Timing
0	Down	Set after dead time has elapsed
	Up	Reset immediately
1	Down	Set immediately
	Up	Reset after dead time has elapsed

29.14.6.2 Equations

Pulse cycle = (TAUBnCDRm (master) + 1) × count clock cycle

$0000_H \leq$ TAUBnCDRm (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUBnCDRm (master) + 1) × 2 × count clock cycle

PWM signal width (positive phase) = [(TAUBnCDRm (master) + 1 – TAUBnCDRm (slave 2)) × 2 – (TAUBnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (negative phase) = [(TAUBnCDRm (master) + 1 – TAUBnCDRm (slave 2)) × 2 + (TAUBnCDRm (slave 3) + 1)] × count clock cycle

29.14.6.3 Block Diagram and General Timing Diagram

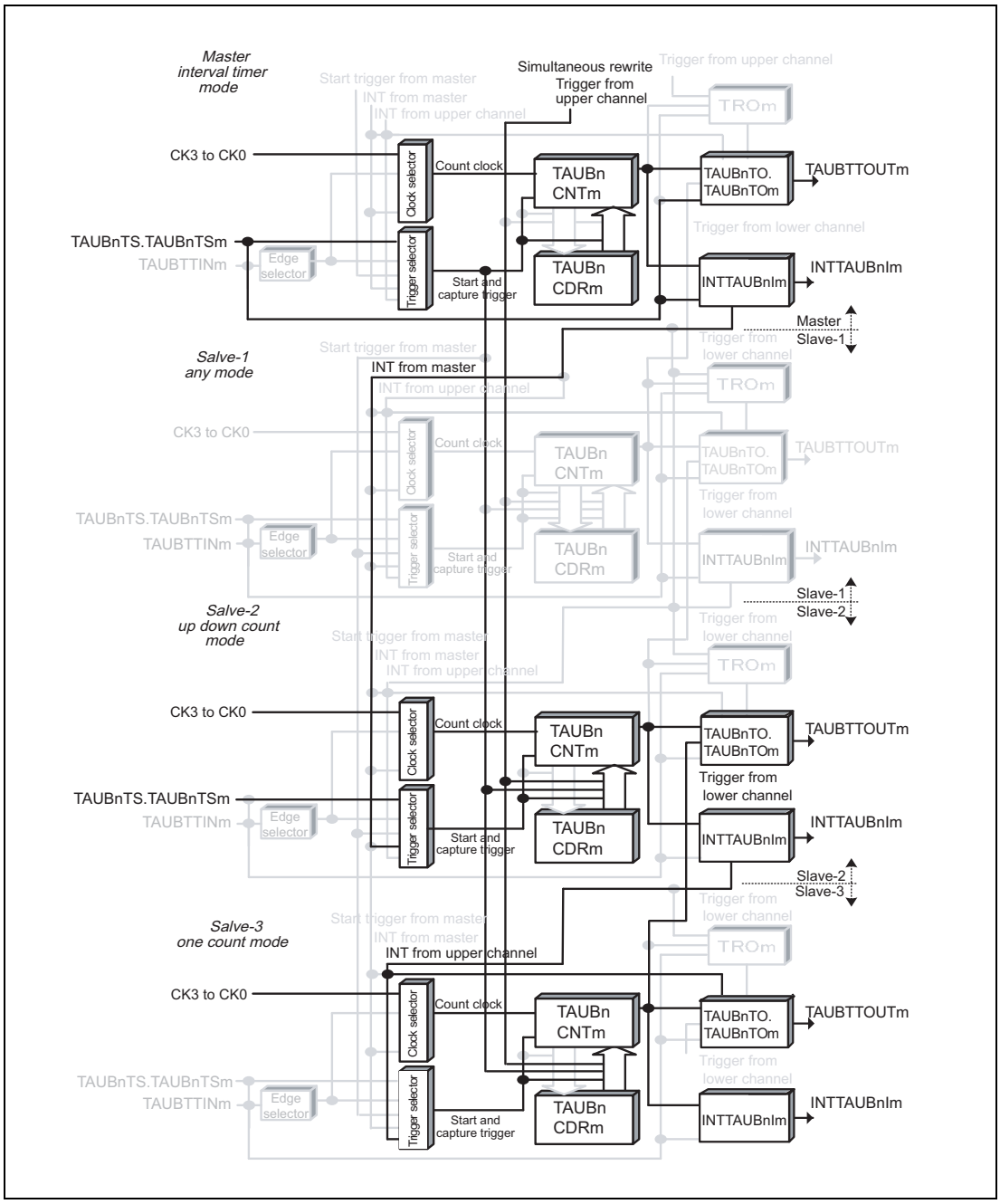


Figure 29.102 Block Diagram for Triangle PWM Output Function with Dead Time

The following settings apply to the general timing diagram.

- Master channel:
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)
- Slave channel 2:

– INTTAUBnIm is not generated at operation start (TAUBnCMORm.TAUBnMD0 = 0)
– TAUBnTDL.TAUBnTDLm = 0
– Positive logic (TAUBnTOL.TAUBnTOLm = 0)

- Slave channel 3:
  – Enables start trigger detection during counting(TAUBnCMORm.TAUBnMD0 = 1)
  – TAUBnTDL.TAUBnTDLm = 1
  – Positive logic (TAUBnTOL.TAUBnTOLm = 0)

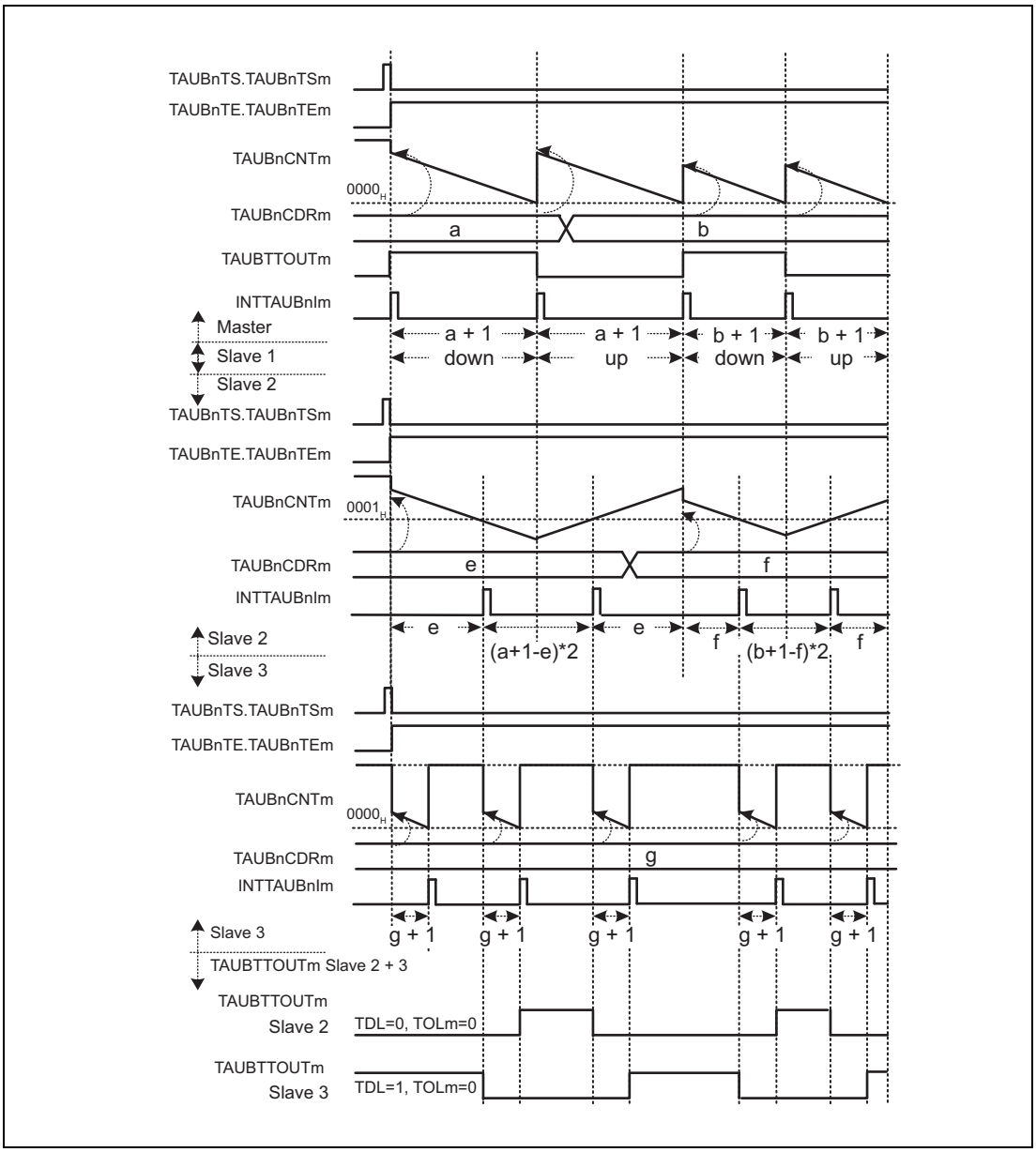


Figure 29.103 General Timing Diagram for Triangle PWM Output Function with Dead Time

29.14.6.4 Register Settings for the Master Channel

(1) TAUBnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS [1:0]		—	TAUBn CCS0	TAUBn MAS	TAUBnSTS[2:0]			TAUBnCOS [1:0]		—	TAUBnMD[4:1]				TAUBn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.143 Contents of the TAUBnCMORm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 1_B.
10 to 8	TAUBnSTS[2:0]	Write 000_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0000_B.
0	TAUBnMD0	0: INTTAUBnIm not generated and TAUBTTOUTm does not toggle at operation start 1: Generates INTTAUBnIm and toggles TAUBTTOUTm at operation start

(2) TAUBnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.144 Contents of the TAUBnCMURm Register for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

Table 29.145 Control Bit Settings for Independent Channel Output Mode 1

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 0_B.
TAUBnTOC.TAUBnTOCm	Write 0_B.
TAUBnTOL.TAUBnTOLm	Write 0_B.
TAUBnTDE.TAUBnTDEm	Write 0_B.
TAUBnTDL.TAUBnTDLm	Write 0_B.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 29.146 Simultaneous Rewrite Settings for the Master Channel of the Triangle PWM Output Function with Dead Time

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

NOTE

If TAUBnRDS.TAUBnRDSm = 1, the master channel requires an upper channel that generates the simultaneous rewrite trigger signal.

29.14.6.5 Register Settings for Slave Channel 2

(1) TAUBnCMORm for slave channel 2

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.147 Contents of the TAUBnCMORm Register for the Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 111_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 1001_B.
0	TAUBnMD0	Write 0_B.

(2) TAUBnCMURm for slave channel 2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.148 Contents of the TAUBnCMURm Register for the Slave Channel 2 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 2

Table 29.149 Control Bit Settings for Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 1_B.
TAUBnTDL.TAUBnTDLm	0: Dead time is added to the positive phase 1: Dead time is added to the negative phase

CAUTION

Set TAUBnTDL.TAUBnTDLm exclusively to the odd channel.

(4) Simultaneous rewrite for slave channel 2

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 29.150 Simultaneous Rewrite Settings for Slave Channel 2 of the Triangle PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

29.14.6.6 Register Settings for Slave Channel 3

(1) TAUBnCMORm for slave channel 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUBnCKS[1:0]		—	TAUBnCCS0	TAUBnMAS	TAUBnSTS[2:0]			TAUBnCOS[1:0]		—	TAUBnMD[4:1]				TAUBnMD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 29.151 Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
15, 14	TAUBnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3 The value of the TAUBnCKS[1:0] bits of the master and slave channel(s) must be identical.
13	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
12	TAUBnCCS0	Write 0_B.
11	TAUBnMAS	Write 0_B.
10 to 8	TAUBnSTS[2:0]	Write 110_B.
7, 6	TAUBnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUBnMD[4:1]	Write 0100_B.
0	TAUBnMD0	Write 1_B.

(2) TAUBnCMURm for slave channel 3

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUBnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 29.152 Contents of the TAUBnCMURm Register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUBnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for slave channel 3

Table 29.153 Control Bit Settings for Synchronous Channel Output Mode 2 with Dead Time Output

Bit Name	Setting
TAUBnTOE.TAUBnTOEm	Write 1_B.
TAUBnTOM.TAUBnTOMm	Write 1_B.
TAUBnTOC.TAUBnTOCm	Write 1_B.
TAUBnTOL.TAUBnTOLm	0: Positive logic 1: Negative logic
TAUBnTDE.TAUBnTDEm	Write 1_B.
TAUBnTDL.TAUBnTDLm	0: Dead time is added to the positive phase 1: Dead time is added to the negative phase

CAUTION

Set TAUBnTDL.TAUBnTDLm exclusively to the even channel.

(4) Simultaneous rewrite for slave channel 3

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 29.154 Simultaneous Rewrite Settings for Slave Channel 3 of the Triangle PWM Output Function

Bit Name	Setting
TAUBnRDE.TAUBnRDEm	1: Enables simultaneous rewrite
TAUBnRDS.TAUBnRDSm	0: Selects master channel for simultaneous rewrite triggers. 1: Selects upper channel outside the channel group for simultaneous rewrite triggers.
TAUBnRDM.TAUBnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUBnRDC.TAUBnRDCm	0: Does not use the channel to generate the simultaneous rewrite trigger.

29.14.6.7 Operating Procedure for Triangle PWM Output Function with Dead Time

Table 29.155 Operating Procedure for Triangle PWM Output Function with Dead Time

	Operation	Status of TAUBn
Initial channel setting	Master channel: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.6.4, Register Settings for the Master Channel** Slave channel 2: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.6.5, Register Settings for Slave Channel 2** Slave channel 3: set the TAUBnCMORm and TAUBnCMURm registers and the channel output mode as described in **Section 29.14.6.6, Register Settings for Slave Channel 3** Set the values of the TAUBnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUBnTS.TAUBnTSm of the master and slave channels to 1 simultaneously. TAUBnTS.TAUBnTSm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUBnIm (master) is generated when TAUBnCMORm.TAUBnMD0 is set to 1 on the master channel. TAUBnCMORm.TAUBnMD0 is set to 1.
During operation	TAUBnCDRm can be changed at any time. TAUBnCNTm and TAUBnRSF.TAUBnRSFm can be read at any time. TAUBnRDT.TAUBnRDTm can be changed during operation.	TAUBnCNTm of the master channel and slave channel 2 load TAUBnCDRm and count down. When the counter of the master channel reaches 0000_H: • INTTAUBnIm (master) is generated • TAUBnCNTm (master) reloads the TAUBnCDRm value and continues count operation • TAUBnCNTm (slave 2) reloads the TAUBnCDRm value or counts in the reverse direction When TAUBnCNTm (slave 2) reaches 0001_H: • INTTAUBnIm (slave 2) is generated • TAUBnCNTm of slave channel 3 loads the TAUBnCDRm value and counts down When TAUBnCNTm of slave channel 3 = 0000_H: • INTTAUBnIm is generated
Stop operation	Set TAUBnTT.TAUBnTTm of the master and slave channels to 1 simultaneously. TAUBnTT.TAUBnTTm is a trigger bit, so it is automatically cleared to 0.	TAUBnTE.TAUBnTEm is cleared to 0 and the counter stops. TAUBnCNTm and TAUBTTOUTm stop and retain their current values.

Restart operation

29.14.6.8 Specific Timing Diagrams

(1) Duty cycle = 0%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

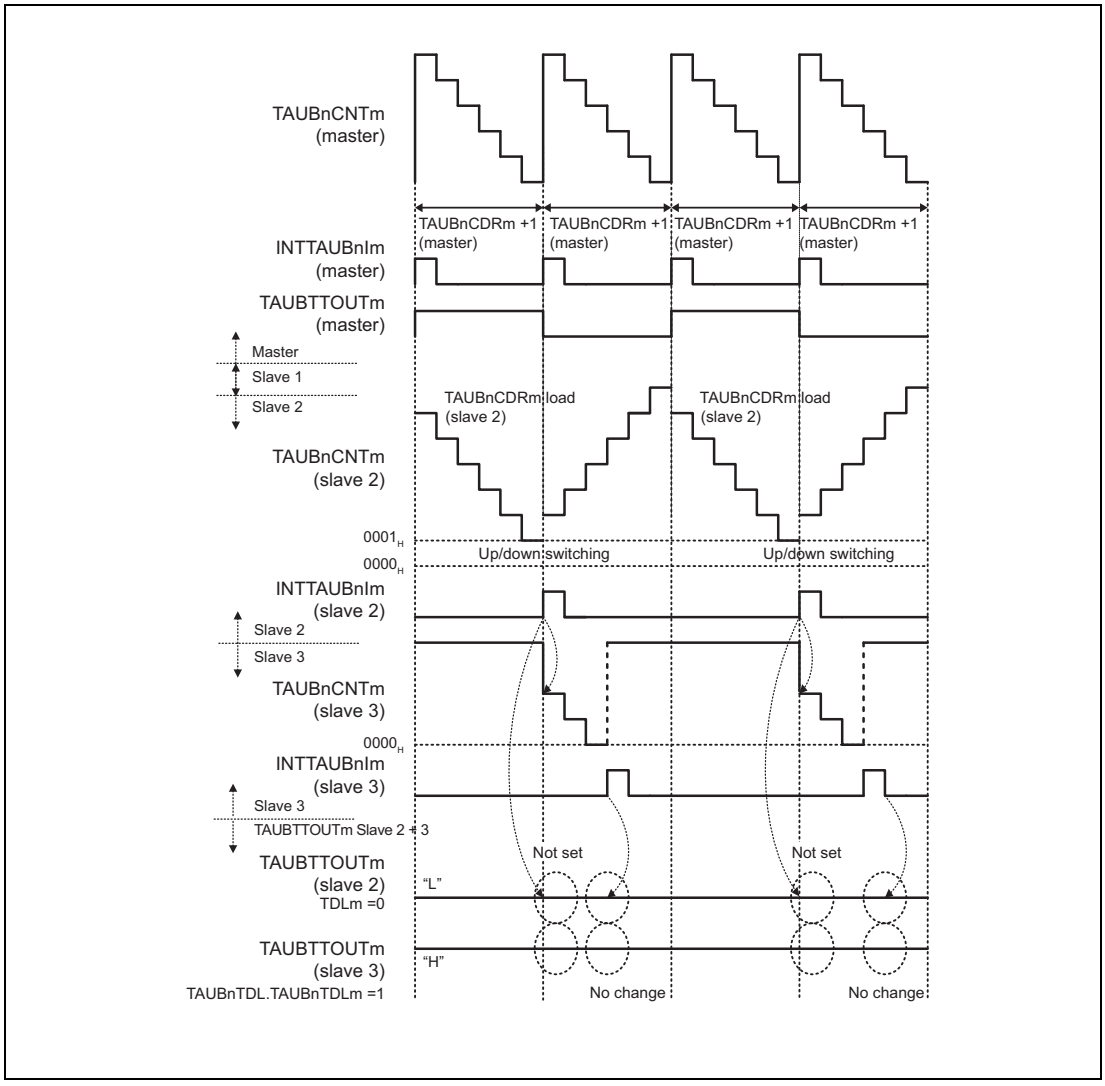


Figure 29.104 TAUBnCDRm (slave 2) ≥ TAUBnCDRm (master) + 1

- If TAUBnCDRm (slave 2) ≥ TAUBnCDRm (master), the counter of slave channel cannot reach 0000_H during counting down. Therefore TAUBTTOUTm cannot toggle, i.e. it remains at its initial state. The interrupt from slave channel 2 occurs during count up, therefore it is a reset signal.

(2) Duty cycle = 100%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

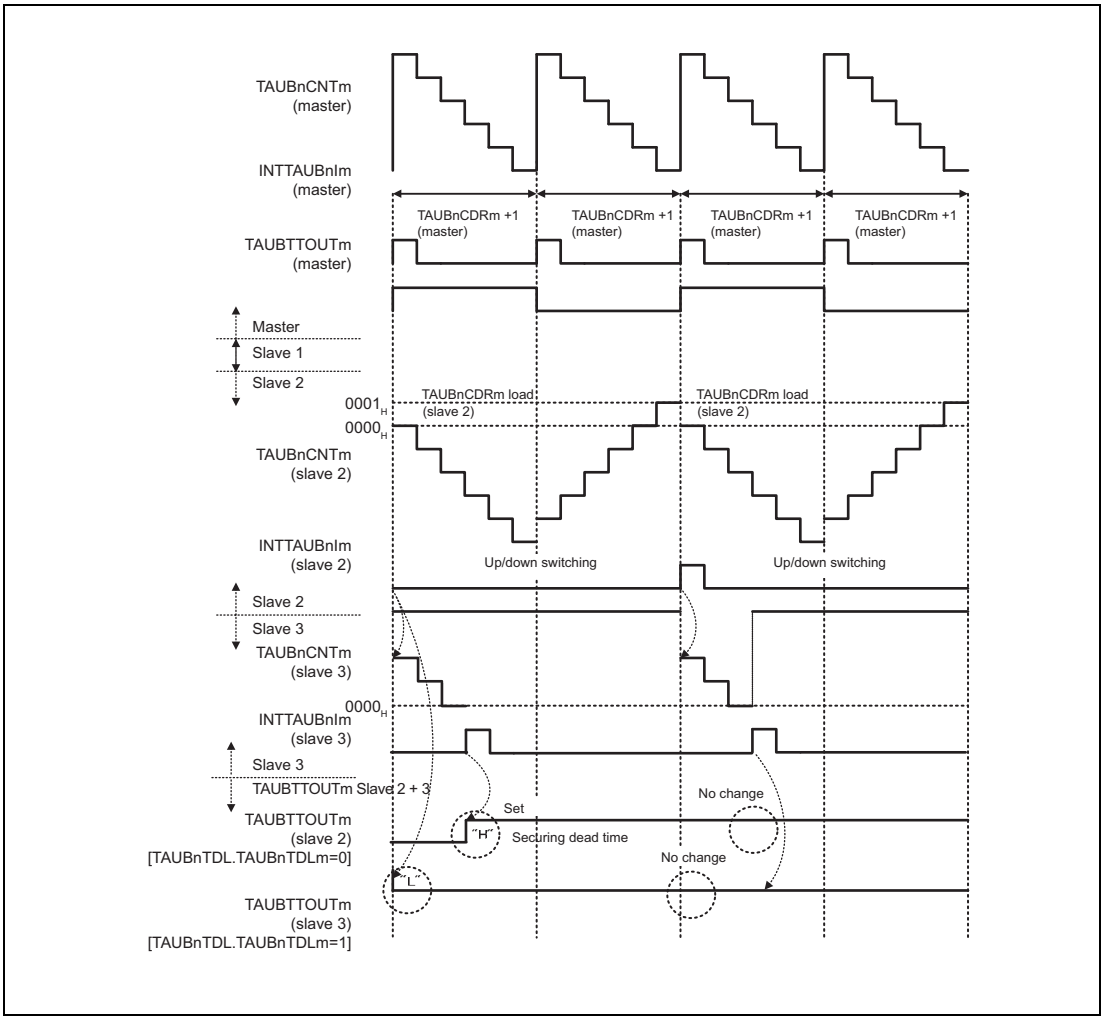


Figure 29.105 TAUBnCDRm (slave 2) = 0000_H

- If TAUBnCDRm (slave 2) = 0000_H the counter of slave channel cannot reach 0001_H while counting up and therefore cannot generate an INTTAUBnIm while counting up.
  - The set conditions for a channel in which TAUBnTDL.TAUBnTDLm = 0 are met after dead time has elapsed. TAUBTTOUTm toggles but remains in the new state because the reset conditions never occur for such a channel.
  - Slave channel 3 in the diagram above is set when the counter starts.
    However, the reset conditions for a channel in which TAUBnTDL.TAUBnTDLm = 1 never occur so TAUBTTOUTm remains in its initial state for such a slave channel.

(3) TAUBTTOUTm (slave 2) = 0% and TAUBTTOUTm (slave 3) ≥ 0%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

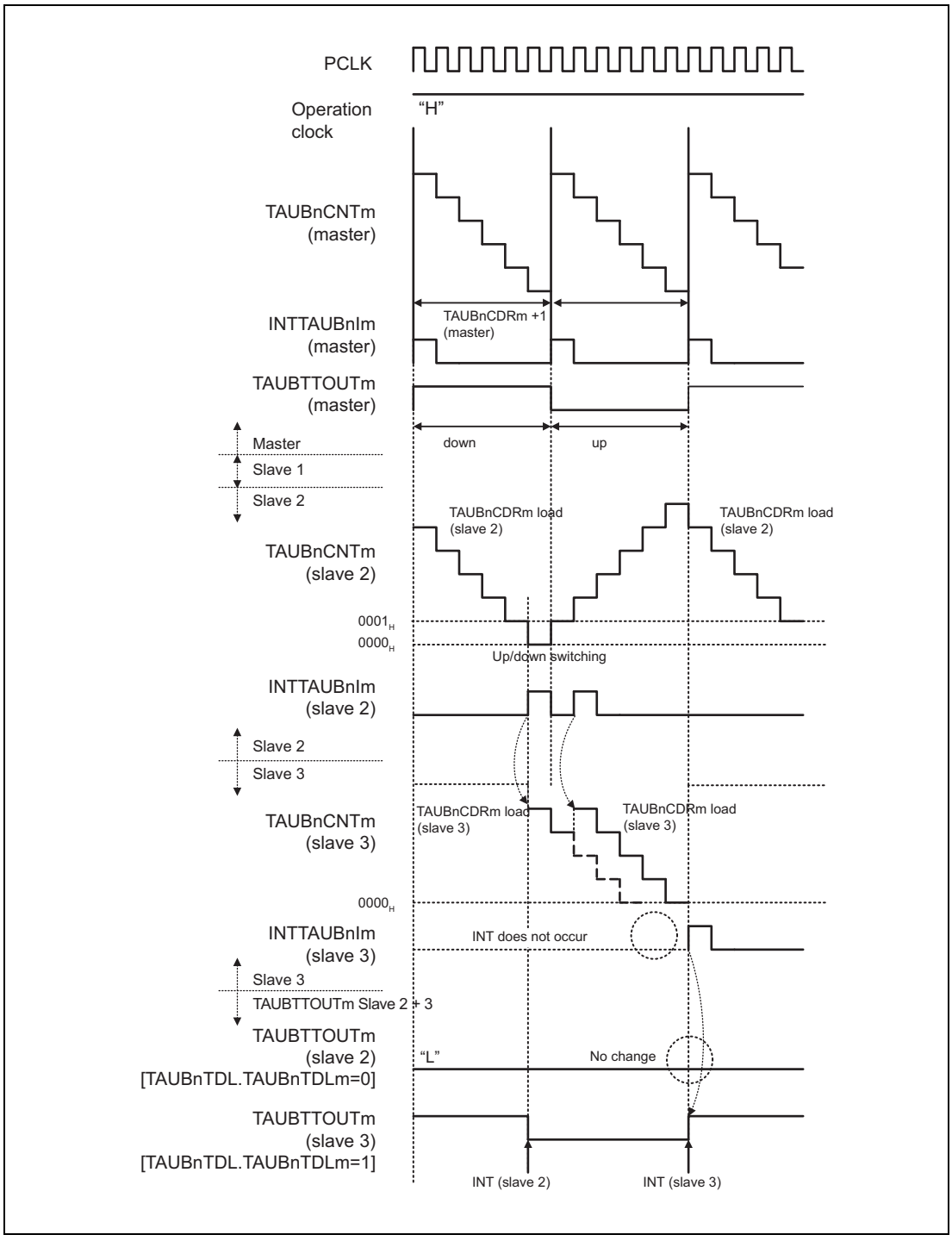


Figure 29.106 TAUBnCDRm (master) = 0005_H, TAUBnCDRm (slave 2) = 0005_H TAUBnCDRm (slave 3) = 0004_H

- When the counter of slave channel 2 reaches 0000_H after detecting that the counter reached 0001_H, INTTAUBnIm (slave 2) is generated. The counter of slave channel 3 starts to count down.
- If another INTTAUBnIm (slave 2) is generated while the counter of slave channel 3 is still counting down, the value of TAUBnCDRm (slave 3) is reloaded and the counter restarts counting down from this value.
- In the diagram above, the first interrupt on channel 2 occurs while the counter is counting down, and the second while it is counting up.
- After the first interrupt, a slave for which TAUBnTDL.TAUBnTDLm = 0 waits for dead time to elapse before setting. However, if another interrupt occurs on slave 2 before the dead time has elapsed, the counter is counting up, so the signal acts as a reset signal, meaning that a channel for which TAUBnTDL.TAUBnTDLm = 0 always remains inactive.
- TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 1 is set and reset as normal when the corresponding INTTAUBnIm is generated.

(4) TAUBTTOUTm (slave 2) > 0% and TAUBTTOUTm (slave 3) = 100%

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

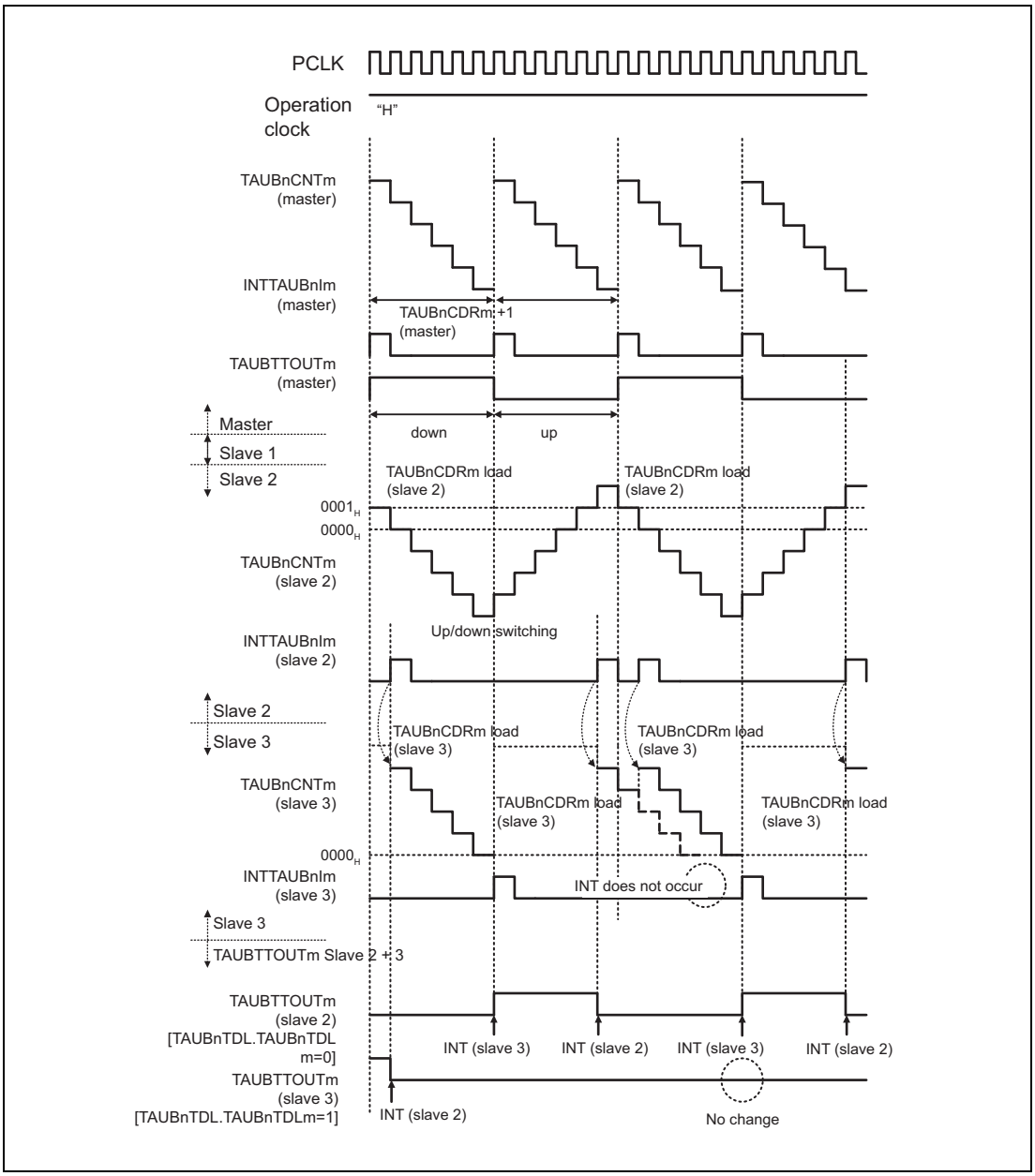


Figure 29.107 TAUBnCDRm (master) = 0005_H, TAUBnCDRm (slave 2) = 0001_H TAUBnCDRm (slave 3) = 0004_H PWM Signal Width (negative phase) ≥ Carrier Cycle

- After the second interrupt on slave channel 2, a slave for which TAUBnTDL.TAUBnTDLm = 1 is reset after the dead time has elapsed. However if another interrupt occurs on slave 2 before the

dead time has elapsed, slave 3 is restarted, and then if an interrupt on slave channel 3 is generated, the counter is counting up, so the signal acts as a setting signal, meaning that a channel for which TAUBnTDL.TAUBnTDLm = 1 always remains active.

- TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 0 is set and reset as normal when the corresponding INTTAUBnIm is generated.

(5) Inhibited INTTAUBnIm to set TAUBTTOUTm positive phase period

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUBnTOL.TAUBnTOLm = 1)

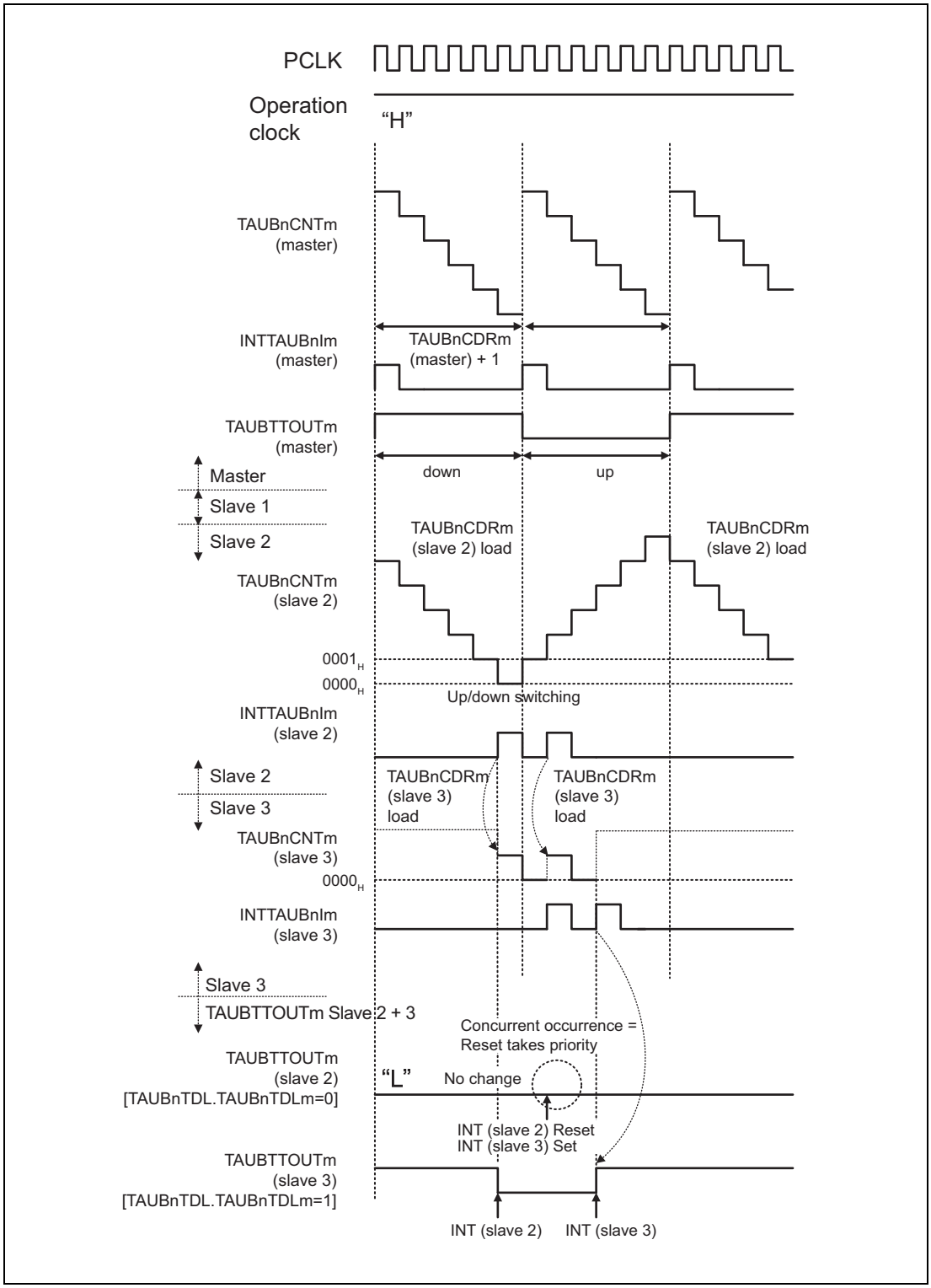


Figure 29.108 TAUBnCDRm (master) = 0005_H, TAUBnCDRm (slave 2) = 0005_H, TAUBnCDRm (slave 3) = 0001_H PWM Signal Width (positive phase) = 0

- The counter of slave channel 3 reaches 0000_H and generates an INTTAUBnIm to set the TAUBTTOUTm of slave channel for which TAUBnTDL.TAUBnTDLm = 0 (slave channel 2 in this example).
- If channel 2 generates an INTTAUBnIm and resets TAUBTTOUTm simultaneously, the reset

signal has priority (assuming TAUBnTOL.TAUBnTOLm = 0, otherwise the set signal has priority).

- Therefore, TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 0 remains in the value after reset.

(6) Inhibited INTTAUBnIm to set TAUBTTOUTm negative phase period

The following settings apply to the diagram below.

- Slave channel 2:
  - Positive logic (TAUBnTOL.TAUBnTOLm = 0)
- Slave channel 3:

– Negative logic (TAUBnTOL.TAUBnTOLm = 1)

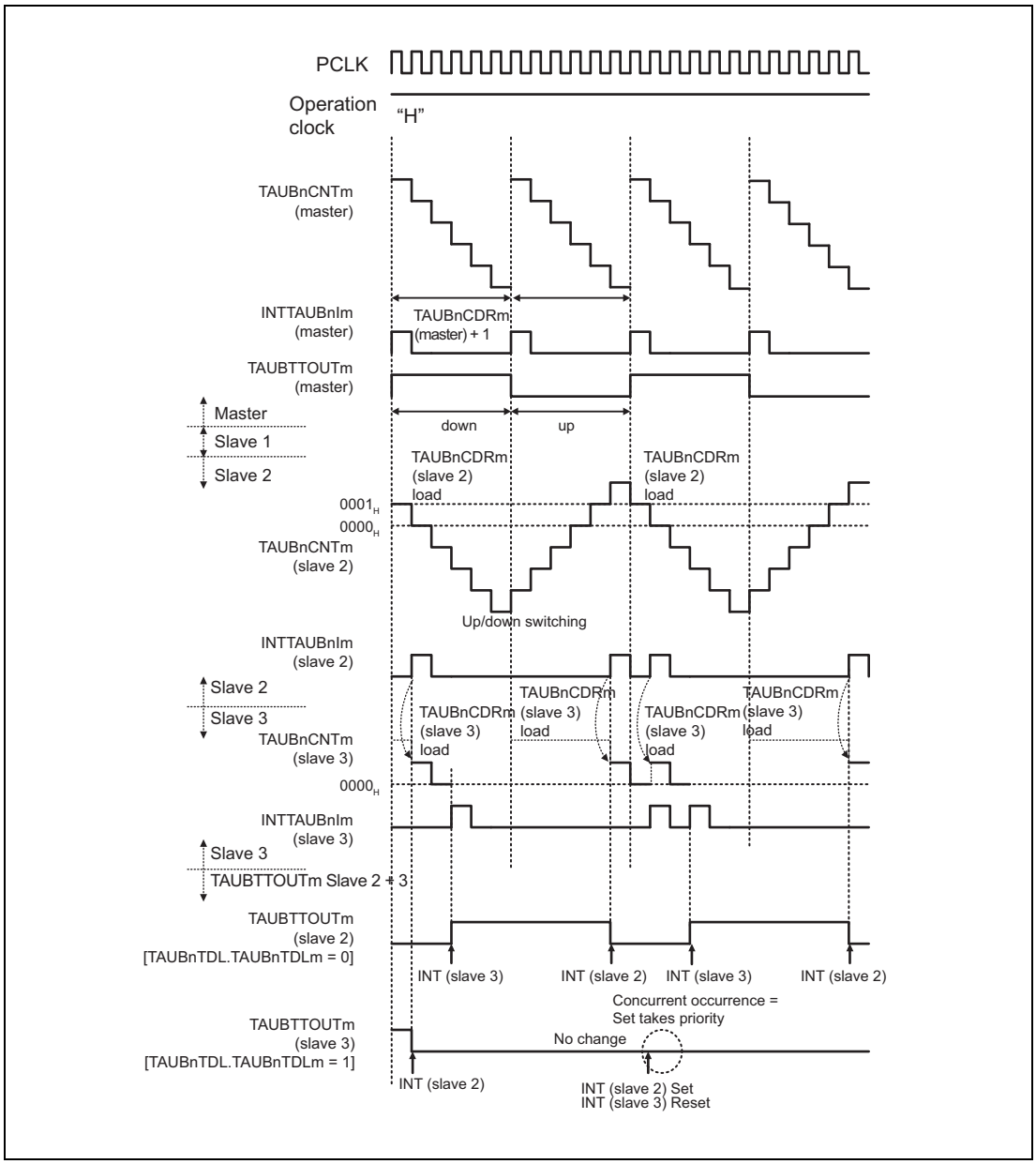


Figure 29.109 TAUBnCDRm (master) = 0005_H, TAUBnCDRm (slave 2) = 0001_H, TAUBnCDRm (slave 3) = 0001_H PWM Signal Width (negative phase) = Carrier Cycle

- The counter of slave channel 3 reaches 0000_H and generates an INTTAUBnIm to set the TAUBTTOUTm of slave channel for which TAUBnTDL.TAUBnTDLm = 1 (slave 3 in this example).
- If slave channel 2 generates an INTTAUBnIm and resets TAUBTTOUTm simultaneously, the set signal is given priority if TAUBnTOL.TAUBnTOLm = 1 (if TAUBnTOL.TAUBnTOLm = 0, the reset signal is given priority).
- Therefore, TAUBTTOUTm of a slave channel for which TAUBnTDL.TAUBnTDLm = 1 remain in the value after reset.

(7) Slave 2 TAUBnCDRm = 0000_H (Duty cycle = 100%)

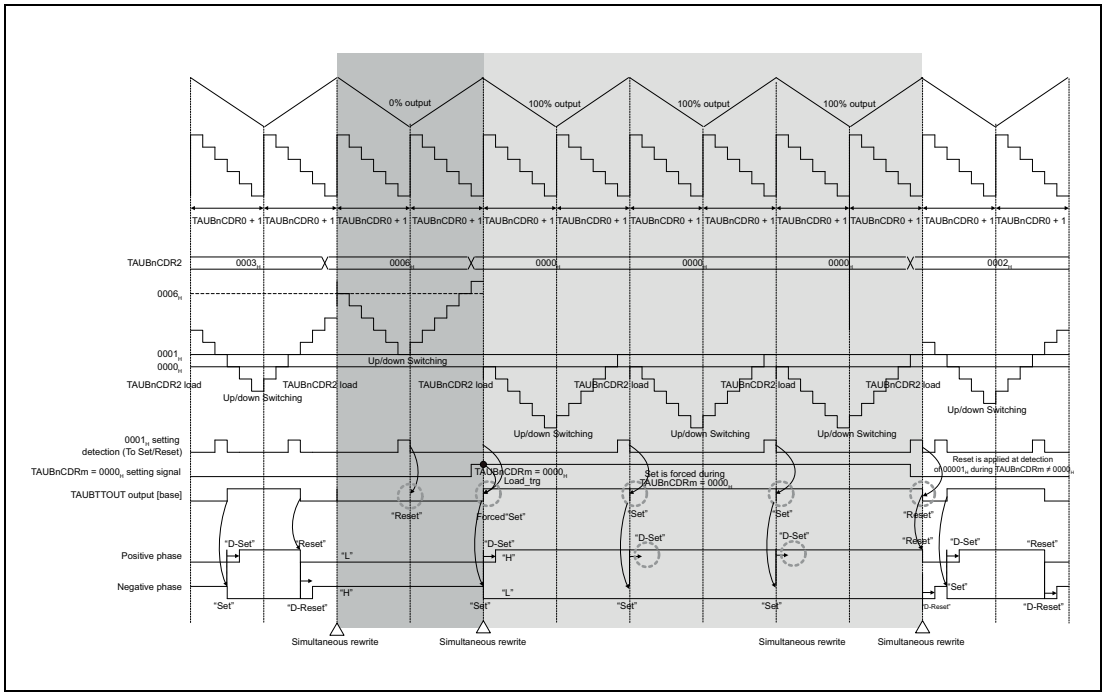


Figure 29.110 Slave 2 TAUBnCDRm = 0000_H (Duty cycle = 100%)

When rewriting (slave channel 2) TAUBnCDRm ≠ 0000_H to (slave channel 2) TAUBnCDRm = 0000_H (100% output), set the negative phase side at the start of the carrier cycle, and set the positive phase side after dead time is secured.

When rewriting (slave channel 2) TAUBnCDRm = 0000_H (100% output) to (slave channel 2) TAUBnCDRm ≠ 0000_H, reset the positive phase side at the end of the carrier cycle, and reset the negative phase side after dead time is secured.

29.14.7 AD Conversion Trigger Output Function Type 2

29.14.7.1 Overview

Summary

This function is identical to Section 29.14.5, Triangle PWM Output Function, except that TAUBTTOUTm is not output.

This is enabled by setting the channel output mode of the slave to independent channel output mode controlled by software.

29.14.7.2 Block Diagram and General Timing Diagram

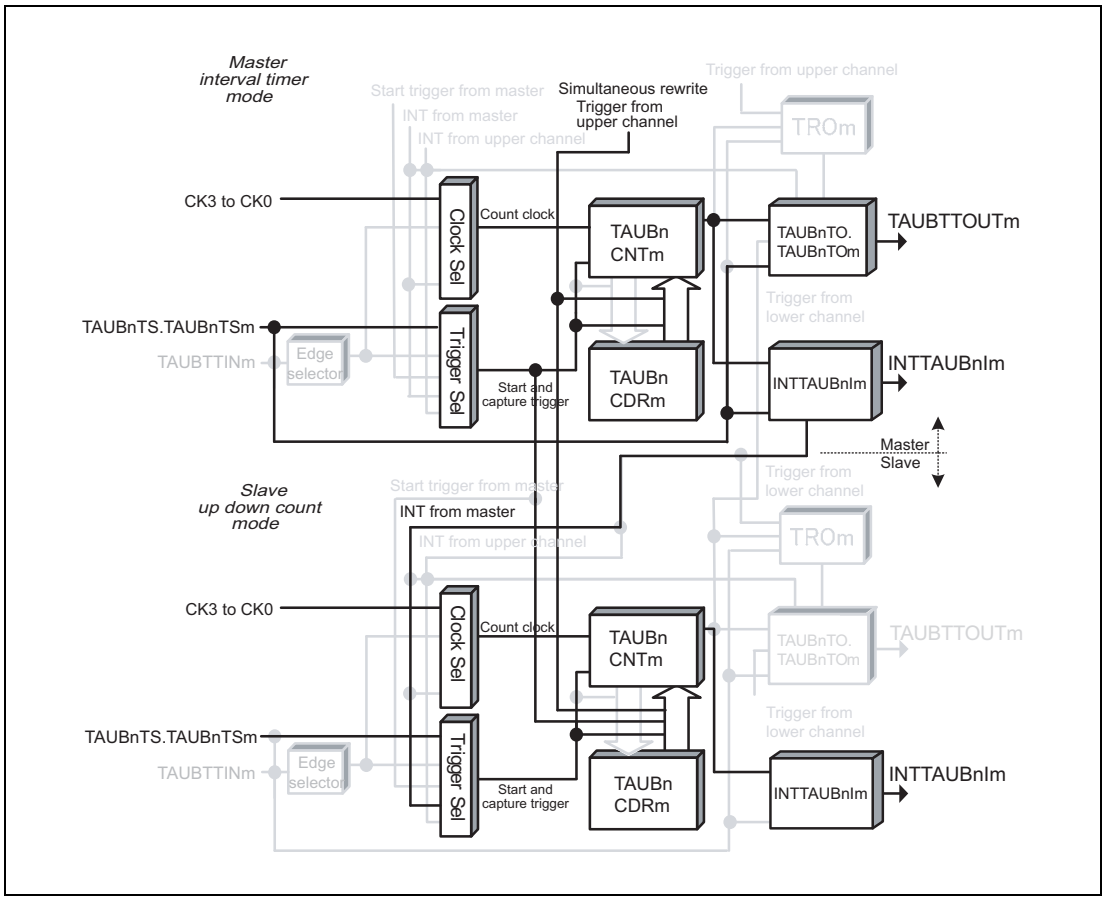


Figure 29.111 Block Diagram for AD Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram.

- Master channel
  - INTTAUBnIm is generated at operation start (TAUBnCMORm.TAUBnMD0 = 1)

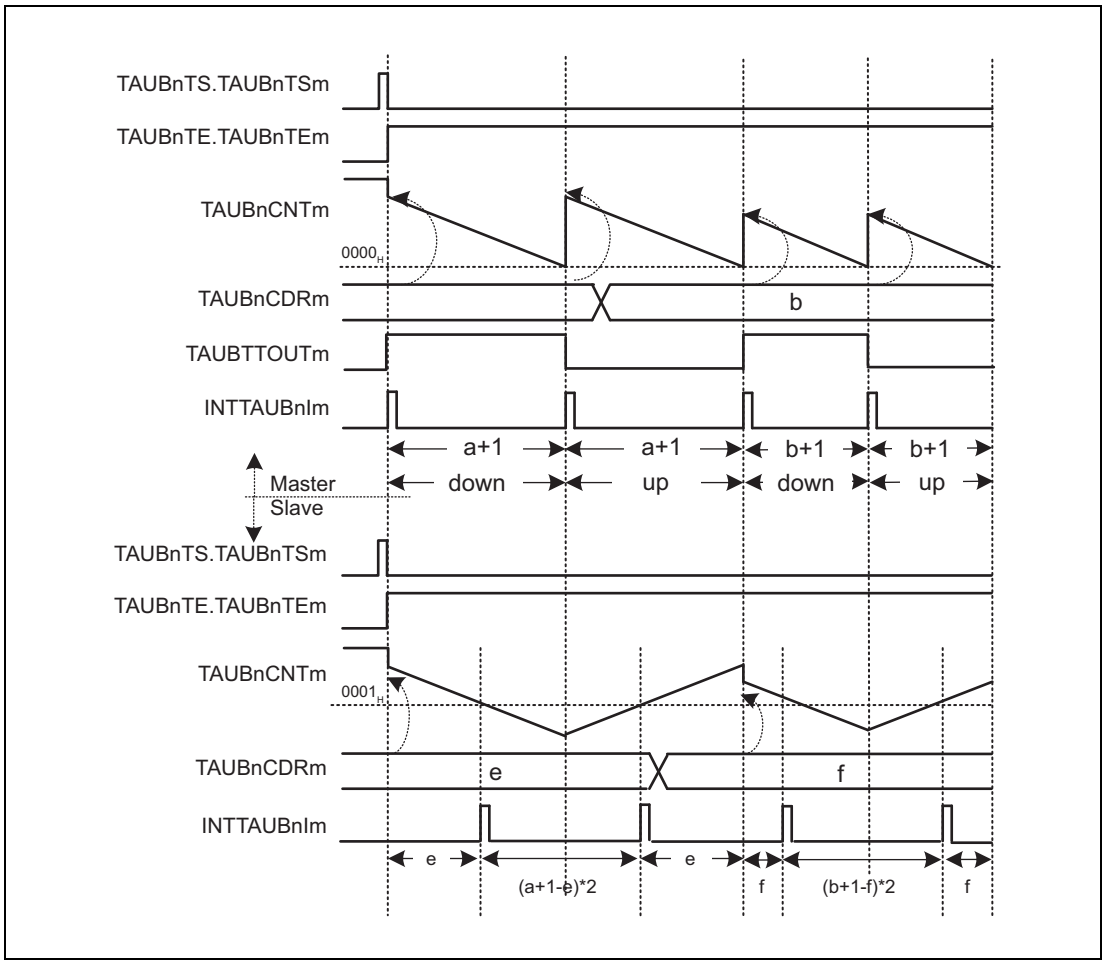


Figure 29.112 General Timing Diagram for AD Conversion Trigger Output Function Type 2

Section 30 Timer Array Unit J (TAUJ)

This section contains a generic description of the timer array unit J (TAUJ).

The first part of this section describes all RH850/D1L/D1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the TAUJ.

30.1 Features of RH850/D1L/D1M TAUJ

30.1.1 Number of Units

This microcontroller has the following number of TAUJ units.

Table 30.1 Number of Units

Product Name	All products
Number of Units	1
Name	TAUJn (n = 0)

Table 30.2 Index

Index	Meaning
n	Throughout this section, the individual TAUJ units are identified by the index “n”; for example, TAUJnTOM is the TAUJn channel output mode register.
m	The TAUJ has 4 channels. Throughout this section, the individual channels are identified by the index “m” (m = 0 to 3), thus a certain channel is denoted as CHm. The even numbered channels (m = 0, 2) are denoted as CHm_even. The odd numbered channels (m = 1, 3) are denoted as CHm_odd.

30.1.2 Register Base Address

TAUJn base address is listed in the following table.

TAUJn register addresses are given as offsets from the base addresses.

Table 30.3 Register Base Address

Base Address Name	Base Address
<TAUJ0_base>	FFE5 0000_H

30.1.3 Clock Supply

The TAUJn clock supply is shown in the following table.

Table 30.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
TAUJ0	PCLK	C_ISO_TAUJ

30.1.4 Interrupt Requests

TAUJn interrupt requests are listed in the following table.

Table 30.5 Interrupt Requests

TAUJn signals	Function	Connected to
TAUJ0		
INTTAUJ0I0	Channel 0 interrupt	Interrupt Controller INTTAUJ0I0 DMA Controller trigger ID 83
INTTAUJ0I1	Channel 1 interrupt	Interrupt Controller INTTAUJ0I1 DMA Controller trigger ID 84
INTTAUJ0I2	Channel 2 interrupt	Interrupt Controller INTTAUJ0I2 DMA Controller trigger ID 85
INTTAUJ0I3	Channel 3 interrupt	Interrupt Controller INTTAUJ0I3 DMA Controller trigger ID 86

30.1.5 Reset Sources

TAUJn reset sources are listed in the following table. TAUJn is initialized by these reset sources.

Table 30.6 Reset Sources

Unit Name	Reset Source
TAUJ0	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

30.1.6 External Input/Output Signals

External input/output signals of TAUJn are listed below.

Table 30.7 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
TAUJ0		
TAUJTTIN0 to TAUJTTIN3	Channel 0 to 3 input	TAUJ0I0 to TAUJ0I3*1
TAUJTTOUT0 to TAUJTTOUT3	Channel 0 to 3 output	TAUJ0O0 to TAUJ0O3

Note 1. These input signals are passed through a noise filter, refer to the section “Port Filters” in the section “Port Functions”.

30.2 Overview

30.2.1 Functional Overview

The TAUJ has the following functions:

- Independent channel operation function (operated using a single channel)
- Synchronous channel operation function (operated using a master channel and multiple slave channels)

The TAUJ is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 4 channels, each equipped with a 32-bit counter TAUJnCNTm and a 32-bit data register TAUJnCDRm to hold the count start value or compare value.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in two operating modes, either independently or in combination with other channels (synchronously). When one master channel and one or more slave channels operate in combination, the slave channels depend on the master channel.

When a channel is operated independently, it can be operated independent of all other channels.

The synchronous operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

30.2.2 Terms

In this section, the following terms are used.

Independent channel operation function/synchronous operation channel operation function

TAUJ has 4 channels, and provides an independent channel operation function that individual channels operate independently and a synchronous channel operation function that is implemented by using a combination of channels.

- The independent channel operation function can use any channel independent of all other channels.
- The synchronous channel operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

Channel group

In the synchronous channel operation function, all channels that depend on each other are referred to as a “channel group”.

A channel group has one master channel and one or more slave channels.

Upper/lower channel

Depending on the channel number m, a channel with a smaller channel number or higher channel number can be referred to as “upper” or “lower” channel:

- Upper channel: Channel with a smaller channel number
- Lower channel: Channel with a larger channel number

For instance, as to channel 2, channel 1 is an upper channel and channel 3 is a lower channel. Channel 0 is the highest channel and channel 3 is the lowest channel.

30.2.3 Functional List of Timer Operations

This timer provides the following functions by operating each channel independently or by combining multiple channels.

Table 30.8 Functional List of TAUJ Operations

Operation Function	Example
Independent Channel Operation Functions	**Section 30.12**
Interval Timer Function	**Section 30.12.1**
TAUJTTINm Input Interval Timer Function	**Section 30.12.2**
TAUJTTINm Input Pulse Interval Measurement Function	**Section 30.12.3**
TAUJTTINm Input Signal Width Measurement Function	**Section 30.12.4**
TAUJTTINm Input Position Detection Function	**Section 30.12.5**
TAUJTTINm Input Period Count Detection Function	**Section 30.12.6**
Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)	**Section 30.12.7**
Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)	**Section 30.12.8**
Synchronous Channel Operation Functions	**Section 30.13**
PWM Output Function	**Section 30.13.1**

30.2.4 TAUJ I/O and Interrupt Request Signals

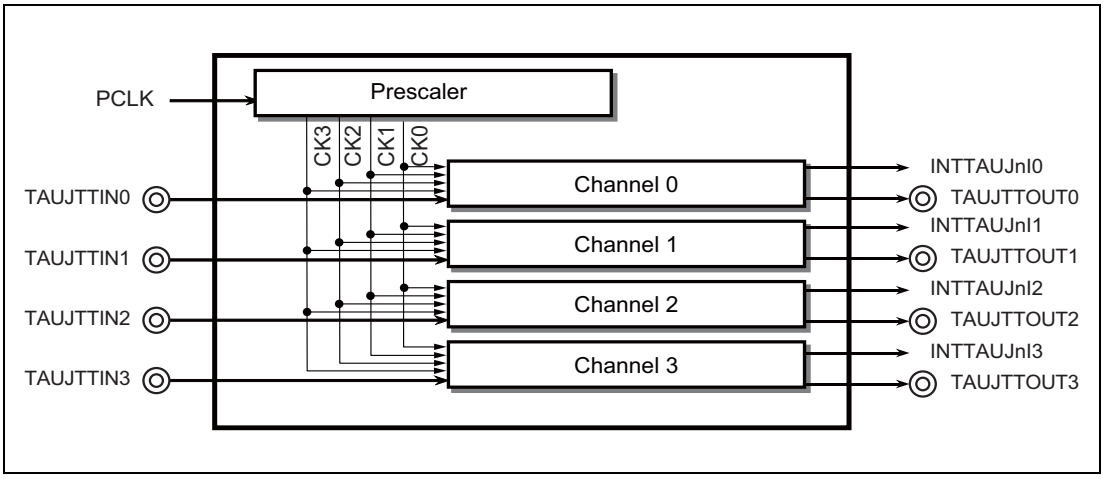


Figure 30.1 TAUJ I/O and Interrupt Request Signals

30.2.5 Block Diagram

The following figure shows the main components of the TAUJ.

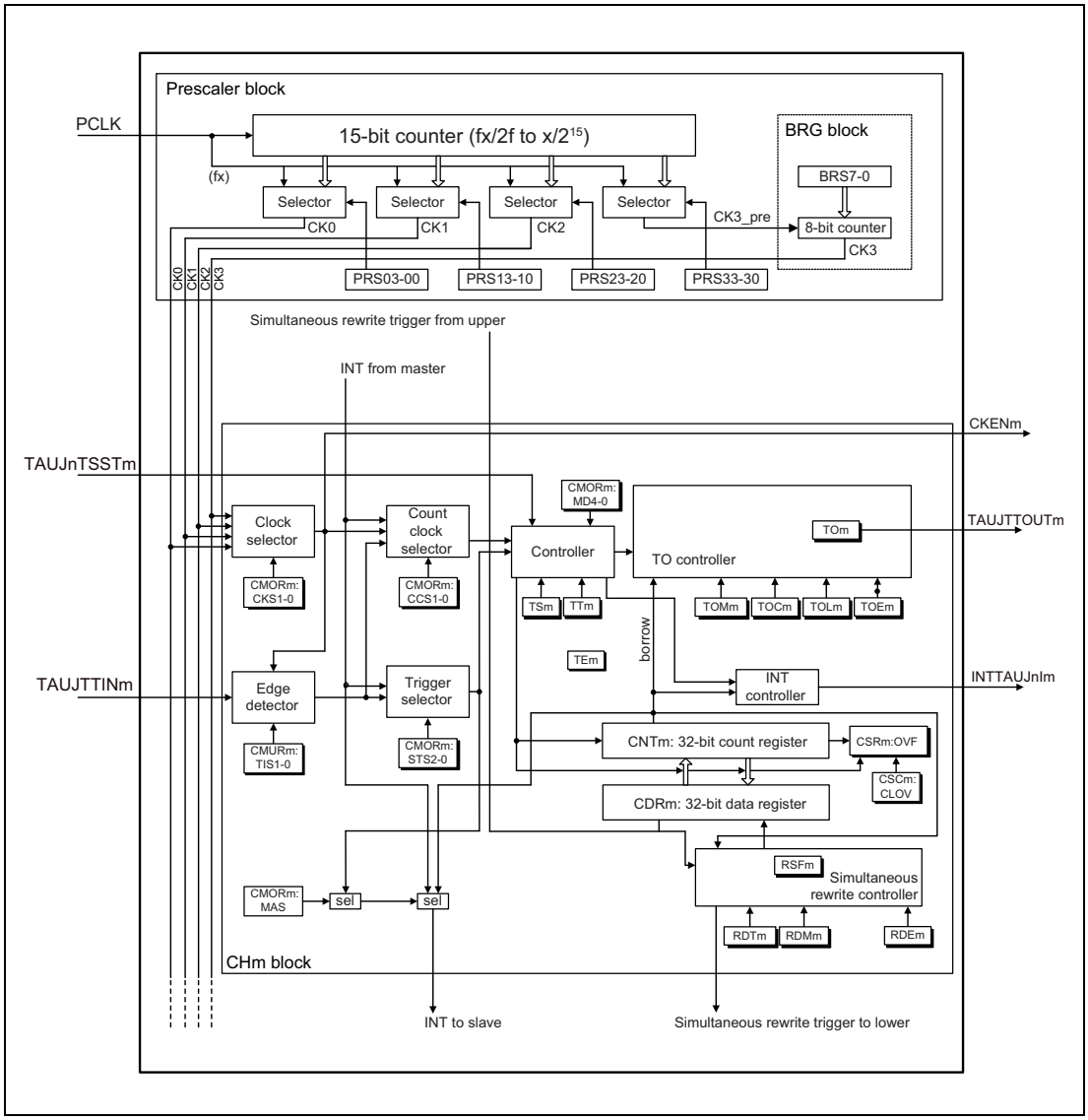


Figure 30.2 Block Diagram of the TAUJ

The prefix “TAUJn” has been omitted from the register names for the sake of clarity in the above figure.

30.2.6 Description of Blocks

The following describes the functional blocks.

Prescaler block

The prescaler block provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

Count clocks CK0 to CK2 are derived from PCLK by a configurable prescaler division factor of 2^0 to 2^{15}. The fourth count clock CK3 can be adjusted more precisely by an additional division factor that is not a power of 2 by using the baud rate generator.

Clock and count selection

For every channel, the count clock selector selects which of the following is used as the clock source.

- One of CK0 to CK3 clocks (selected by the clock selector)

Controller

The controller controls the main operations of the counter.

- Operating mode (selected with the TAUJnCMORm.TAUJnMD[4:0] bits)
- Counter start enable (TAUJnTS.TAUJnTSm) and counter stop (TAUJnTT.TAUJnTTm)

When counter start is enabled, status flag TAUJnTE.TAUJnTEm is set.

Trigger selector

The counter starts automatically when it is enabled (TAUJnTE.TAUJnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger.

- Synchronous channel start trigger input TAUJnTSSTm
- Valid edge of the TAUJTTINm input signal
- INTTAUJnIm from master channel

Simultaneous rewrite controller

Simultaneous rewrite control is enabled in synchronous operating modes. The data registers of all channels in a channel group (TAUJnCDRm) can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUJnTO controller

The output control of every channel enables the generation of various output signals such as PWM signals.

30.3 Registers

30.3.1 List of Registers

TAUJ registers are listed in the following table.

For details about <TAUJn_base>, see Section 30.1.2, Register Base Address.

Table 30.9 List of Registers

Module Name	Register Name	Symbol	Address
TAUJn prescaler registers			
TAUJn	TAUJn prescaler clock select register	TAUJnTPS	<TAUJn_base> + 90_H
TAUJn	TAUJn prescaler baud rate setting register	TAUJnBRS	<TAUJn_base> + 94_H
TAUJn control registers			
TAUJn	TAUJn channel data register m	TAUJnCDRm	<TAUJn_base> + m × 4_H
TAUJn	TAUJn channel counter register m	TAUJnCNTm	<TAUJn_base> + 10_H + m × 4_H
TAUJn	TAUJn channel mode OS register m	TAUJnCMORm	<TAUJn_base> + 80_H + m × 4_H
TAUJn	TAUJn channel mode user register m	TAUJnCMURm	<TAUJn_base> + 20_H + m × 4_H
TAUJn	TAUJn channel status register m	TAUJnCSRm	<TAUJn_base> + 30_H + m × 4_H
TAUJn	TAUJn channel status clear trigger register m	TAUJnCSCm	<TAUJn_base> + 40_H + m × 4_H
TAUJn	TAUJn channel start trigger register	TAUJnTS	<TAUJn_base> + 54_H
TAUJn	TAUJn channel enable status register	TAUJnTE	<TAUJn_base> + 50_H
TAUJn	TAUJn channel stop trigger register	TAUJnTT	<TAUJn_base> + 58_H
TAUJn output registers			
TAUJn	TAUJn channel output enable register	TAUJnTOE	<TAUJn_base> + 60_H
TAUJn	TAUJn channel output register	TAUJnTO	<TAUJn_base> + $5C_H$
TAUJn	TAUJn channel output mode register	TAUJnTOM	<TAUJn_base> + 98_H
TAUJn	TAUJn channel output configuration register	TAUJnTOC	<TAUJn_base> + $9C_H$
TAUJn	TAUJn channel output active level register	TAUJnTOL	<TAUJn_base> + 64_H
TAUJn reload data registers			
TAUJn	TAUJn channel reload data enable register	TAUJnRDE	<TAUJn_base> + $A0_H$
TAUJn	TAUJn channel reload data mode register	TAUJnRDM	<TAUJn_base> + $A4_H$
TAUJn	TAUJn channel reload data trigger register	TAUJnRDT	<TAUJn_base> + 68_H
TAUJn	TAUJn channel reload status register	TAUJnRSF	<TAUJn_base> + $6C_H$
TAUJn emulation register			
TAUJn	TAUJn emulation register	TAUJnEMU	<TAUJn_base> + $A8_H$

30.3.2 Details of TAUJn Prescaler Registers

30.3.2.1 TAUJnTPS — TAUJn Prescaler Clock Select Register

This register specifies clocks CK0, CK1, CK2, and CK3_PRE for all channels of the PCLK prescalers. CK3 is generated by dividing CK3_PRE by the factor specified in TAUJnBRS.

Access: Readable/writable in 16-bit units.

Address: <TAUJn_base> + 90_H

Value after reset: $FFFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnPRS3[3:0]				TAUJnPRS2[3:0]				TAUJnPRS1[3:0]				TAUJnPRS0[3:0]			
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.10 TAUJnTPS Register Contents (1/3)

Bit Position	Bit Name	Function
15 to 12	TAUJnPRS3 [3:0]	Specifies a CK3_PRE clock. The CK3_PRE clock is an input clock of the BRG unit which supplies CK3 operation clocks to all channels.

TAUJnPRS3[3:0]	CK3_PRE clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

The above bits are rewritable only when all the counters using CK3 are stopped (TAUJnTE.TAUJnTEm = 0).

Table 30.10 TAUJnTPS Register Contents (2/3)

Bit Position	Bit Name	Function
11 to 8	TAUJnPRS2 [3:0]	Specifies a CK2 clock.

TAUJnPRS2[3:0]	CK2 clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

The above bits are rewritable only when all the counters using CK2 are stopped (TAUJnTE.TAUJnTEm = 0).

7 to 4	TAUJnPRS1 [3:0]	Specifies a CK1 clock.

TAUJnPRS1[3:0]	CK1 clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

The above bits are rewritable only when all the counters using CK1 are stopped (TAUJnTE.TAUJnTEm = 0).

Table 30.10 TAUJnTPS Register Contents (3/3)

Bit Position	Bit Name	Function
3 to 0	TAUJnPRS0 [3:0]	Specifies a CK0 clock.

TAUJnPRS0[3:0]	CK0 clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

The above bits are rewritable only when all the counters using CK0 are stopped (TAUJnTE.TAUJnTEm = 0).

NOTE

TAUJn clock input PCLK is defined in the first part of this section, **Section 30.1.3, Clock Supply**.

30.3.2.2 TAUJnBRS — TAUJn Prescaler Baud Rate Setting Register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in TAUJnTPS. TAUJnPRS3[3:0].

Access: Readable/writable in 8-bit units.

Address: <TAUJn_base> + 94_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TAUJnBRS[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.11 TAUJnBRS Register Contents

Bit Position	Bit Name	Function
7 to 0	TAUJnBRS [7:0]	Specifies a CK3_PRE clock division factor for generating CK3.

TAUJnBRS[7:0]	CK3 clock
$0000\ 0000_B$	CK3_PRE / 1
$0000\ 0001_B$	CK3_PRE / 2
$0000\ 0010_B$	CK3_PRE / 3
$0000\ 0011_B$	CK3_PRE / 4
:	:
$1111\ 1110_B$	CK3_PRE / 255
$1111\ 1111_B$	CK3_PRE / 256

30.3.3 Details of TAUJn Control Registers

30.3.3.1 TAUJnCDRm — TAUJn Channel Data Register

This register functions either as a compare register or as a capture register, depending on the operating mode specified in TAUJnCMORm.TAUJnMD[4:1].

Access: Readable/writable in 32-bit units.
- When this register functions as a capture register, only reading is possible. Write operation is ignored.
- When this register functions as a compare register, reading and writing is possible.

Address: <TAUJn_base> + 0_H + m × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAUJnCDR[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCDR[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30.12 TAUJnCDRm Register Contents

Bit Position	Bit Name	Function
31 to 0	TAUJnCDR [31:0]	Data register for capture/compare values

30.3.3.2 TAUJnCNTm — TAUJn Channel Counter Register

This is a channel m counter register.

Access: Only readable in 32-bit units.

Address: <TAUJn_base> + 10_H + m × 4_H

Value after reset: FFFF $FFFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TAUJnCNT[31:16]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCNT[15:0]															
Value after reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 30.13 TAUJnCNTm Register Contents

Bit Position	Bit Name	Function
31 to 0	TAUJnCNT [31:0]	32-bit counter value

The read value depends on a counter, an operating mode change, or TAUJnTS.TAUJnTSm/TAUJnTT.TAUJnTTm bit value.

The initial counter read value depends on the operating mode and how the counter is stopped.

- By a reset
- By a counter stop trigger (TAUJnTT.TAUJnTTm = 1)

The following table lists the initial counter read values after the counter is stopped (TAUJnTE.TAUJnTEm = 0) and re-enabled (TAUJnTS.TAUJnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUJnTS.TAUJnTSm = 1) with the counter waiting for a start trigger.

Table 30.14 TAUJnCNTm Read Values after Re-Enabling Counter

Mode Name	Count Method (Up/Down)	TAUJnCNTm		
		Start Value*1	After Stop Trigger	After One Count
Interval timer mode	Count down	FFFF FFFF$_H$	Stop value	—
Capture mode	Count up	0000 0000$_H$	Stop value	—
One-count mode	Count down	FFFF FFFF$_H$	Stop value	Stop value
Capture and one-count mode	Count up	0000 0000$_H$	Stop value	Capture value + 1 (TAUJnCDRm)
Count capture mode	Count up	0000 0000$_H$	Stop value	—
Gate count mode	Count down	FFFF FFFF$_H$	Stop value	Stop value
Capture and gate count mode	Count up	0000 0000$_H$	Stop value	Stop value

Note 1. The value set for TAUJnCNTm when operating mode is changed after reset release.

30.3.3.3 TAUJnCMORm — TAUJn Channel Mode OS Register

This register controls channel m operation.

Access: Readable/writable in 16-bit units.
Writable only when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + 80_H + m × 4_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 30.15 TAUJnCMORm Register Contents (1/3)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Selects an operation clock, which is used with the TAUJTTINm input edge detection circuit. Setting of TAUJnCMORm.TAUJnCCS[1:0] bits also allows the operation clock to serve as the TAUJnCNTm count clock.

TAUJn CKS1	TAUJn CKS0	Selection of Operation Clock
0	0	CK0
0	1	CK1
1	0	CK2
1	1	CK3

Bit Position	Bit Name	Function
13, 12	TAUJnCCS[1:0]	Selects a count clock for TAUJnCNTm counter.

TAUJn CCS1	TAUJn CCS0	Selection of Operation Clock
0	0	Operation clock specified by TAUJnCMORm.TAUJnCKS[1:0].
0	1	Setting prohibited
1	0	
1	1	

Bit Position	Bit Name	Function
11	TAUJnMAS	Specifies whether the channel is a master or slave channel during synchronous channel operation. 0: Slave 1: Master This bit setting is valid only for even channels (CHm_even). Odd channels (CHm-odd) are fixed to 0.

Table 30.15 TAUJnCMORm Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUJnSTS[2:0]</td><td>Selects an external start trigger.
<table>
<tr><th>TAUJn STS2</th><th>TAUJn STS1</th><th>TAUJn STS0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of TAUJTTINm input signal, which is specified by TAUJnCMURm.TAUJnTIS[1:0].</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of TAUJTTINm input signal is used as a start trigger and the reverse edge as a stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INT of master channel</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td></td></tr>
<tr><td>1</td><td>1</td><td>1</td><td></td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>TAUJnCOS[1:0]</td><td>Specifies the timing for updating capture register TAUJnCDRm and overflow flag TAUJnCSRm.TAUJnOVF of channel m. These bits are only valid if channel m is for capture function (capture mode and capture & one-count mode)..
<table>
<tr><th>TAUJn COS1</th><th>TAUJn COS0</th><th>TAUJnCDRm</th><th>TAUJnCSRm.TAUJnOVF</th></tr>
<tr><td>0</td><td>0</td><td>Updated when valid edge of TAUJTTINm input is detected.</td><td>Updated (cleared or set) when valid edge of TAUJTTINm input is detected.
• Set TAUJnCSRm.TAUJnOVF if a counter overflow has occurred since the last valid edge was detected.
• Clear TAUJnCSRm.TAUJnOVF if no counter overflow has occurred since the last valid edge was detected.</td></tr>
<tr><td>0</td><td>1</td><td></td><td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated when valid edge of TAUJTTINm input is detected and when a counter overflow occurs.
• Detection of valid edge of TAUJTTINm input: The counter value is written into TAUJnCDRm.
• Occurrence of overflow: FFFF FFFF$_H$ is loaded into TAUJnCDRm. Detection of the next valid edge of TAUJTTINm is ignored.</td><td>No setting</td></tr>
<tr><td>1</td><td>1</td><td>Set when a counter overflow occurs and cleared by setting TAUJnCSCm.TAUJnCLOV to 1.</td></tr>
</table></td></tr>
<tr><td>5</td><td>Reserved</td><td>When read, the value after reset is returned. When writing to this bit, write the value after reset.</td></tr>
</table>

Table 30.15 TAUJnCMORm Register Contents (3/3)

Bit Position	Bit Name	Function
4 to 0	TAUJnMD[4:0]	Specifies an operating mode.

TAUJn MD4	TAUJn MD3	TAUJn MD2	TAUJn MD1	TAUJn MD0	Functional Description
0	0	0	0	1/0	Interval timer mode
0	0	0	1	1/0	Setting prohibited
0	0	1	0	1/0	Capture mode
0	0	1	1	0	Setting prohibited
0	1	0	0	1/0	One-count mode
0	1	0	1	1/0	Setting prohibited
0	1	1	0	0	Capture and one-count mode
0	1	1	1	1/0	Setting prohibited
1	0	0	0	0	Setting prohibited
1	0	0	1	0	Setting prohibited
1	0	1	0	1/0	Setting prohibited
1	0	1	1	1/0	Count capture mode
1	1	0	0	0	Gate count mode
1	1	0	1	0	Capture and gate count mode
Others					Setting prohibited

Mode	Role of TAUJnMD0 Bit
Interval timer mode Capture mode Count capture mode	Specifies whether INTTAUJnIm is generated at the beginning of count operation (when a start trigger is entered) or not. 0: INTTAUJnIm is not generated. 1: INTTAUJnIm is generated.
One-count mode	Enables/disables start trigger detection during counting. 0: Disables detection. 1: Enables detection. **CAUTION** In one-count mode, INTTAUJnIm signal is not output at the beginning of count operation.
Capture and one-count mode Gate count mode Capture and gate count mode	This bit should be set to 0. **CAUTION** INTTAUJnIm signal is not output at the beginning of count operation. In addition, start trigger detected during counting is disabled.

30.3.3.4 TAUJnCMURm — TAUJn Channel Mode User Register

This register specifies a type of valid edge detection used for TAUJTTINm input.

Access: Readable/writable in 8-bit units.

Address: <TAUJn_base> + 20_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.16 TAUJnCMURm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	Specifies a valid edge of TAUJTTINm input signal.

TAUJn TIS1	TAUJn TIS0	Functional Description
0	0	Falling edge
0	1	Rising edge
1	0	Detection of falling and rising edges (selection of low width measurement) Start trigger: Falling edge Stop trigger (capture): Rising edge
1	1	Detection of falling and rising edges (selection of high width measurement) Start trigger: Rising edge Stop trigger (capture): Falling edge

Edge detection of TAUJTTINm input signal is based on the operation clock selected by TAUJnCMORm.TAUJnCKS[1:0].

30.3.3.5 TAUJnCSRm — TAUJn Channel Status Register

This register indicates the overflow status of channel m.

Access: Only readable in 8-bit units.

Address: <TAUJn_base> + 30_H + m × 4_H

Value after reset: 000000x0_B

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnOVF
Value after reset	0	0	0	0	0	0	×	0
R/W	R	R	R	R	R	R	R	R

Table 30.17 TAUJnCSRm Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	Reserved	When read, undefined value is returned.
0	TAUJnOVF	Indicates the counter overflow status: 0: No overflow occurs 1: Overflow occurs This bit is used only in the following modes: • Capture mode • Capture and one-count mode The function of this bit depends on the setting of control bits TAUJnCMORm.TAUJnCOS[1:0].

30.3.3.6 TAUJnCSCm — TAUJn Channel Status Clear Trigger Register

This register is a trigger register for clearing the overflow flag TAUJnCSRm.TAUJnOVF of channel m.

Access: Only writable in 8-bit units. It is always read as 00_H.

Address: <TAUJn_base> + 40_H + m × 4_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TAUJnCLOV
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 30.18 TAUJnCSCm Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	TAUJnCLOV	0: No function 1: Clears the overflow flag TAUJnCSRm.TAUJnOVF

30.3.3.7 TAUJnTS — TAUJn Channel Start Trigger Register

This register enables the counter operation for each channel.

Access: Only writable in 8-bit units. It is always read as 00_H.

Address: <TAUJn_base> + 54_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTS03	TAUJnTS02	TAUJnTS01	TAUJnTS00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 30.19 TAUJnTS Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing to these bits, write the value after reset.
3 to 0	TAUJnTSm	Enables the counter operation for channel m: 0: No function 1: Enables the counter operation and sets TAUJnTE.TAUJnTEm = 1. Only the counter operation is enabled even if TAUJnTE.TAUJnTEm = 1. Whether the counter is started or not depends on the selected operating mode.

Note that index m is representing a double-digit number for this register. See above.

30.3.3.8 TAUJnTE — TAUJn Channel Enable Status Register

This register indicates whether a counter operation is enabled.

Access: Only readable in 8-bit units.

Address: <TAUJn_base> + 50_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTE03	TAUJnTE02	TAUJnTE01	TAUJnTE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 30.20 TAUJnTE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	TAUJnTEm	Indicates whether channel m's counter operation is enabled. 0: Counter operation is disabled 1: Counter operation is enabled This bit is set to 1 when trigger input of TAUJnTSSTm (synchronous channel start trigger signal) is detected or when TAUJnTS.TAUJnTSm is set to 1. This bit is reset to 0 when TAUJnTT.TAUJnTTm is set to 1.

Note that index m is representing a double-digit number for this register. See above.

30.3.3.9 TAUJnTT — TAUJn Channel Stop Trigger Register

This register stops the counter operation of each channel.

Access: Only writable in 8-bit units. It is always read as 00_H.

Address: <TAUJn_base> + 58_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTT03	TAUJnTT02	TAUJnTT01	TAUJnTT00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 30.21 TAUJnTT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing to these bits, write the value after reset.
3 to 0	TAUJnTTm	Stops channel m's counter operation. 0: No function 1: Stops the counter operation and resets TAUJnTE.TAUJnTEm. TAUJnCNTm, TAUJnTO.TAUJnTOm, and TAUJTTOUTm retain the values provided before the counter is stopped.

Note that index m is representing a double-digit number for this register. See above.

30.3.4 Details of TAUJn Simultaneous Rewrite Register

30.3.4.1 TAUJnRDE — TAUJn Channel Reload Data Enable Register

This register enables and disables simultaneous rewrite of the data register TAUJnCDRm. It also enables and disables simultaneous rewrite of the data register TAUJnTOLm for the PWM output function.

Access: This register can be read/written in 8-bit units. It can only be written when TAUJnTE.TAUJnTEm = 0.

Address: <TAUJn_base> + $A0_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDE03	TAUJnRDE02	TAUJnRDE01	TAUJnRDE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 30.22 TAUJnRDE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3 to 0	TAUJnRDEm	Enables/disables simultaneous rewrite of the data register of channel m: 0: Disables simultaneous rewrite 1: Enabled simultaneous rewrite

Note that index m is representing a double-digit number for this register. See above.

30.3.4.2 TAUJnRDM — TAUJn Channel Reload Data Mode Register

This register selects when the signal that controls simultaneous rewrite is generated.

Access: This register can be read/written in 8-bit units. It can only be written when TAUJnTE.TAUJnTEm = 0.

Address: <TAUJn_base> + $A4_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDM03	TAUJnRDM02	TAUJnRDM01	TAUJnRDM00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 30.23 TAUJnRDM Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3 to 0	TAUJnRDMm	Specifies when the signal that triggers simultaneous rewrite is generated: 0: When the master channel counter starts counting 1: No function These bits only apply when TAUJnRDE.TAUJnRDEm = 1.

Note that index m is representing a double-digit number for this register. See above.

30.3.4.3 TAUJnRDT — TAUJn Channel Reload Data Trigger Register

This register triggers the simultaneous rewrite enabling state.

Access: This register can only be written in 8-bit unit. It is always read as 00_H.

Address: <TAUJn_base> + 68_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRDT03	TAUJnRDT02	TAUJnRDT01	TAUJnRDT00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 30.24 TAUJnRDT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing to these bits, write the value after reset.
3 to 0	TAUJnRDTm	Triggers the simultaneous rewrite enabling state. 0: No function 1: Simultaneous rewrite enabling state is triggered. The simultaneous rewrite enabling flag (TAUJnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger. These bits only apply when: • TAUJnRDE.TAUJnRDEm = 1

Note that index m is representing a double-digit number for this register. See above.

30.3.4.4 TAUJnRSF — TAUJn Channel Reload Status Register

This flag register indicates the simultaneous rewrite status.

Access: This register can only be read in 8-bit units.

Address: <TAUJn_base> + $6C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnRSF03	TAUJnRSF02	TAUJnRSF01	TAUJnRSF00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 30.25 TAUJnRSF Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3 to 0	TAUJnRSFm	Indicates the simultaneous rewrite status. 0: Indicates that simultaneous rewrite has been completed due to the generation of simultaneous rewrite trigger. 1: Indicates that the system waits for a simultaneous rewrite trigger in the simultaneous rewrite enabling state (TAUJnRDTm = 1).

Note that index m is representing a double-digit number for this register. See above.

30.3.5 Details of TAUJn Output Registers

30.3.5.1 TAUJnTOE — TAUJn Channel Output Enable Register

This register enables and disables independent channel output mode controlled by software.

Access: This register can be read/written in 8-bit units.

Address: <TAUJn_base> + 60_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOE03	TAUJnTOE02	TAUJnTOE01	TAUJnTOE00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 30.26 TAUJnTOE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3 to 0	TAUJnTOEm	Enables/disables independent channel output function: 0: Disables independent timer output function (controlled by software) 1: Enables independent timer output function

Note that index m is representing a double-digit number for this register. See above.

30.3.5.2 TAUJnTO — TAUJn Channel Output Register

This register specifies and reads the level of TAUJTTOUTm.

Access: This register can be read/written in 8-bit units.

Address: <TAUJn_base> + $5C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTO03	TAUJnTO02	TAUJnTO01	TAUJnTO00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 30.27 TAUJnTO Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3 to 0	TAUJnTOm	Specifies and reads the level of TAUJTTOUTm: 0: Low 1: High Only TAUJnTOm bits for which Independent Channel Output function is disabled (TAUJnTOEm = 0) can be written.

Note that index m is representing a double-digit number for this register. See above.

30.3.5.3 TAUJnTOM — TAUJn Channel Output Mode Register

This register specifies the output mode of each channel.

Access: This register can be read/written in 8-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + 98_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOM03	TAUJnTOM02	TAUJnTOM01	TAUJnTOM00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 30.28 TAUJnTOM Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3 to 0	TAUJnTOMm	Specifies the channel output mode: 0: Independent channel output mode 1: Synchronous channel output mode The output mode depends on the settings of channel output control (TAUJnTOE.TAUJnTOEm) bits.

Note that index m is representing a double-digit number for this register. See above.

30.3.5.4 TAUJnTOC — TAUJn Channel Output Configuration Register

This register specifies the output mode of each channel in combination with TAUJnTOMm.

Access: This register can be read/written in 8-bit units. It can only be written when the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address: <TAUJn_base> + $9C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOC03	TAUJnTOC02	TAUJnTOC01	TAUJnTOC00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 30.29 TAUJnTOC Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3 to 0	TAUJnTOCm	Specifies the output mode: 0: Operation mode 1 (= Toggle mode) 1: No function This bit must be set to 0 for all output modes except independent channel output mode controlled by software. The output mode also depends on TAUJnTOM.TAUJnTOMm, as shown in the following table.

TAUjnTOMm	TAUJnTOCm	Function
0	0	Toggle mode: Toggling proceeds when INTTAUJnIm occurs.
0	1	No function
1	0	Synchronous channel operation mode 1: Set when INT occurs on the master channel and reset when INT occurs on the slave channel.
1	1	No function

Note that index m is representing a double-digit number for this register. See above.

30.3.5.5 TAUJnTOL — TAUJn Channel Output Level Register

This register specifies the output logic of the channel output bit (TAUJnTO.TAUJnTOm).

Access: This register can be read/written in 8-bit units.

Address: <TAUJn_base> + 64_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	TAUJnTOL03	TAUJnTOL02	TAUJnTOL01	TAUJnTOL00
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 30.30 TAUJnTOL Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
3 to 0	TAUJnTOLm	Specifies the output logic of the channel m output bit (TAUJnTO.TAUJnTOm): 0: Positive logic (active high) 1: Negative logic (active low) These bits apply in all channel output modes except independent channel output mode controlled by software and independent channel output mode 1.

Note that index m is representing a double-digit number for this register. See above.

30.3.5.6 TAUJnEMU — TAUJn Emulation Register

This register controls operation by SVSTOP.

Access: This register can be read/written in 8-bit units.
A write should be performed when counters are stopped (TAUJnTE.TAUJnTEm = 0) and EPC.SVSTOP = 0.

Address: <TAUJn_base> + $A8_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	TAUJn SVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 30.31 TAUJnEMU Register Contents

Bit Position	Bit Name	Function
7	TAUJnSVSDIS	When EPC.SVSTOP bit = 0: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint), regardless of the value of this bit (1 or 0). When EPC.SVSTOP bit = 1: 0: The count clock is stopped when the debugger takes control of the microcontroller (as in the breakpoint). 1: Supply of the count clock continues when the debugger takes control of the microcontroller (as in the breakpoint).
6 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

30.4 Operating Procedure

The following lists the general operation procedure for the TAUJn:

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. All circuits and registers of all channels are initialized. The control register of TAUJTTOUTm is also initialized and outputs a low level.

1. Set the TAUJnTPS and TAUJnBRS registers to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUJn function:
   - Set the operation mode
   - Set any other control bits
3. Enable the counter by setting the TAUJnTS.TAUJnTSm bit to 1.
   The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.
4. If desired, and if possible for the configured function, stop the counter or perform a forced restart operation during count operation. The counter can be stopped by setting the TAUJnTT.TAUJnTTm bit to 1. The counter can be forcibly restarted by setting the TAUJnTS.TAUJnTSm bit to 1.
5. Stop the function by setting the TAUJnTT.TAUJnTTm bit to 1.

NOTES

1. A detailed description of the required control bits and the operation of the individual functions are given in Section 30.12, Independent Channel Operation Functions and Section 30.13, Synchronous Channel Operation Functions.
2. The function can be changed while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

30.5 Concepts of Synchronous Channel Operation Function

The synchronous channel operation function is implemented by using a combination of channel groups (comprised of master and slave channels).

Several rules apply to the settings of channels.

These rules are detailed in Section 30.5.1, Rules of Synchronous Channel Operation Function.

The synchronous channel operation function are detailed in the following section.

- **Section 30.13, Synchronous Channel Operation Functions**

30.5.1 Rules of Synchronous Channel Operation Function

Number of master and slave channels

- Only even channels (CH0, CH2) can be set as master channels. Any channel other than CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and several slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 can be set as slave channel.
- If two master channels are used, slave channels cannot cross the master.
  Example: If CH0 and CH2 are master channels, CH1 can be set as slave channel for CH0, but CH3 cannot.

Operation clock

- The same operation clock should be set for the master channel and the slave channels synchronizing to the master channel. This is achieved by setting the same value in the TAUJnCMORm.TAUJnCKS[1:0] bits of the master and slave channels.

The basic concepts of master/slave usage and operation clocks are illustrated in **Figure 30.3**.

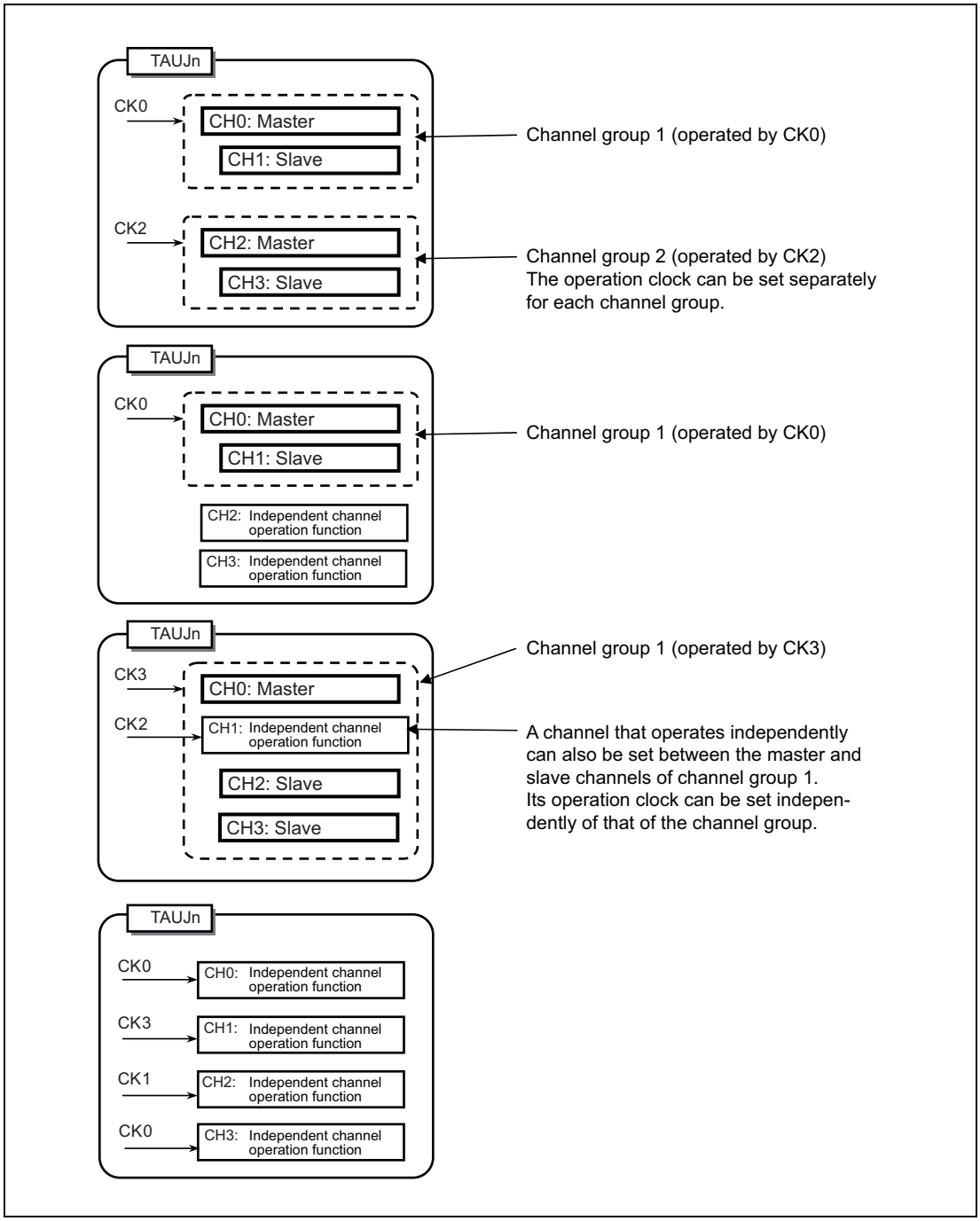


Figure 30.3 Grouping of Channels and Assignment of Operation Clocks

30.5.2 Simultaneous Start and Stop Of Synchronous Channel Counters

Channels that are operated synchronously can be started and stopped simultaneously, both within a TAUJ unit and between TAUJ units.

30.5.2.1 Simultaneous Start and Stop within a TAUJ Unit

- To simultaneously start synchronized channels, the TAUJnTS.TAUJnTSm bits of the channels should be set at the same time.
- To simultaneously stop synchronized channels, the TAUJnTT.TAUJnTTm bits of the channels should be set at the same time.

Setting 1 in the TAUJnTS.TAUJnTSm bits sets the corresponding TAUJnTE.TAUJnTEm bits to 1, enabling counting. The count start timing of the counter depends on the operating mode.

30.6 Simultaneous Rewrite

30.6.1 How to Control Simultaneous Rewrite

The following figure shows the general procedure for simultaneous rewrite. The three main blocks (Initial settings, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

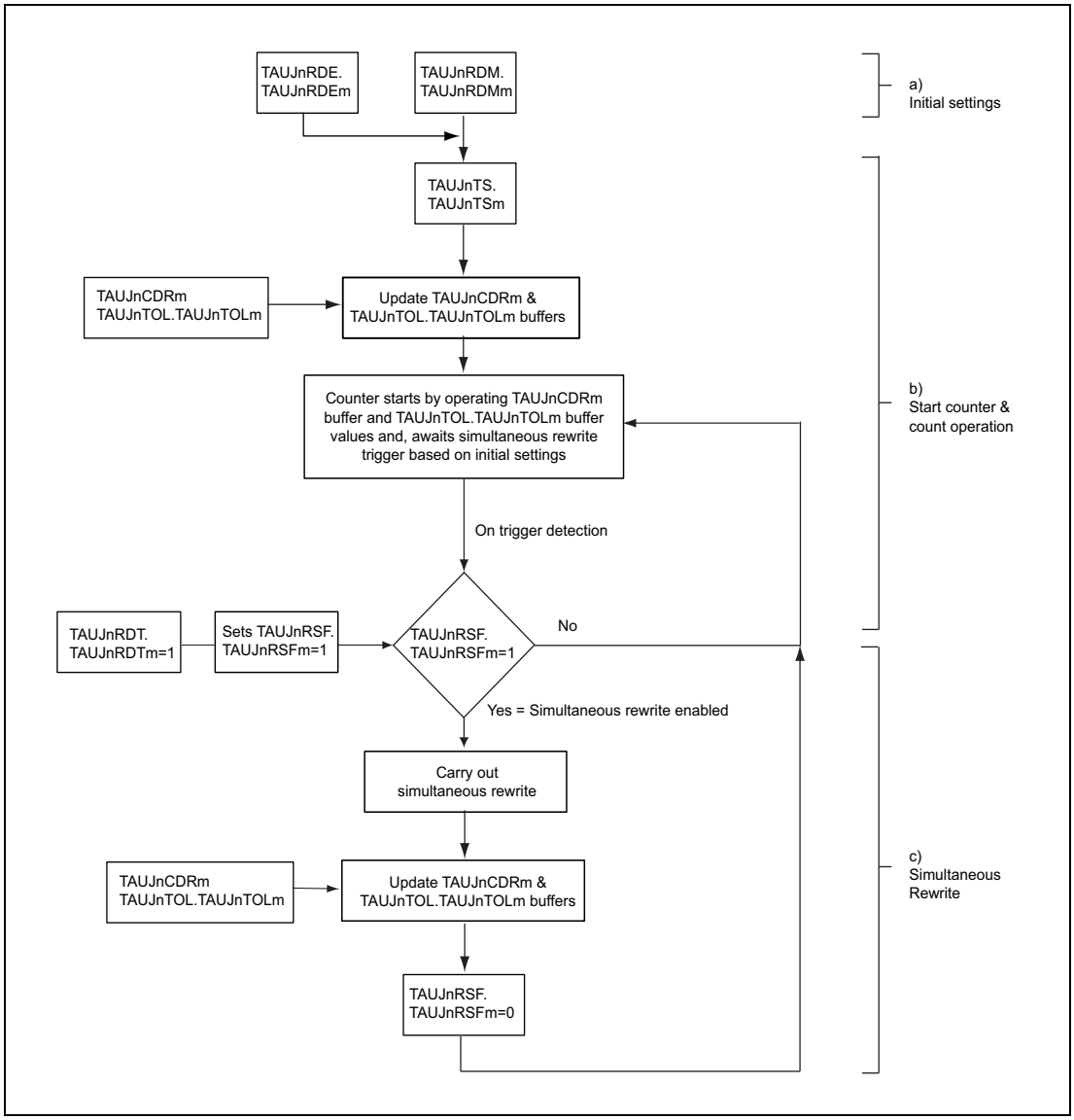


Figure 30.4 General Procedure for Simultaneous Rewrite

30.6.1.1 Initial Settings

- To enable simultaneous rewrite in channel m, set TAUJnRDE.TAUJnRDEm = 1.
- To select simultaneous rewrite when the master channel starts counting, set TAUJnRDM.TAUJnRDMm.

30.6.1.2 Start Counter and Count Operation

- To start all the TAUJnCNTm counters in the channel group, set the corresponding TAUJnTS.TAUJnTSm bits to 1. The values of TAUJnTOL.TAUJnTOLm and the data registers (TAUJnCDRm) are loaded into the corresponding TAUJnTOL.TAUJnTOLm buffer (TAUJnTOL.TAUJnTOLm buf) and data buffer registers (TAUJnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUJnRDT.TAUJnRDTm) to 1 sets the reload flag (TAUJnRSF.TAUJnRSFm) to 1, enabling simultaneous rewrite. TAUJnRSF.TAUJnRSFm remains set to 1 until simultaneous rewrite is completed.
- When a specified trigger for simultaneous rewrite is detected, the TAUJnRSF.TAUJnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1). If enabled, simultaneous rewrite is carried out. Otherwise simultaneous rewrite is not carried out and the system waits for detection of the next simultaneous rewrite trigger.

30.6.1.3 Simultaneous Rewrite

- When the simultaneous rewrite trigger is detected and simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1), the current values of the data registers are copied to their buffers. These values are then loaded into the corresponding counters and the values are applied the next time the counter starts or restarts.
- When the simultaneous rewrite is completed, the TAUJnRSF.TAUJnRSFm bit is set to 0, and the system awaits the next simultaneous rewrite trigger.

30.6.2 Other General Rules for Simultaneous Rewrite

The following rules also apply.

- TAUJnRDE.TAUJnRDEm and TAUJnRDM.TAUJnRDMm cannot be changed while the counter is in operation (TAUJnTE.TAUJnTEm = 1).
- TAUJnTOL.TAUJnTOLm can be rewritten only during operation using the PWM output function. For all other functions, TAUJnTOL.TAUJnTOLm should be written before the counter starts. If it is rewritten while any other function is used, TAUJTTOUTm outputs an invalid waveform.

30.6.3 Simultaneous Rewrite Procedure

The simultaneous rewrite procedure with PWM output function is described in the following figure.

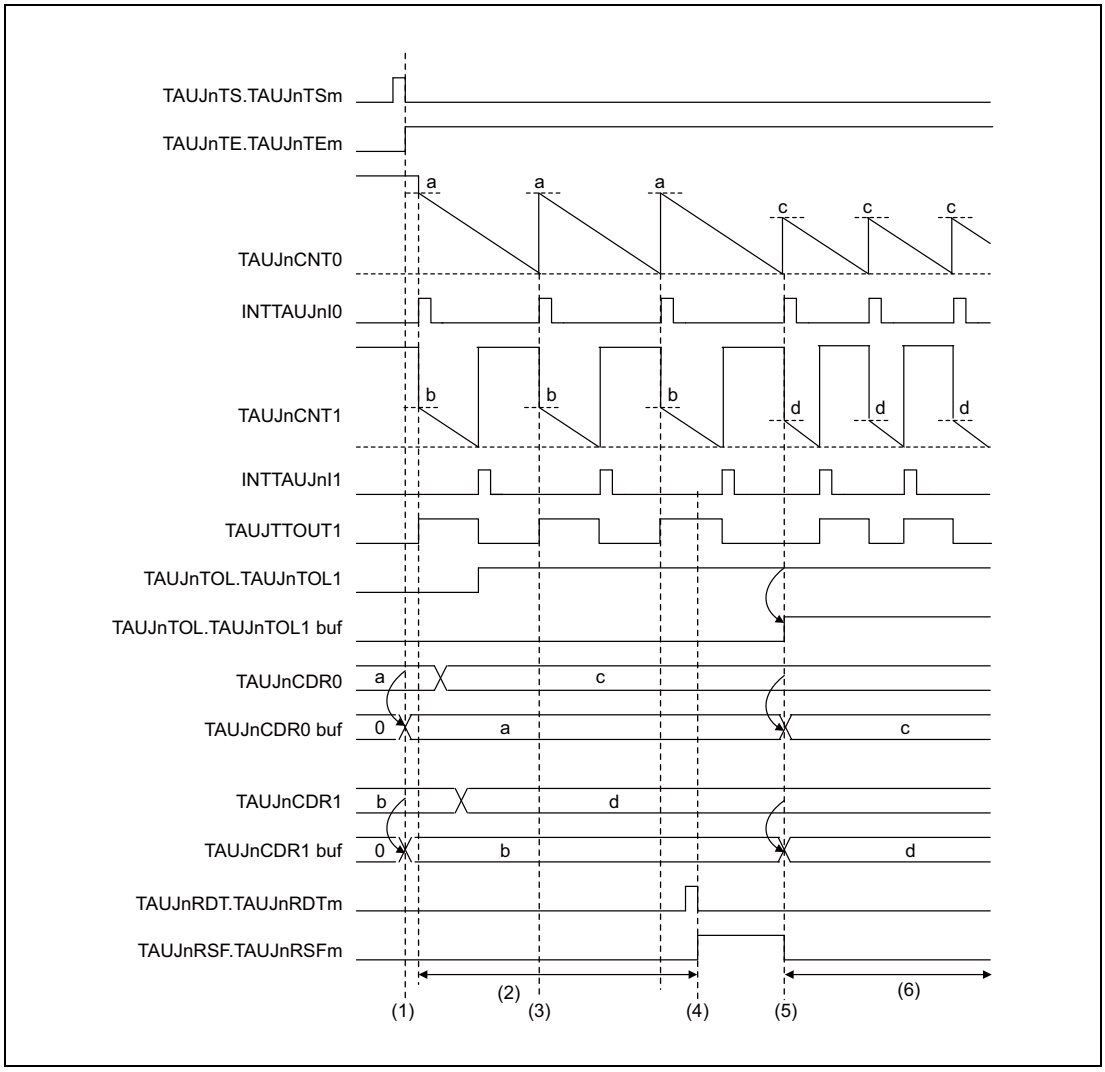


Figure 30.5 Simultaneous Rewrite with PWM Output Function

Setting:

CH0 is a master channel of PWM output function, and CH1 is a slave channel of PWM output function. Simultaneous rewrite is applied when the master channel starts counting.

Description:

(1) When TAUJnTS.TAUJnTSm = 1 is set, the value of TAUJnCDRm is copied to the TAUJnCDRm buffer and the value of TAUJnTOL.TAUJnTOLm is copied to the TAUJnTOL.TAUJnTOLm buffer.

(2) The TAUJnCDRm and TAUJnTOL.TAUJnTOLm registers can be written at any time.

(3) CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUJnRSF.TAUJnRSFm = 0).

(4) The reload data trigger bit (TAUJnRDT.TAUJnRDTm) is set to 1 which sets the status flag (TAUJnRSF.TAUJnRSFm = 1), enabling simultaneous rewrite.

(5) Simultaneous rewrite is triggered when CH0 restarts counting, because simultaneous rewrite is enabled. The TAUJnCDRm value is loaded into the TAUJnCDRm buffer and the TAUJnTOL.TAUJnTOLm value is loaded into the TAUJnTOL.TAUJnTOLm buffer.

(6) The counters count down and await the next simultaneous rewrite trigger. The values of TAUJnCDRm and TAUJnTOL.TAUJnTOLm can be changed again.

30.7 Channel Output Modes

The output of the TAUJTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes.

- By software (TAUJnTOE.TAUJnTOEm = 0)
  When controlled by software, the value written in the output register bit (TAUJnTO.TAUJnTOm) is sent to the output pin (TAUJTTOUTm).
- By TAUJ signals (TAUJnTOE.TAUJnTOEm = 1)
  When controlled by TAUJ signals, the output level of TAUJTTOUTm is set or reset or toggled by internal signals. The value of TAUJnTO.TAUJnTOm is updated accordingly to reflect the value of TAUJTTOUTm.
  - Independently (TAUJnTOM.TAUJnTOMm = 0)
    In case of independent operation, the output of the TAUJTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation should be selected (TAUJnTOM.TAUJnTOMm = 0).
  - Synchronously (TAUJnTOM.TAUJnTOMm = 1)
    In case of synchronous operation, the output of the TAUJTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation should be selected for all synchronized channels (TAUJnTOM.TAUJnTOMm = 1).

The TAUJnTO.TAUJnTOm bit can always be read to determine the current value of TAUJTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in Table 30.32, Channel Output Modes.

The channel output modes are described in details below.

- Section 30.7.2, Channel Output Modes Controlled Independently by TAUJn Signals
- Section 30.7.3, Channel Output Modes Controlled Synchronously by TAUJn Signals

Batch operation of TAUJnTOm bit

Whether a set value is reflected to the TAUJnTOm bit or not is controlled by the TAUJnTOE.TAUJnTOEm bit.

The TAUJnTOm setting is written only to the bit (channel) set with TAUJnTOE.TAUJnTOEm bit = 0 when a write to the TAUJnTO register is attempted. No TAUJnTOm setting is reflected to the bit (channel) set with TAUJnTOE.TAUJnTOEm bit = 1.

NOTE

TAUJnTO.TAUJnTOm bit is placed so that its bit number corresponds to a channel number.

Output logic

Positive logic or negative logic of the output is specified by control bit TAUJnTOL.TAUJnTOLm.

The value of TAUJnTOL.TAUJnTOLm bit should be set before the counter is started. It can only be changed during operation with PWM output function. Otherwise, changes to TAUJnTOL.TAUJnTOLm result in an invalid TAUJTTOUTm signal output.

See Section 30.6, Simultaneous Rewrite.

The various channel output modes and the channel output control bits are listed in **Table 30.32**.

Table 30.32 Channel Output Modes

Channel Output Mode	TAUJnTOE.TAUJnTOEm	TAUJnTOM.TAUJnTOMm
By software		
Independent channel output mode controlled by software	0	x
By TAUJ signals, independently		
Independent channel output mode 1	1	0
By TAUJ signals, synchronously		
Synchronous channel output mode 1	1	1

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

NOTE

The following bits cannot be changed during count operation (TAUJnTE.TAUJnTEm = 1):

- TAUJnTOM.TAUJnTOMm
- TAUJnTOC.TAUJnTOCm

30.7.1 General Procedures for Specifying a Channel Output Mode

This section describes the general procedures for specifying a TAUJTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUJnTOE.TAUJnTOEm = 0).

(1) Set TAUJnTO.TAUJnTOm to specify the initial level of the TAUJTTOUTm output.

(2) Set channel output mode according to Table 30.32, Channel Output Modes, and the output logic using the TAUJnTOL.TAUJnTOLm bit.

(3) Start the counter (TAUJnTS.TAUJnTSm = 1).

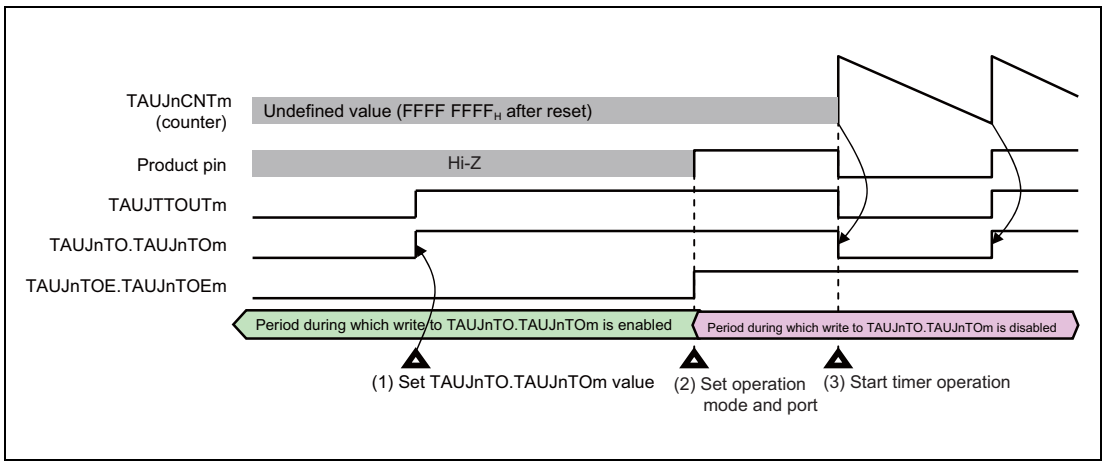


Figure 30.6 General Procedure for Specifying a TAUJTTOUTm Channel Output Mode

30.7.2 Channel Output Modes Controlled Independently by TAUJn Signals

This section lists the channel output modes that are controlled independently by TAUJn signals. The control bits used to specify a mode are listed in Table 30.32, Channel Output Modes.

30.7.2.1 Independent Channel Output Mode 1

Set/reset conditions

In this output mode, TAUJTTOUTm toggles when INTTAUJnIm is detected. The value of TAUJnTOL.TAUJnTOLm is ignored.

Prerequisites

There are no prerequisites other than those shown in Table 30.32, Channel Output Modes.

30.7.3 Channel Output Modes Controlled Synchronously by TAUJn Signals

This section lists the channel output modes that are controlled synchronously by TAUJn signals. The control bits used to specify a mode are listed in Table 30.32, Channel Output Modes.

30.7.3.1 Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUJnIm of master channel serves as a set signal and INTTAUJnIm of the slave channel as a reset signal. If INTTAUJnIm of master channel and INTTAUJnIm of the slave channel are generated at the same time, INTTAUJnIm of the slave channel (reset signal) has priority over INTTAUJnIm (set signal) of master channel, i.e., the master channel is ignored.

Prerequisites

There are no prerequisites other than those shown in Table 30.32, Channel Output Modes.

30.8 Start Timing in Each Operating Modes

This section describes the timing at which the counter starts after TAUJnTS.TAUJnTSm is set to 1 in each operating mode.

In all modes, the value of data register and whether or not an interrupt occurs depends on mode and register settings.

CAUTION

The count start timing described in this section is for your reference. Actually, the count start timing depends on the count clock timing.

30.8.1 Interval Timer Mode, Judge Mode, Capture Mode, Count-Up/-Down Mode, and Count Capture Mode

The counter starts operating with the next count clock cycle after TAUJnTS.TAUJnTSm is set to 1. The value of data register is also loaded when the counter starts.

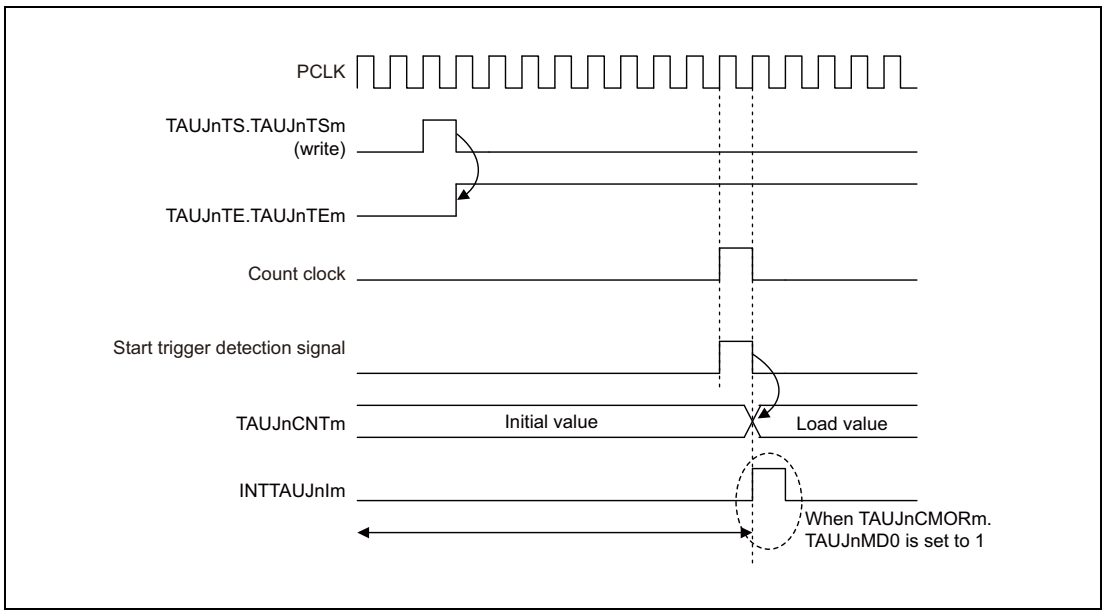


Figure 30.7 Start Timing in Interval Timer Mode, Capture Mode, and Count Capture Mode

30.8.2 Other Operating Modes

In other operating modes, count clock cycle is irrelevant to start of counter operation. The counter operation start timing is triggered only upon detection of a valid edge of TAUJTTINm. Once the counter starts, the value of data register is also loaded. The count clock cycles, which is irrelevant to start of counter operation, determine the frequency with which all operations take place.

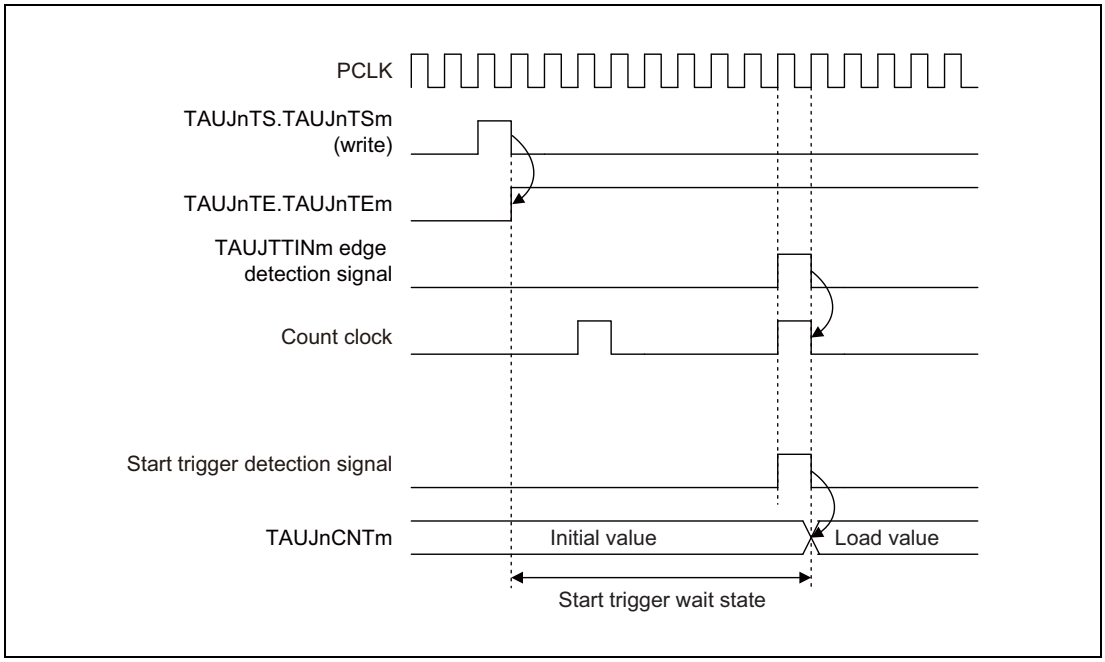


Figure 30.8 Start Timing in Other Operating Modes

30.9 TAUJTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUJnIm is generated using the TAUJnCMORm.TAUJnMD0 bit. The generation of INTTAUJnIm when the MD0 bit starts counting and the effect to TAUJTTOUTm depend on the selected function. For details, refer to the description of TAUJnCMORm.TAUJnMD0 of each function.

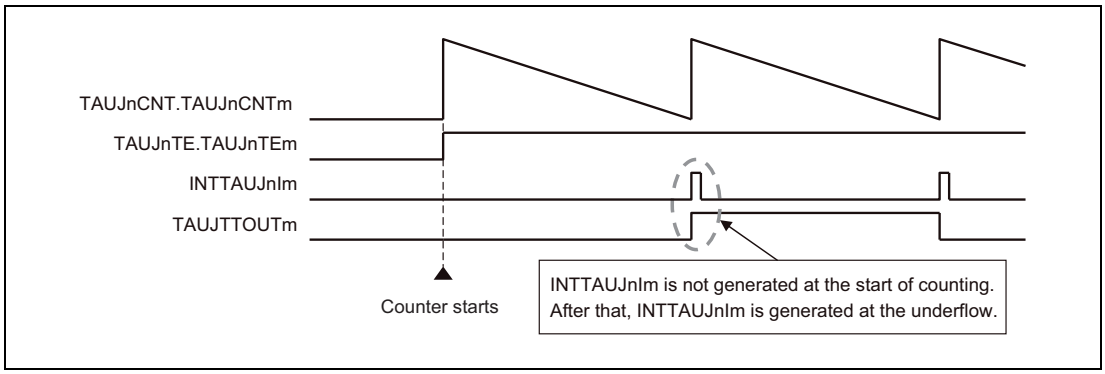


Figure 30.9 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJnMD0 = 0)

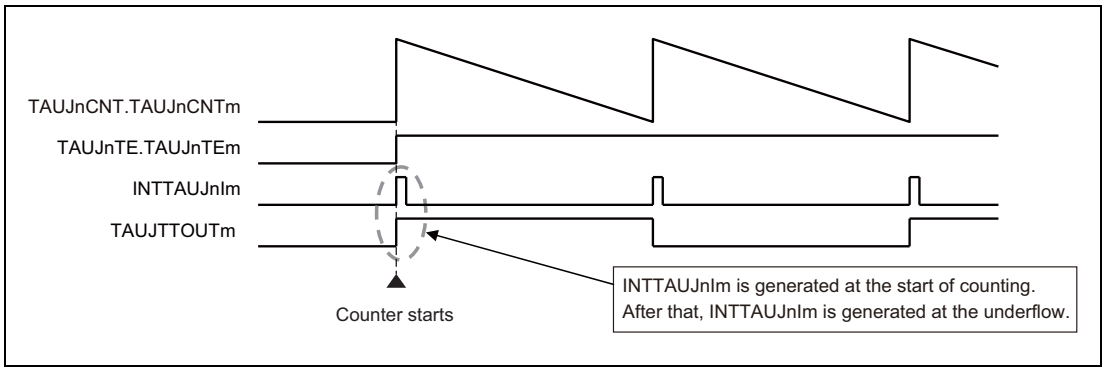


Figure 30.10 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJnMD0 = 1)

30.10 Interrupt Generation upon Overflow

Certain independent functions that count up, overflow without generating an interrupt when they reach FFFF $FFFF_H$. This section describes how it is possible to generate an interrupt, by combining a channel operating in one of these modes with a channel in a different operation mode which counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find an operation mode for the second channel that counts down in such a manner, that it reaches 0000 0000_H at the same time as the first channel overflows (TAUJnCNTm = FFFF $FFFF_H$).
- Set TAUJnCDRm of the second channel to FFFF $FFFF_H$.
- The two channels must count at the same speed (i.e. they must have the same count clock).
- Both channels are triggered by the same TAUJTTINm input.
- The trigger detection settings (TAUJnCMORm.TAUJnSTS[2:0] and TAUJnCMURm.TAUJnTIS[1:0]) must be identical for both channels.

Result:

The down-counter of the second channel reaches 0000 0000_H at exactly the same time as the up-counter of the first channel overflows (TAUJnCNTm = FFFF $FFFF_H$). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

30.11 TAUJTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

The following figure shows when edge detection takes place.

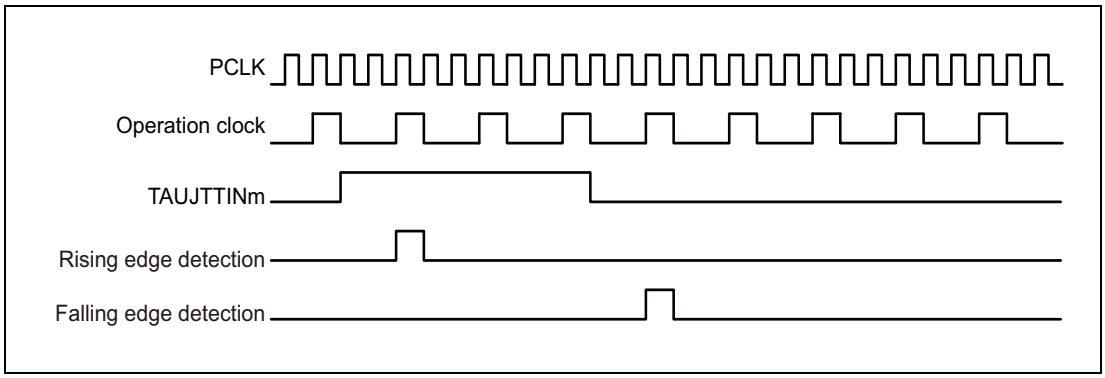


Figure 30.11 Basic Edge Detection Timing

Figure 30.11 shows an image of operation timing. In the actual operation, delay time occurs due to noise filter and synchronization circuit between the TAUJnIm pin and TAUJn.

30.12 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by the TAUJ. For a general overview of independent channel operation functions, see Section 30.2, Overview.

30.12.1 Interval Timer Function

30.12.1.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals. When an interrupt is generated, the TAUJTTOUTm signal toggles, resulting in a square wave.

Functional description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is loaded to TAUJnCNTm and the counter starts to count down from this value.

When the counter reaches 0000 0000_H, INTTAUJnIm is generated and the TAUJTTOUTm signal toggles. TAUJnCNTm then loads the TAUJnCDRm value and subsequently continues operation.

The value of TAUJnCDRm can be rewritten at any time, and the changed value of TAUJnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJTTOUTm stop but retain their values. The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.

Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUJTTOUTm does not toggle. This results in a negative TAUJTTOUTm signal compared to when TAUJnCMORm.TAUJnMD0 is set to 1.

30.12.1.2 Equations

INTTAUJnIm cycle = count clock cycle × (TAUJnCDRm + 1)

TAUJTTOUTm square wave cycle = count clock cycle × (TAUJnCDRm+1) × 2

30.12.1.3 Block Diagram and General Timing Diagram

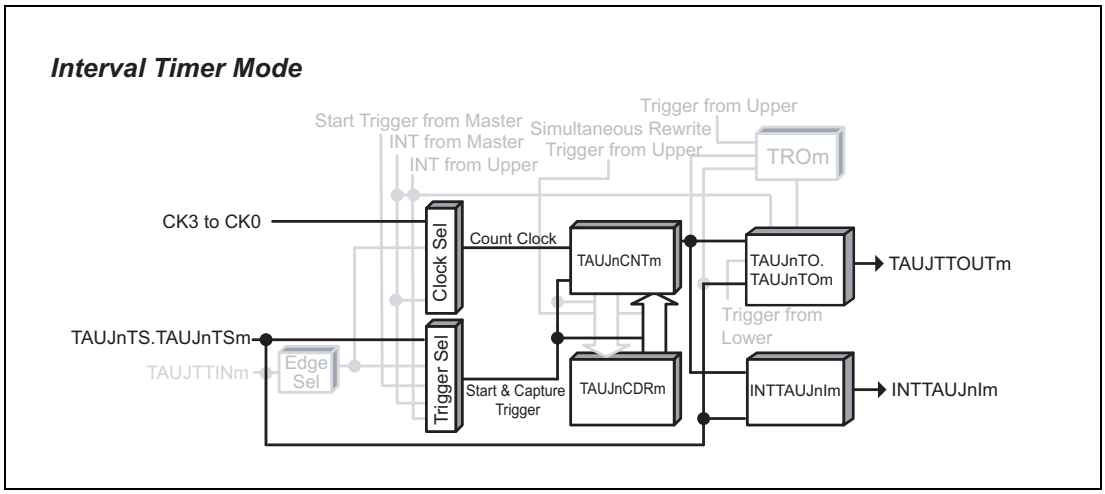


Figure 30.12 Block Diagram for Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at operation start (TAUJnCMORm.(TAUJnMD0 = 1)

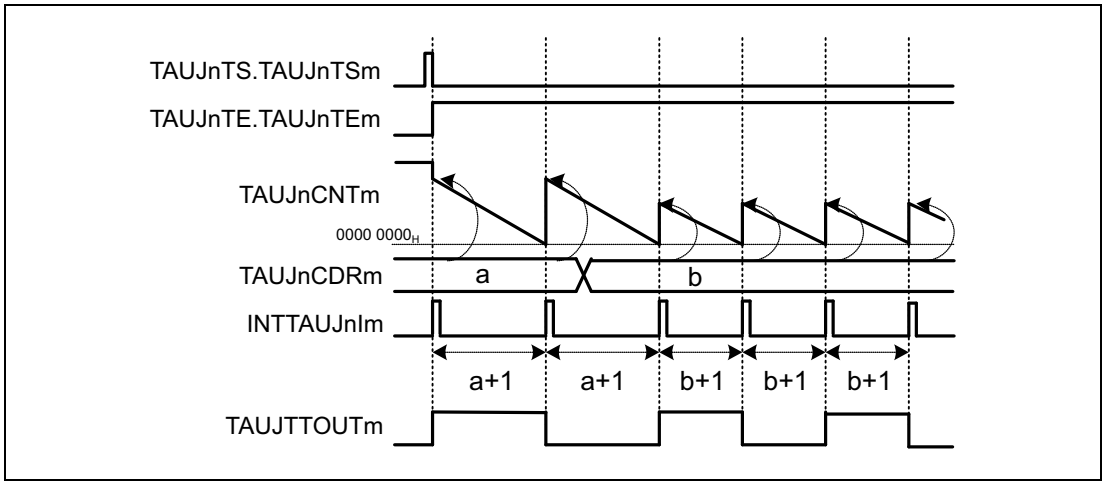


Figure 30.13 General Timing Diagram for Interval Timer Function

30.12.1.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 30.33 Contents of the TAUJnCMORm register for Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 000_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated and TAUJTTOUTm does not toggle when operation starts or restarts. 1: Generates INTTAUJnIm and toggles TAUJTTOUTm when operation starts or restarts.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.34 Contents of the TAUJnCMURm register for Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode

Table 30.35 Control Bit Settings in Independent Channel Output Mode 1

Bit name	Setting
TAUJnTOE.TAUJnTOEm	Write 1_B.
TAUJnTOM.TAUJnTOMm	Write 0_B.
TAUJnTOC.TAUJnTOCm	Write 0_B.
TAUJnTOL.TAUJnTOLm	Write 0_B.

NOTE

The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJTTOUTm can then be controlled independently of the interrupts. For details refer to Section 30.7, Channel Output Modes.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the interval timer function. Therefore, these registers must be set to 0.

Table 30.36 Simultaneous Rewrite Settings for Interval Timer Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

30.12.1.5 Operating Procedure for Interval Timer Function

Table 30.37 Operating Procedure for Interval Timer Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 30.33, Contents of the TAUJnCMORm register for Interval Timer Function** and **Table 30.34, Contents of the TAUJnCMURm register for Interval Timer Function**. Set the value of the TAUJnCDRm register. Set the channel output mode by setting the control bits as described in **Table 30.35, Control Bit Settings in Independent Channel Output Mode 1**.	Channel operation is stopped.

Table 30.37 Operating Procedure for Interval Timer Function

	Operation	Status of TAUJn
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJTTOUTm toggles.
During operation	The TAUJnCDRm register value can be changed at any time. The TAUJnCNTm register can be read at all times.	TAUJnCNTm counts down. When the counter reaches 0000 0000_H: • TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. • INTTAUJnIm is generated and TAUJTTOUTm toggles.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJTTOUTm stop and retain their current values.

Restart operation

30.12.1.6 Specific Timing Diagrams

(1) TAUJnCDRm = 0000 0000_H, count clock = PCLK/2

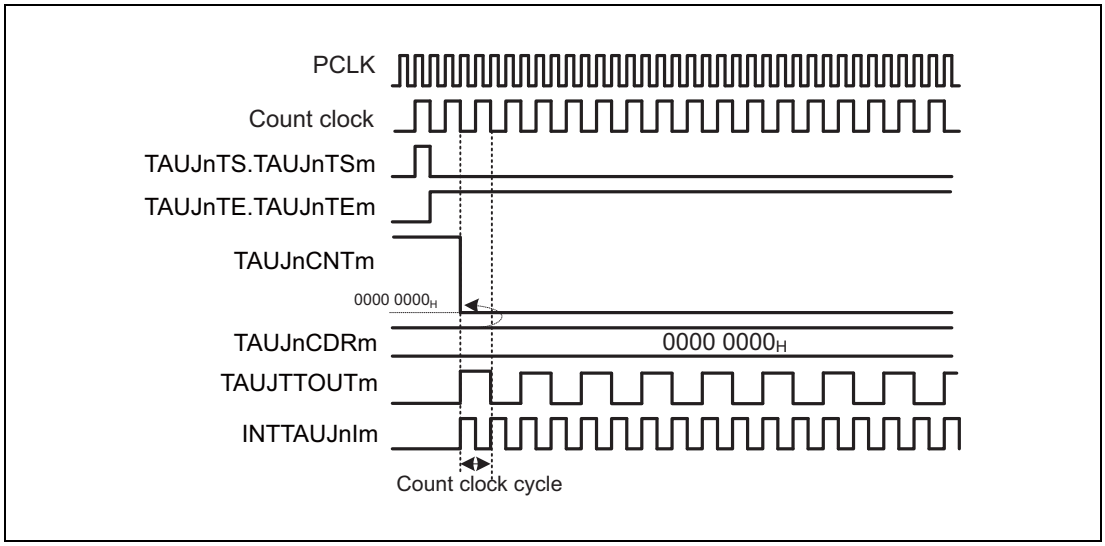


Figure 30.14 TAUJnCDRm = 0000 0000_H, Count Clock = PCLK/2

- If TAUJnCDRm = 0000 0000_H and the count clock = PCLK/2, the TAUJnCDRm value is loaded to TAUJnCNTm every count clock, meaning that TAUJnCNTm is always 0000 0000_H.
- INTTAUJnIm is generated every count clock, resulting in TAUJTTOUTm toggling every count clock.

(2) TAUJnCDRm = 0000 0000_H, count clock = PCLK

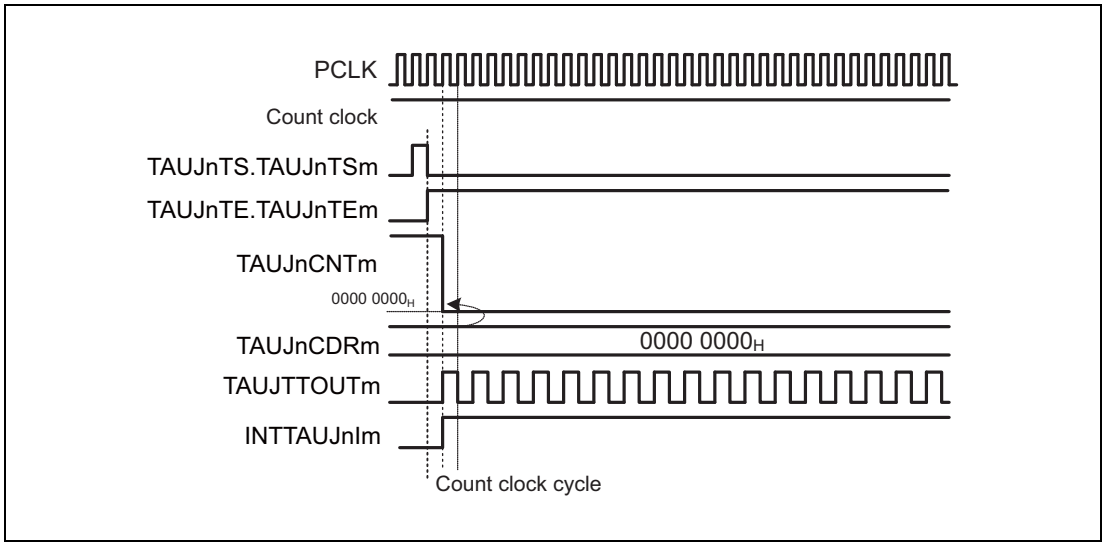


Figure 30.15 TAUJnCDRm = 0000 0000_H, Count Clock = PCLK

- If TAUJnCDRm = 0000 0000_H and the count clock = PCLK, the TAUJnCDRm value is loaded to TAUJnCNTm every PCLK clock, meaning that TAUJnCNTm is always 0000 0000_H.
- INTTAUJnIm is fixed to high level.
  Though the first interrupt is generated, subsequent interrupts are not generated. TAUJTTOUTm is toggled every PCLK clock.

(3) Operation stop and restart

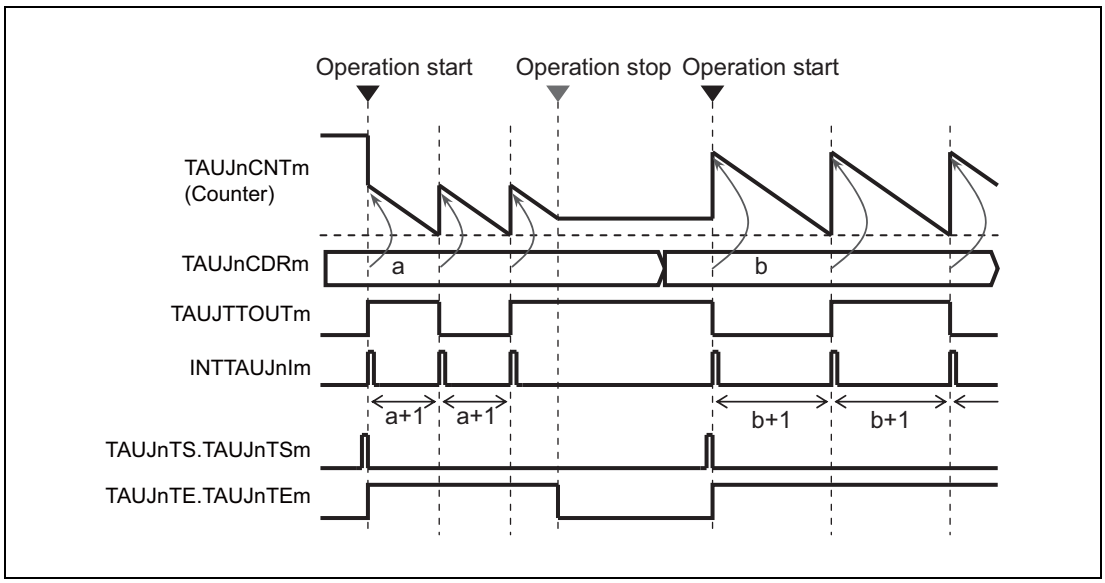


Figure 30.16 Operation Stop and Restart (TAUJnCMORm.TAUJnMD0 = 1)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1.

(4) Forced restart

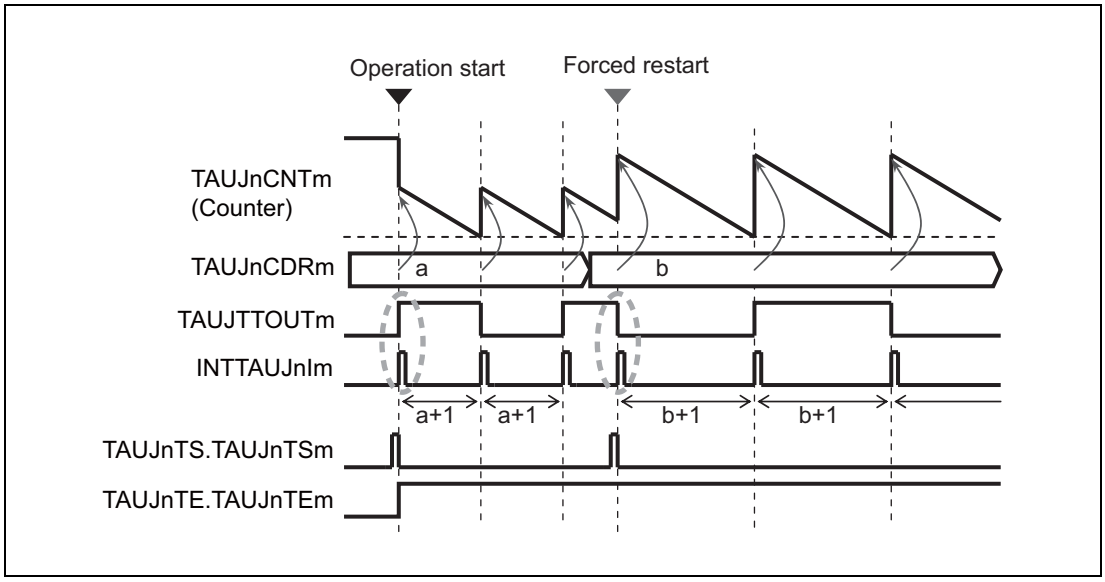


Figure 30.17 Forced Restart Operation (TAUJnCMORm.TAUJnMD0 = 1)

- The counter can be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.
- If the TAUJnCMORm.TAUJnMD0 bit is set to 1, the first interrupt after a start or restart is generated.
- When a forced restart is made, the TAUJnCDRm value is reflected to TAUJnCNTm and counting starts. Execute a forced restart to reflect the changed TAUJnCDRm value immediately.

30.12.2 TAUJTTINm Input Interval Timer Function

30.12.2.1 Overview

Summary

This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals or when a valid TAUJTTINm input edge is detected. When an interrupt is generated, the TAUJTTOUTm signal toggles, resulting in a square wave.

Description

This function operates in an identical manner to the interval timer function (see **Section 30.12.1, Interval Timer Function**), except that this function is restarted by a valid TAUJTTINm input edge. The type of edge used as the trigger is specified using the TAUJnCMURm.TAUJnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edges can be selected.

30.12.2.2 Equations

INTTAUJnIm cycle = count clock cycle × (TAUJnCDRm + 1)

TAUJTTOUTm square wave cycle = count clock cycle × (TAUJnCDRm + 1) × 2

30.12.2.3 Block Diagram and General Timing Diagram

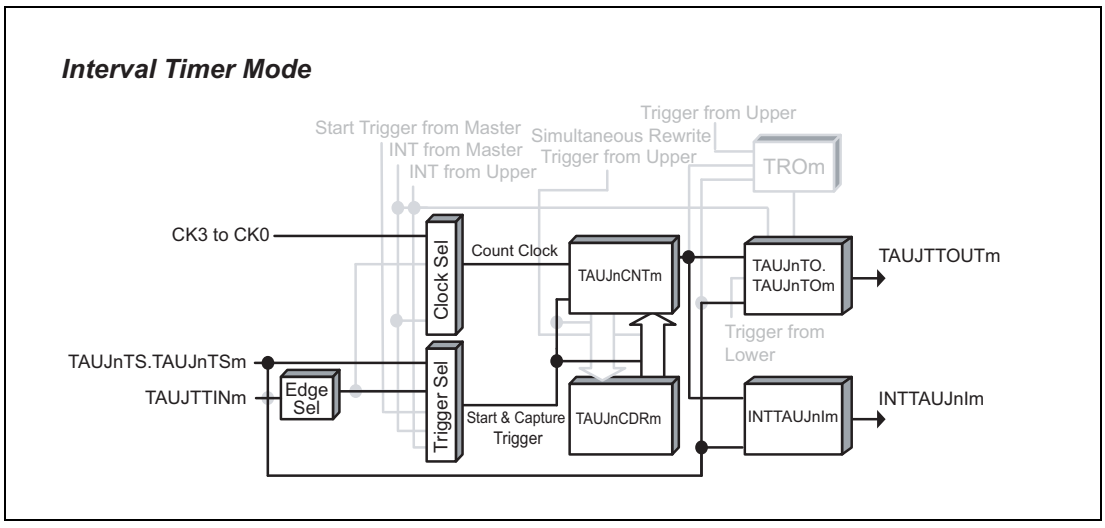


Figure 30.18 Block Diagram for TAUJTTINm Input Interval Timer Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1)
- Rising edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 01_B)

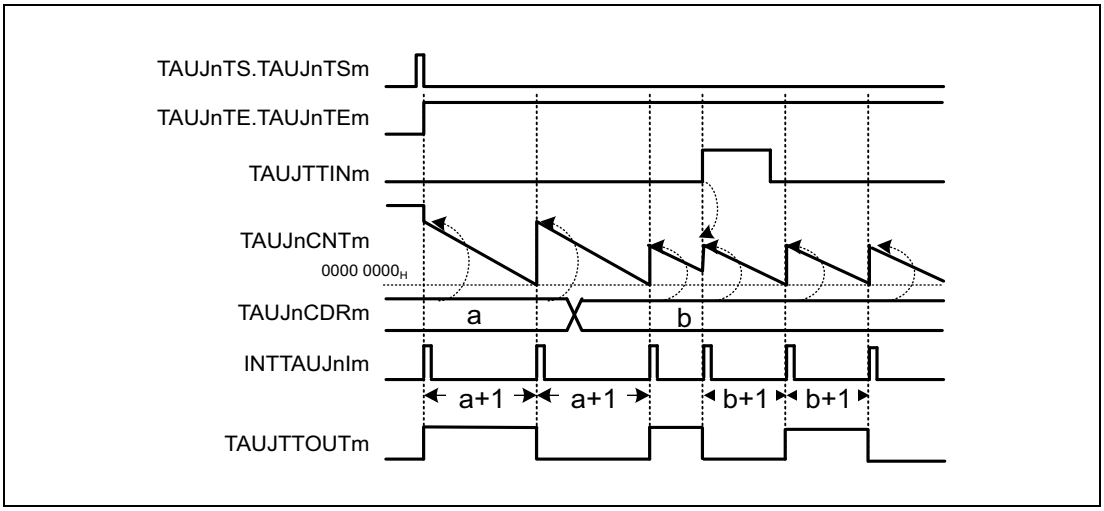


Figure 30.19 General Timing Diagram for TAUJTTINm Input Interval Timer Function

30.12.2.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 30.38 Contents of the TAUJnCMORm register for TAUJTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 001_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated and TAUJTTOUTm does not toggle when operation starts. 1: Generates INTTAUJnIm and toggles TAUJTTOUTm when operation starts.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.39 Contents of the TAUJnCMURm register for TAUJTTINm Input Interval Timer Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

Table 30.40 Control Bit Settings for Independent Channel Output Mode 1

Bit name	Setting
TAUJnTOE.TAUJnTOEm	Write 1_B.
TAUJnTOM.TAUJnTOMm	Write 0_B.
TAUJnTOC.TAUJnTOCm	Write 0_B.
TAUJnTOL.TAUJnTOLm	Write 0_B.

NOTE

The channel output mode can also be set to channel output mode controlled by software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJTTOUTm can then be controlled independently of the interrupts. For details refer to Section 30.7, Channel Output Modes.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input interval timer function. Therefore, these registers must be set to 0.

Table 30.41 Simultaneous Rewrite Settings for TAUJTTINm Input Interval Timer Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

30.12.2.5 Operating Procedure for TAUJTTINm Input Interval Timer Function

Table 30.42 Operating Procedure for TAUJTTINm Input Interval Timer Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 30.38, Contents of the TAUJnCMORm register for TAUJTTINm Input Interval Timer Function** and **Table 30.39, Contents of the TAUJnCMURm register for TAUJTTINm Input Interval Timer Function**. Set the value of the TAUJnCDRm register Set the channel output mode by setting the control bits as described in Table 30.40, Control Bit Settings for Independent Channel Output Mode 1.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJTTOUTm toggles.
During operation	The values of the TAUJnCMURm.TAUJnTIS[1:0] and TAUJnCDRm registers can be changed at any time. The TAUJnCNTm register can be read at all times. Detection of TAUJTTINm edge	TAUJnCNTm counts down. When the counter reaches 0000 0000_H: • TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. • INTTAUJnIm is generated and TAUJTTOUTm toggles. When a TAUJTTINm input valid edge is detected during count operation, TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJTTOUTm stop and retain their current values.

Restart operation

30.12.2.6 Specific Timing Diagrams

The timing diagrams in Section 30.12.1, Interval Timer Function apply, and in addition the counter can also be restarted by an effective TAUJTTINm input edge.

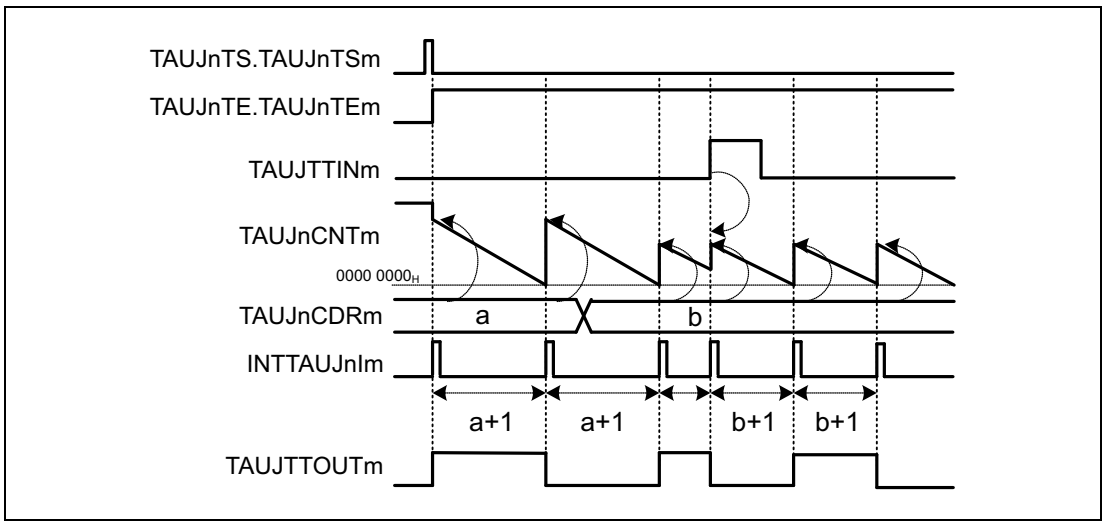


Figure 30.20 Counter Triggered By Rising TAUJTTINm input edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B), TAUJnCMORm.TAUJnMD0 = 1

If a valid TAUJTTINm input edge is detected, an interrupt is generated which causes TAUJTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B).

30.12.3 TAUJTTINm Input Pulse Interval Measurement Function

30.12.3.1 Overview

Summary

This function captures the count value and uses this value and the overflow bit TAUJnCSRm.TAUJnOVF to measure the interval of the TAUJTTINm input signal.

Prerequisites

TAUJTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJTTINm edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter resets to 0000 0000_H and subsequently continues operation.

If the counter reaches FFFF $FFFF_H$ before a valid TAUJTTINm edge is detected, it overflows to 0000 0000_H. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0].

Table 30.43 Effects of an Overflow

	When Overflow Occurs		When a Valid TAUJTTINm Input is then Detected	
TAUJnCMORm.COS[1:0]	TAUJnCDRm	TAUJnCSRm.TAUJnOVF	TAUJnCDRm and TAUJnCNTm	TAUJnCSRm.TAUJnOVF
00	Unchanged	0	TAUJnCNTm loaded to TAUJnCDRm	1
01		1		
10	Set to FFFF $FFFF_H$	0	TAUJnCNTm set to 0, TAUJnCDRm unchanged	Unchanged
11		1		

When TAUJnCMORm.TAUJnCOS[0] = 1, the overflow by (TAUJnCSRm.TAUJnOVF = 1) can only be cleared by setting TAUJnCSCm.TAUJnCLOV = 1.

The combination of the values of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the interval of the TAUJTTINm signal. However, if an overflow occurs multiple times before a valid TAUJTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

The function can be stopped by setting TAUJnTT.TAUJnTTm = 1, which in turn sets TAUJnTE.TAUJnTEm = 0. TAUJnCNTm stops but retains its value. While the function is stopped, TAUJTTINm input valid edge detection and TAUJnCNTm capture are not performed.

Conditions

If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details, refer to Section 30.9, TAUJTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts.

NOTE

When TAUJnCMORm.TAUJnCOS[1] = 1, the value of TAUJnCNTm is not loaded to TAUJnCDRm when the first valid TAUJTTINm input edge occurs after an overflow. However, an interrupt is generated.

30.12.3.2 Equations

TAUJTTINm input pulse interval = count clock cycle ×
[(TAUJnCSRm.TAUJnOVF × (FFFF $FFFF_H$ + 1)) + TAUJnCDRm capture value + 1]

30.12.3.3 Block Diagram and General Timing Diagram

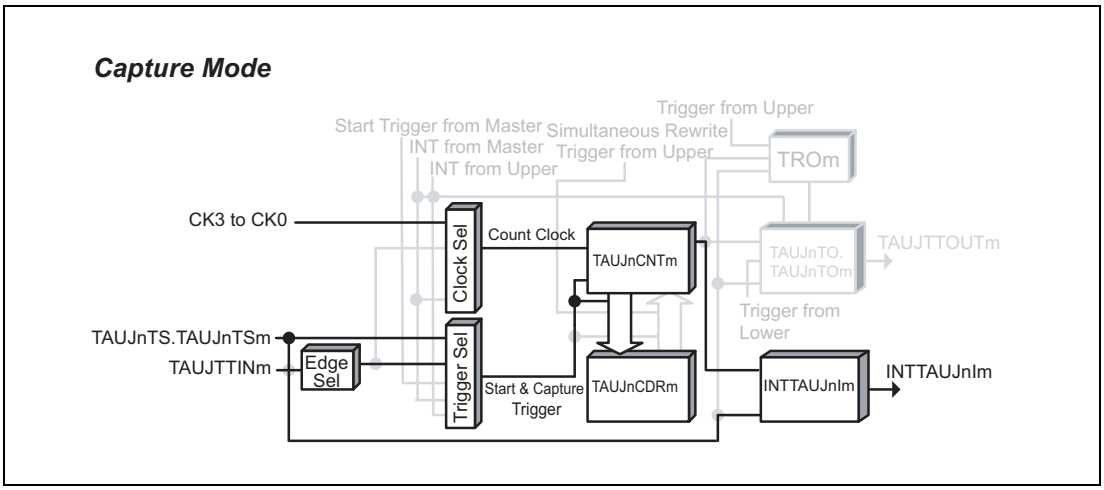


Figure 30.21 Block Diagram for TAUJTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is not generated when operation starts (TAUJnCMORm.TAUJnMD0 = 0).
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)
- When a valid TAUJTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B).

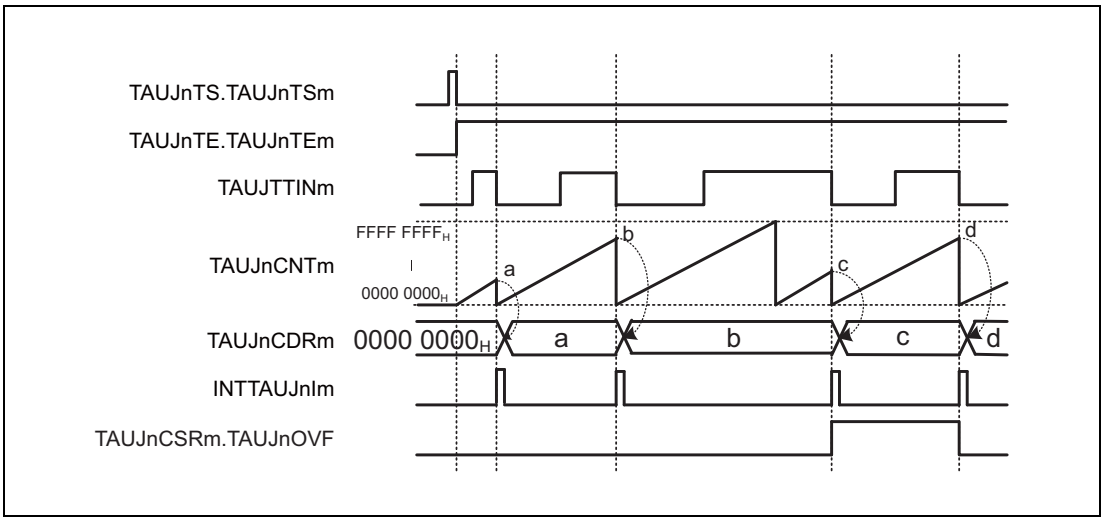


Figure 30.22 General Timing Diagram For TAUJTTINm Input Pulse Interval Measurement Function

30.12.3.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 30.44 Contents of the TAUJnCMORm Register for TAUJTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 001_B.
7, 6	TAUJnCOS[1:0]	See **Table 30.43, Effects of an Overflow**
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0010_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated when operation starts. 1: Generates INTTAUJnIm when operation starts.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.45 Contents of the TAUJnCMURm Register for TAUJTTINm Input Pulse Interval Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input pulse interval measurement function. Therefore, these registers must be set to 0.

Table 30.46 Simultaneous Rewrite Settings for TAUJTTINm Input Pulse Interval Measurement Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

30.12.3.5 Operating Procedure for TAUJTTINm Input Pulse Interval Measurement Function

Table 30.47 Operating Procedure for TAUJTTINm Input Pulse Interval Measurement Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 30.44, Contents of the TAUJnCMORm Register for TAUJTTINm Input Pulse Interval Measurement Function** and **Table 30.45, Contents of the TAUJnCMURm Register for TAUJTTINm Input Pulse Interval Measurement Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm is cleared to 0000 0000_H. INTTAUJnIm is generated when TAUJnCMORm.TAUJnMD0 is set to 1.
During operation	Detection of TAUJTTINm edges. The values of the TAUJnCMURm. TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time. TAUJnCSCm.TAUJnCLOV bit can be written to 1. (TAUJnCSRm.TAUJnOVF bit is cleared to 0.)	TAUJnCNTm starts to count up from 0000 0000_H. When a TAUJTTINm valid edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and returns to 0000 0000_H. • INTTAUJnIm is then generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.

Restart operation

30.12.3.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

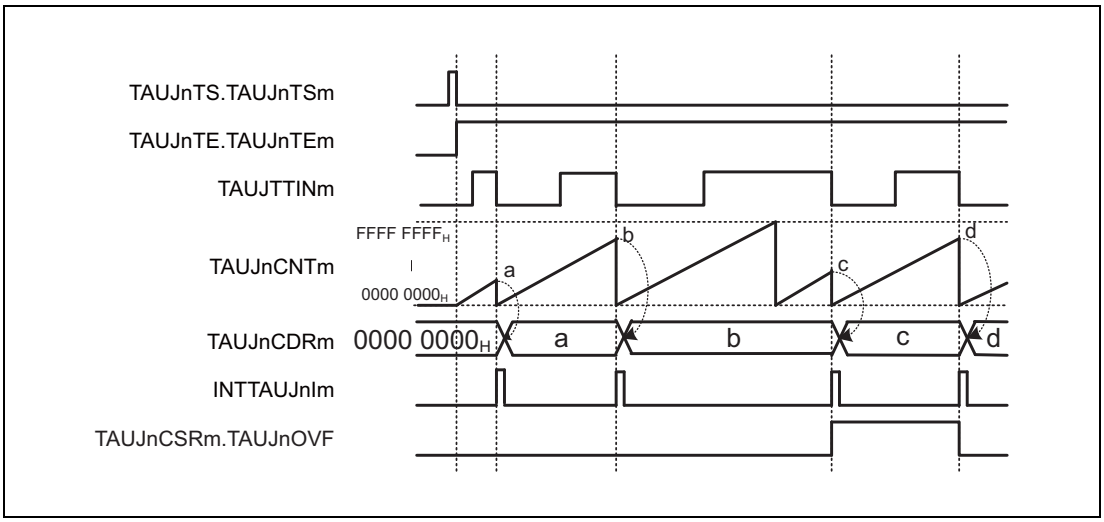


Figure 30.23 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- If the next valid TAUJTTINm input edge is detected when no overflow occurs, TAUJnCSRm.TAUJnOVF is cleared to 0.

(2) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

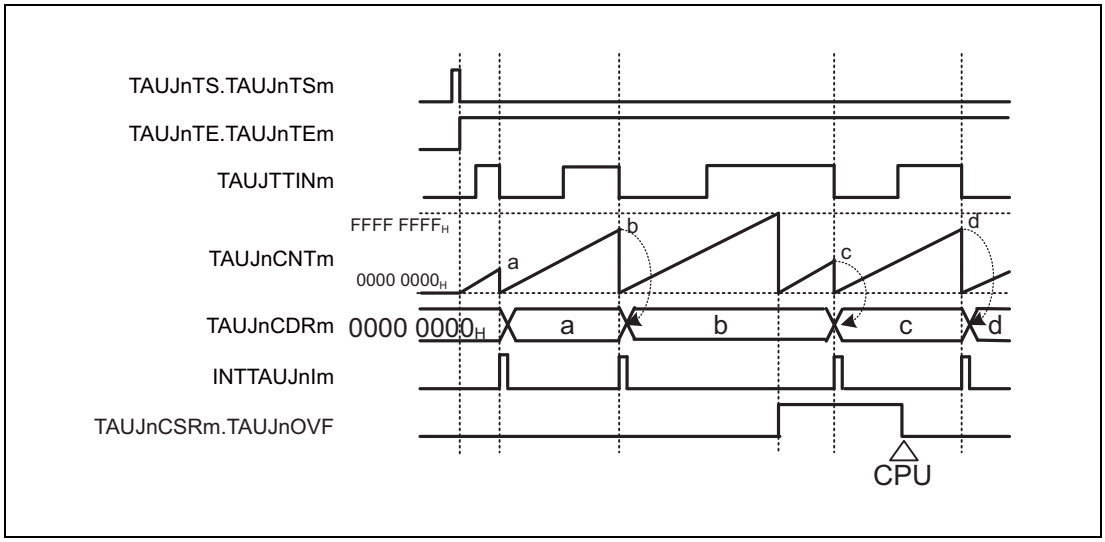


Figure 30.24 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF is set to1.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (by setting the TAUJnCSCm.TAUJnCLOV bit to 1).

(3) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

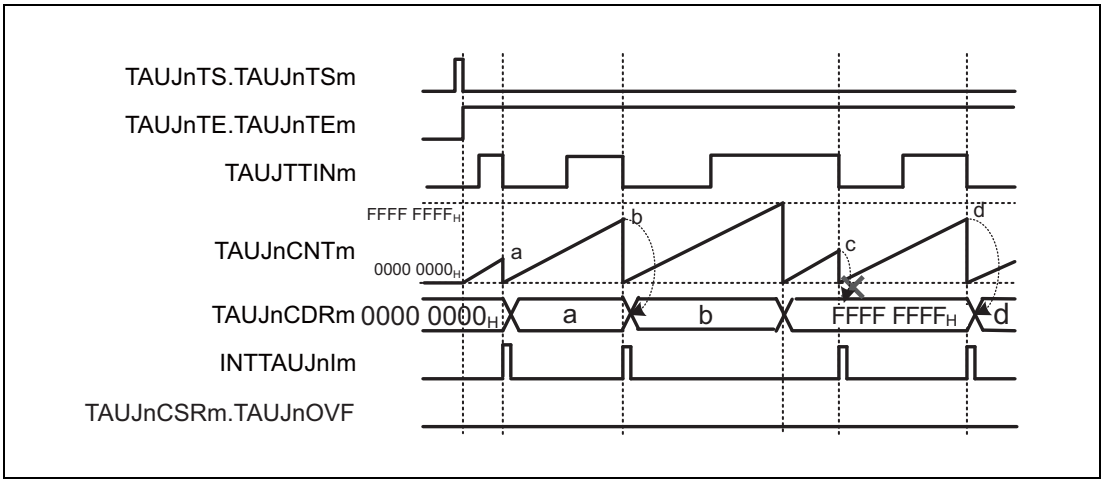


Figure 30.25 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF $FFFF_H$ and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.

(4) TAUJnCMORm.TAUJnCOS[1:0] = 11_B

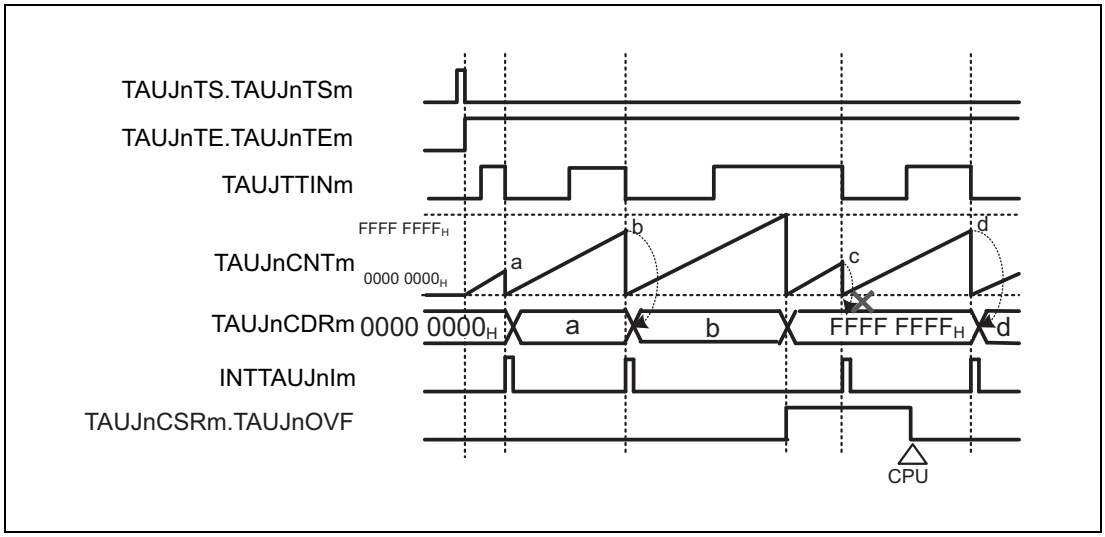


Figure 30.26 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

30.12.4 TAUJTTINm Input Signal Width Measurement Function

30.12.4.1 Overview

Summary

This function measures the width of a TAUJTTINm signal by starting counting on one edge of the TAUJTTINm signal and capturing the counter value on the opposite edge.

Prerequisites

TAUJTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. When a valid TAUJTTINm start edge is detected, the counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJTTINm stop edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter retains its value and awaits the next valid TAUJTTINm input start edge.

If the counter reaches FFFF $FFFF_H$ before a valid TAUJTTINm stop edge is detected, it overflows. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0].

Table 30.48 Effects of an Overflow

	When Overflow Occurs		When a Valid TAUJTTINm Input Stop Edge is Detected	
TAUJnCMORm.COS[1:0]	TAUJnCDRm	TAUJnCSRm.TAUJnOVF	TAUJnCDRm and TAUJnCNTm	TAUJnCSRm.TAUJnOVF
00	Unchanged	0	TAUJnCNTm is loaded to TAUJnCDRm.	1
01		1		
10	Set to FFFF $FFFF_H$	0	TAUJnCNTm stops counting, TAUJnCDRm unchanged	Unchanged
11		1		

When TAUJnCMORm.TAUJnCOS[0] = 1, the overflow bit TAUJnCSRm.TAUJnOVF can only be cleared by setting TAUJnCSCm.TAUJnCLOV to 1.

The combination of the values of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the width of the TAUJTTINm signal. However, if an overflow occurs multiple times before a valid TAUJTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

This function cannot be forcibly restarted.

NOTE

When TAUJnCMORm.COS[1] = 1, the value of TAUJnCNTm is not loaded to TAUJnCDRm when the first valid TAUJTTINm input edge occurs after an overflow. However, an interrupt is generated.

30.12.4.2 Equations

TAUJTTINm input signal width = count clock cycle ×
[(TAUJnCSRm.TAUJnOVF × (FFFF FFFF$_H$ + 1)) + TAUJnCDRm capture value + 1]

30.12.4.3 Block Diagram and General Timing Diagram

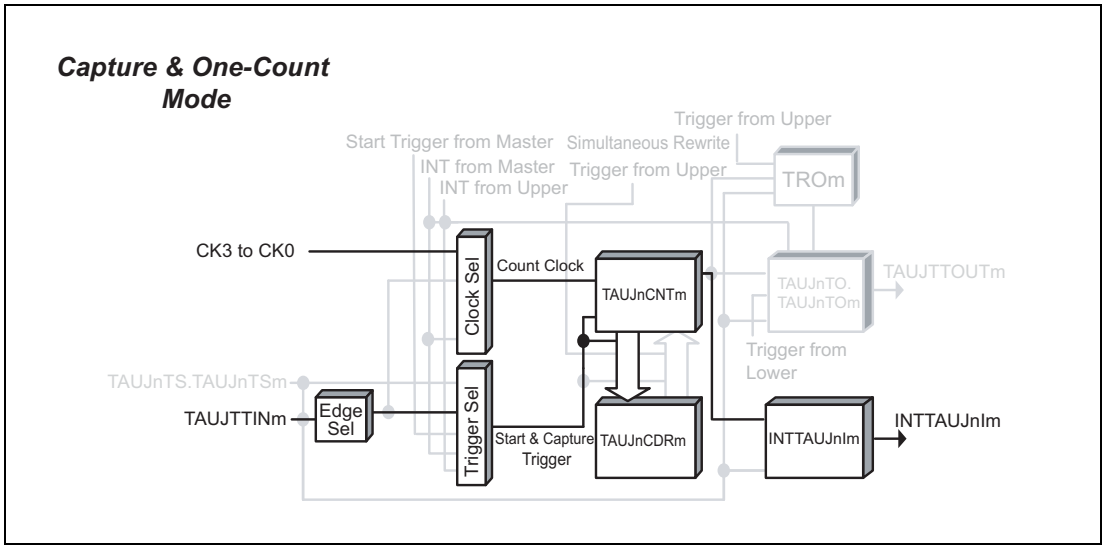


Figure 30.27 Block Diagram for TAUJTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)
- When a valid TAUJTTINm input is detected after an overflow, TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B).

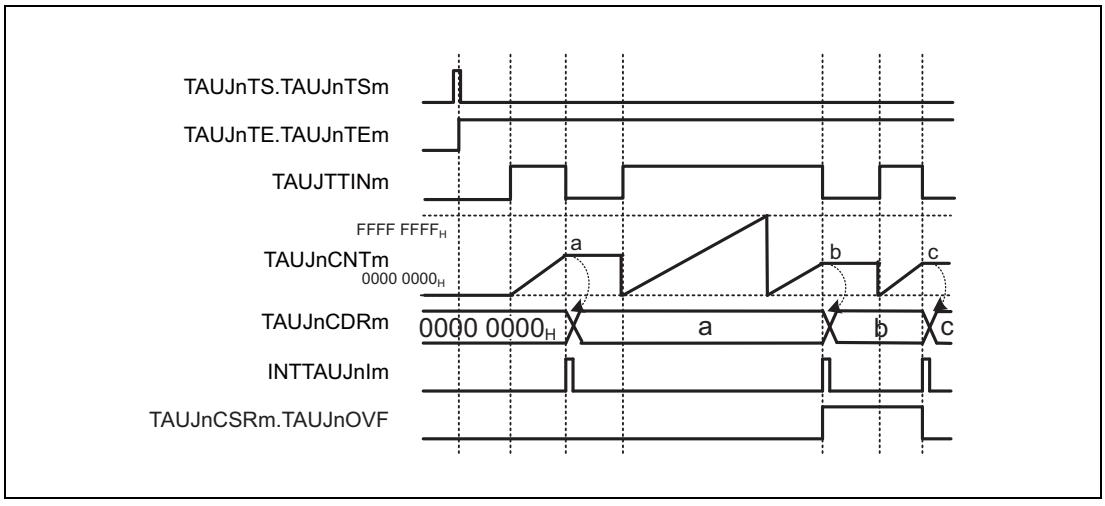


Figure 30.28 General Timing Diagram for TAUJTTINm Input Signal Width Measurement Function

30.12.4.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 30.49 Contents of the TAUJnCMORm Register for TAUJTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	See Table 30.48, Effects of an Overflow.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0110_B.
0	TAUJnMD0	Write 0_B.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.50 Contents of the TAUJnCMURm Register for TAUJTTINm Input Signal Width Measurement Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input signal width measurement function. Therefore, these registers must be set to 0.

Table 30.51 Simultaneous Rewrite Settings for TAUJTTINm Input Signal Width Measurement Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

30.12.4.5 Operating Procedure for TAUJTTINm Input Signal Width Measurement Function

Table 30.52 Operating Procedure for TAUJTTINm Input Signal Width Measurement Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 30.49, Contents of the TAUJnCMORm Register for TAUJTTINm Input Signal Width Measurement Function** and **Table 30.50, Contents of the TAUJnCMURm Register for TAUJTTINm Input Signal Width Measurement Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJTTINm start edge. When a TAUJTTINm start edge is detected, TAUJnCNTm starts to count up.
During operation	The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time. The TAUJnCSCm.TAUJnCLOV bit can be set to 1.	TAUJnCNTm starts to count up from 0000 0000_H. When a TAUJTTINm valid edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and retains its value. • INTTAUJnIm is then generated. • Counting stops at the “value that transferred to TAUJnCDRm + 1” and TAUJnCNTm waits for detection of the TAUJTTINm start edge. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.

Restart operation

30.12.4.6 Specific Timing Diagrams: Overflow Behavior

(1) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

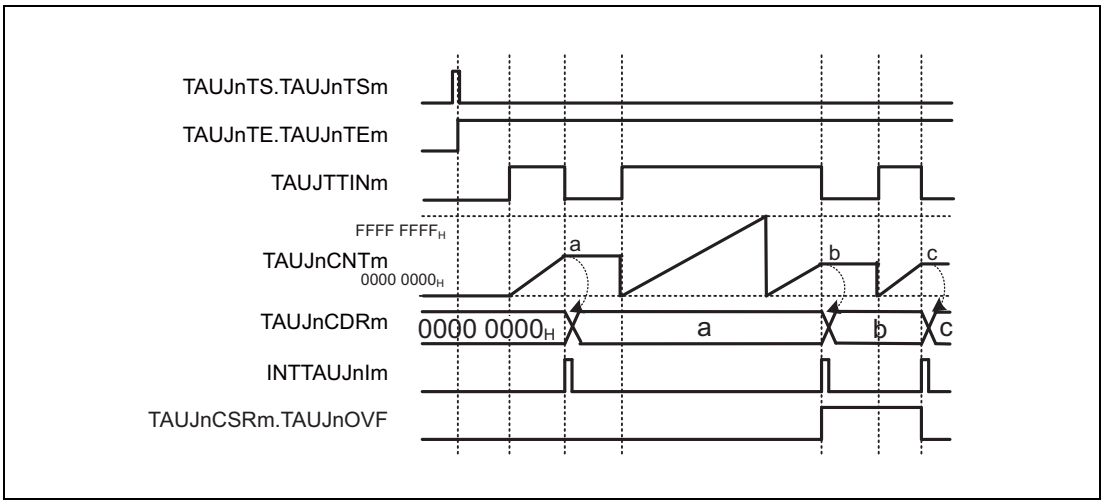


Figure 30.29 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge with no overflow occurring, TAUJnCSRm.TAUJnOVF is cleared to 0.

(2) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

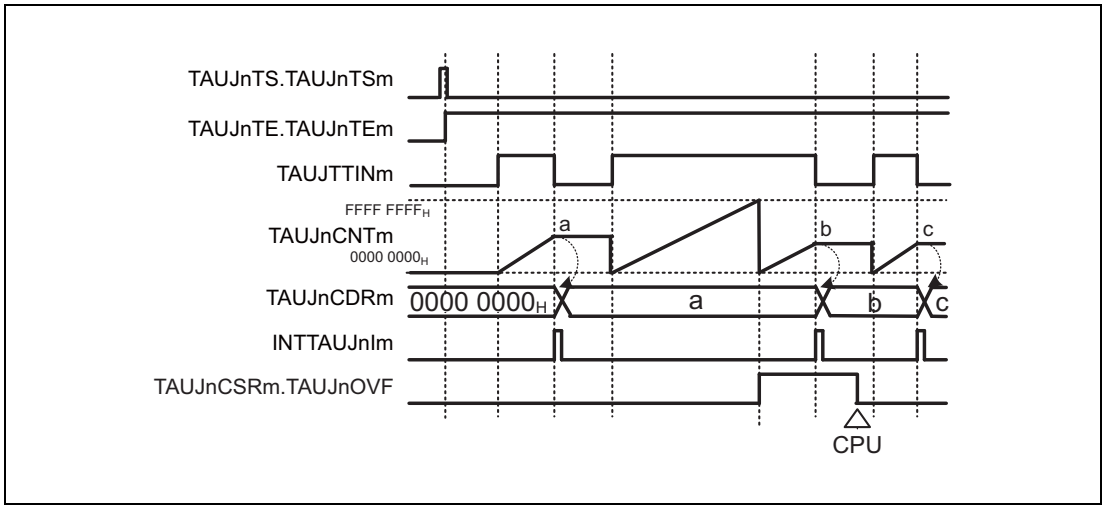


Figure 30.30 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and the value of TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (by setting the TAUJnCSCm.TAUJnCLOV bit to 1).

(3) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

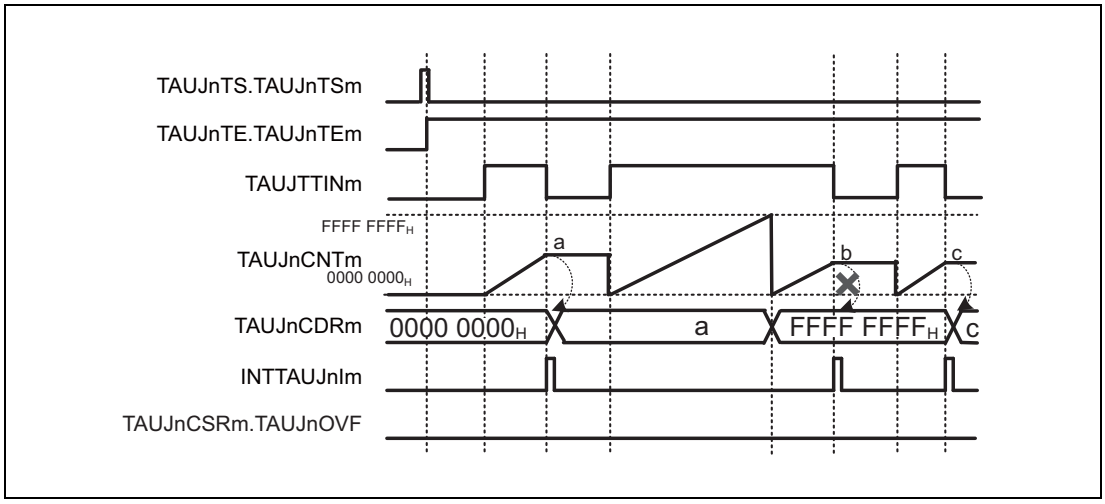


Figure 30.31 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$ and the value of TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm stops counting, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.

(4) TAUJnCMORm.TAUJnCOS[1:0] = 11_B

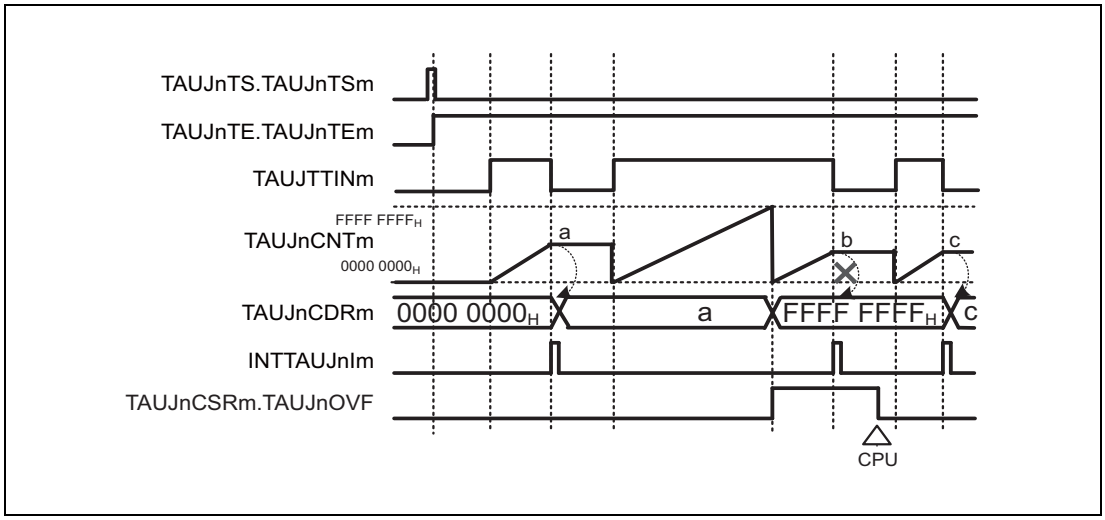


Figure 30.32 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJTTINm input edge, TAUJnCNTm stops counting, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJTTINm input valid edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting the TAUJnCSCm.TAUJnCLOV bit to 1.

30.12.5 TAUJTTINm Input Position Detection Function

30.12.5.1 Overview

Summary

This function measures the interval of an input signal by capturing the counter value on a valid edge of the TAUJTTINm signal.

Prerequisites

TAUJTTOUTm is not used for this function

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter starts to count from 0000 0000_H. When a valid TAUJTTINm input edge is detected, the current TAUJnCNTm value is loaded to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter continues to count.

When the counter reaches FFFF $FFFF_H$, the counter restarts from 0000 0000_H.

Conditions

If the TAUJnCMORm. TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated.

NOTE

The input TAUJTTINm is sampled at the frequency of the operation clock, specified by TAUJnCMORm.TAUJnCKS[1:0] bits. As a result, the output cycle of TAUJTTOUTm has an error of ± 1 operation clock cycle.

30.12.5.2 Equations

Function duration at a TAUJTTINm input pulse =
count clock cycle × (TAUJnCDRm capture value + 1)

30.12.5.3 Block Diagram and General Timing Diagram

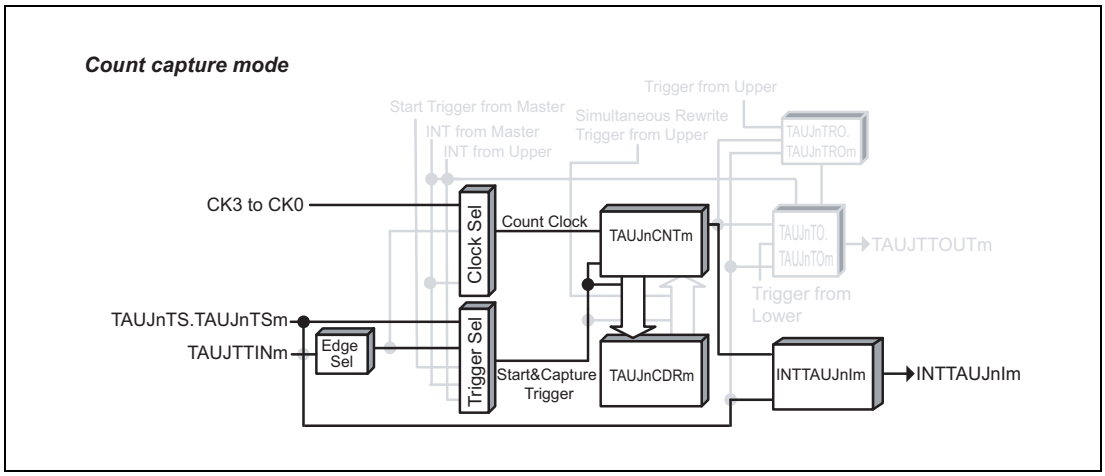


Figure 30.33 Block Diagram of TAUJTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- INTTAUJnIm is not generated when operation starts (TAUJnCMORm.TAUJnMD0 = 0).
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

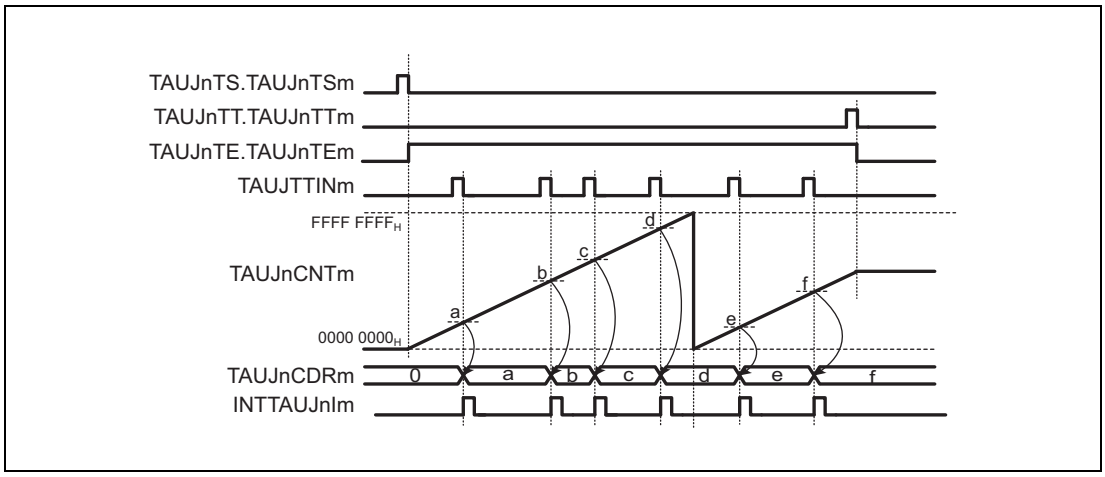


Figure 30.34 General Timing Diagram for TAUJTTINm Input Position Detection Function

30.12.5.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 30.53 Contents of the TAUJnCMORm Register for TAUJTTINm Input Position Detection Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 001_B.
7, 6	TAUJnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1011_B.
0	TAUJnMD0	0: INTTAUJnIm is not generated when operation starts. 1: Generates INTTAUJnIm when operation starts.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.54 Contents of the TAUJnCMURm Register for TAUJTTINm Input Position Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(3) Channel output mode

The channel output mode is not used by this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input position detection function. Therefore, these registers must be set to 0.

Table 30.55 Simultaneous Rewrite Settings for TAUJTTINm Input Position Detection Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0

30.12.5.5 Operating Procedure for TAUJTTINm Input Position Detection Function

Table 30.56 Operating Procedure for TAUJTTINm Input Position Detection Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in **Table 30.53, Contents of the TAUJnCMORm Register for TAUJTTINm Input Position Detection Function** and Table 30.54, Contents of the TAUJnCMURm Register for TAUJTTINm Input Position Detection Function. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. INTTAUJnIm is generated when TAUJnCMORm.TAUJnMD0 is set to 1.
During operation	The values of the TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time.	TAUJnCNTm starts to count up from 0000 0000_H. When a TAUJTTINm valid edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm. • INTTAUJnIm is output. • The counter value is not cleared to 0000 0000_H and TAUJnCNTm continues count operation. Afterwards, this procedure is repeated. When TAUJnCNTm reaches FFFF $FFFF_H$, the counter restarts from 0000 0000_H.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

30.12.5.6 Specific Timing Diagrams

(1) Operation stop and restart

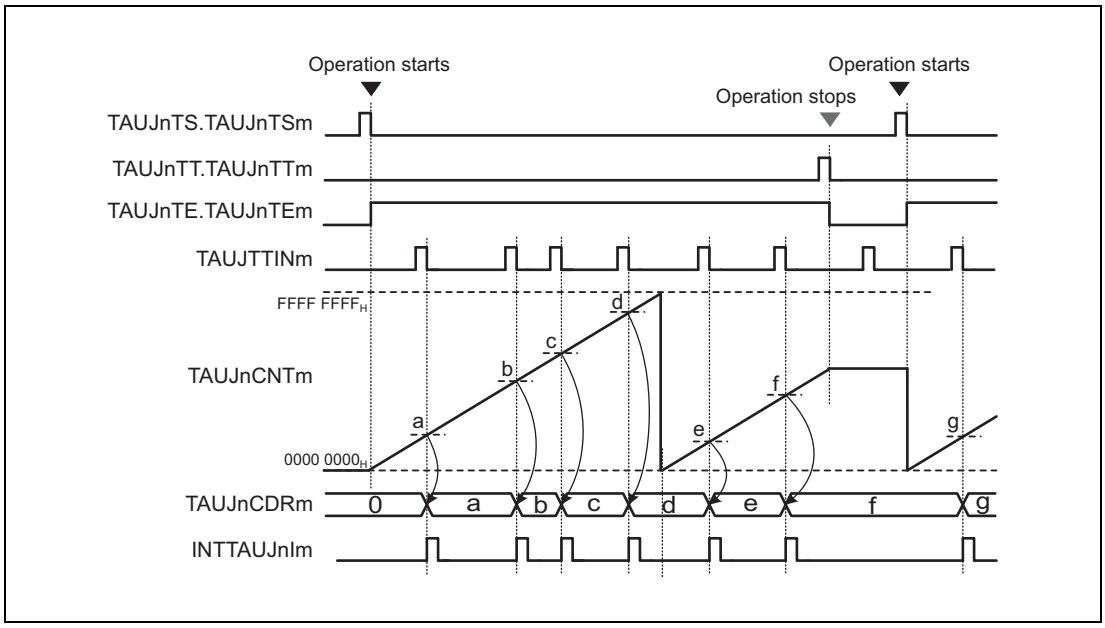


Figure 30.35 Operation Stop and Restart (TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts to count from $0000\ 0000_H$.

30.12.6 TAUJTTINm Input Period Count Detection Function

30.12.6.1 Overview

Summary

This function measures the cumulative width of a TAUJTTINm input signal.

Prerequisites

TAUJTTOUTm is not used for this function.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter awaits a valid TAUJTTINm input edge.

When a valid TAUJTTINm input start edge is detected, the counter starts to count from 0000 0000_H.

When a valid TAUJTTINm input stop edge is detected, the current TAUJnCNTm value is loaded to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter stops and retains its value until the next valid TAUJTTINm input start edge is detected.

When the next valid TAUJTTINm input start edge is detected, the counter restarts counting from the stop value.

When the counter reaches FFFF $FFFF_H$, the counter restarts from 0000 0000_H.

This function cannot be forcibly restarted.

NOTE

The input TAUJTTINm signal is sampled at the frequency of the operation clock, specified by the TAUJnCMORm.TAUJnCKS[1:0] bits.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11_B, the TAUJTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

30.12.6.2 Equations

Cumulative TAUJTTINm input width =
count clock cycle × (TAUJnCDRm capture value + 1)

30.12.6.3 Block Diagram and General Timing Diagram

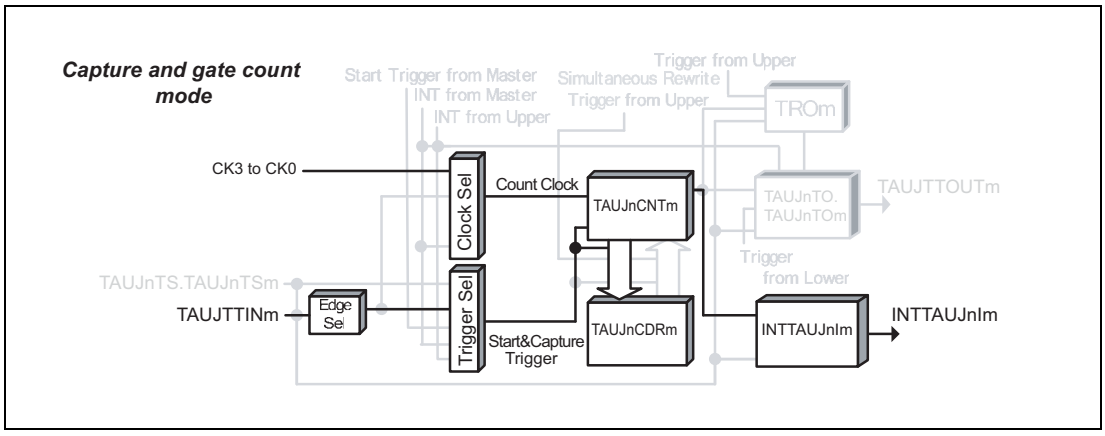


Figure 30.36 Block Diagram for TAUJTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

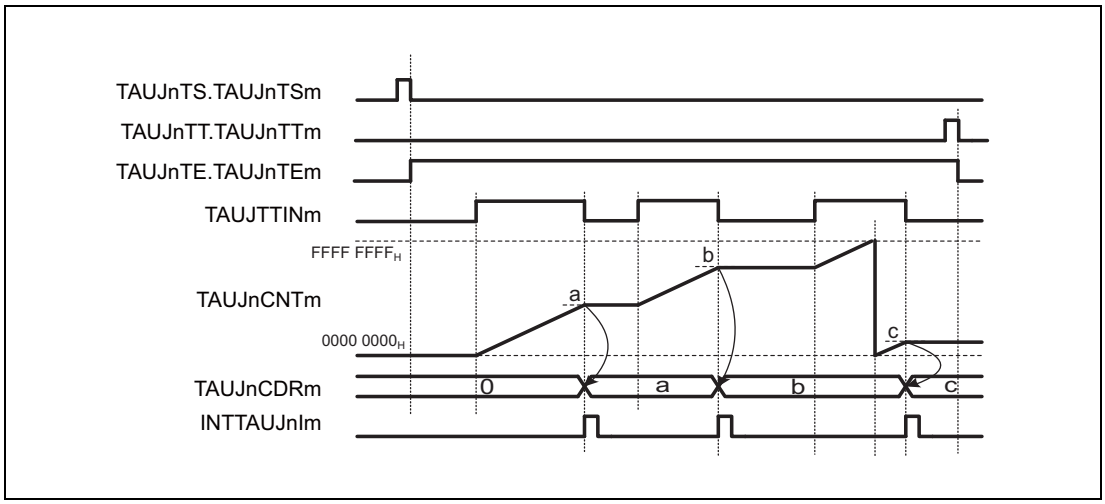


Figure 30.37 General Timing Diagram for TAUJTTINm Input Period Count Detection Function

30.12.6.4 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 30.57 Contents of the TAUJnCMORm Register for TAUJTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	Write 01_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1101_B.
0	TAUJnMD0	Write 0_B.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.58 Contents of the TAUJnCMURm Register for TAUJTTINm Input Period Count Detection Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJTTINm input period count detection function. Therefore, these registers must be set to 0.

Table 30.59 Simultaneous Rewrite Settings for TAUJTTINm Input Period Count Detection Function

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm=0), set these bits to 0

30.12.6.5 Operating Procedure for TAUJTTINm Input Period Count Detection Function

Table 30.60 Operating Procedure for TAUJTTINm Input Period Count Detection Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 30.57, Contents of the TAUJnCMORm Register for TAUJTTINm Input Period Count Detection Function** and **Table 30.58, Contents of the TAUJnCMURm Register for TAUJTTINm Input Period Count Detection Function**. The TAUJnCDRm register functions as a capture register.	Channel operation is stopped.
Start operation	Set TAUJTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJTTINm start edge.
During operation	TAUJTTINm edge detection The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time.	When a TAUJTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUJnCNTm starts to count up from the stop value. When TAUJnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUJnCDRm and INTTAUJnIm is generated. Counting stops at the “value transferred to TAUJnCDRm + 1” and TAUJnCNTm waits for detection of the TAUJTTINm start edge. When TAUJnCNTm reaches FFFF FFFF$_H$, the counter restarts from 0000 0000$_H$. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

30.12.6.6 Specific Timing Diagrams

(1) Operation Stop and Restart

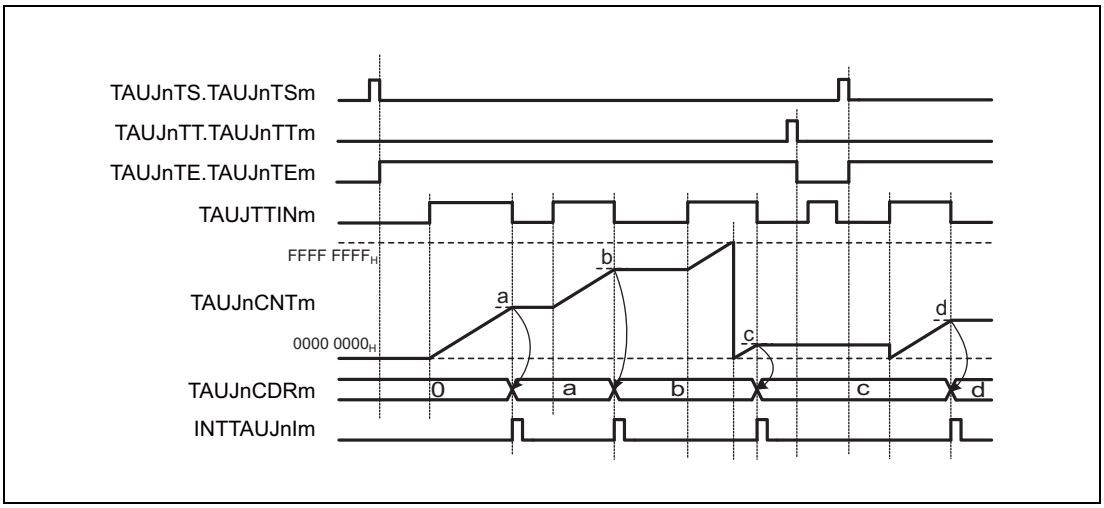


Figure 30.38 Operation Stop and Restart (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts to count from 0000 0000_H.

30.12.7 Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

30.12.7.1 Overview

Summary

This function measures the width of an individual TAUJTTINm input signal. An interrupt is generated if the TAUJTTINm input width is longer than FFFF $FFFF_H$ + 1.

Prerequisites

- TAUJTTOUTm is not used for this function.
- The value of TAUJnCDRm must be set to FFF $FFFF_H$.

Description

he counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation.

The counter starts when a valid TAUJTTINm input start edge is detected. FFFF $FFFF_H$ is loaded to TAUJnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUJTTINm input start edge is detected, TAUJnCNTm loads FFFF $FFFF_H$ and starts to count down.

If the counter reaches 0000 0000_H before a stop edge is detected, an interrupt is generated.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11_B, the TAUJTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

30.12.7.2 Block Diagram and General Timing Diagram

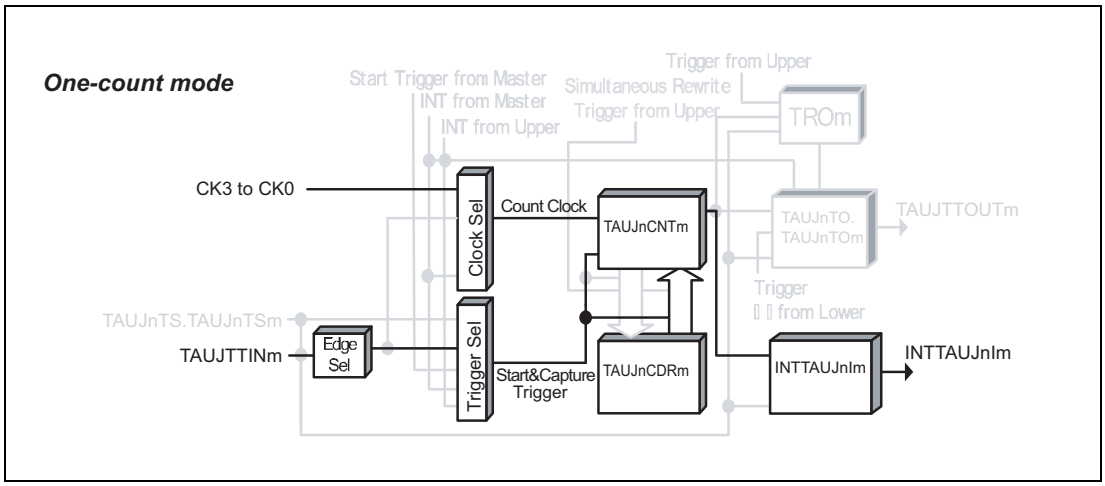


Figure 30.39 Block Diagram for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

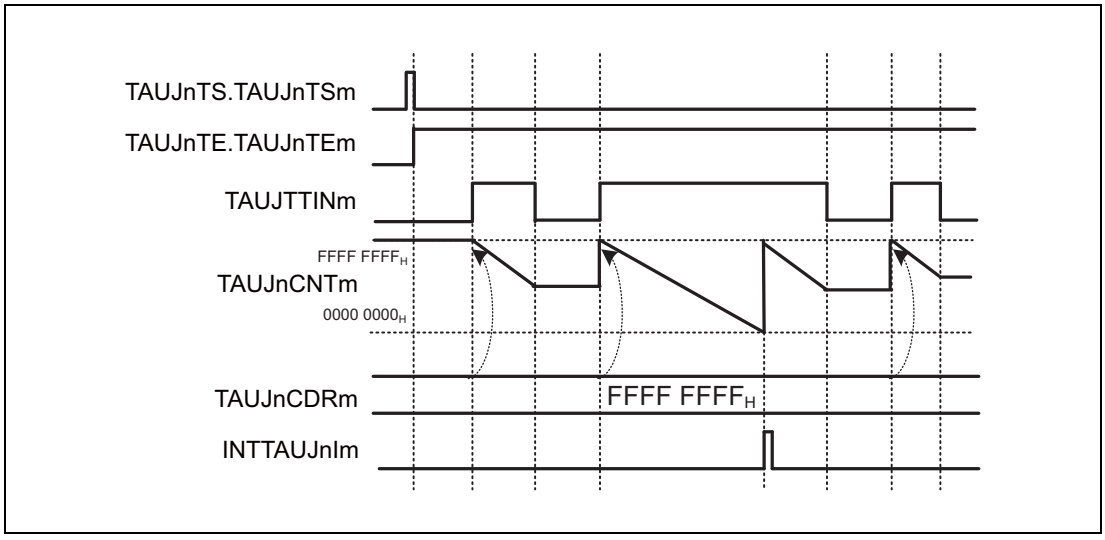


Figure 30.40 General Timing Diagram for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

30.12.7.3 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 30.61 Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0100_B.
0	TAUJnMD0	Write 0_B.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.62 Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the overflow interrupt output function (during TAUJTTINm width measurement). Therefore, these registers must be set to 0.

Table 30.63 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

30.12.7.4 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

Table 30.64 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in **Table 30.61, Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)** and **Table 30.62, Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Width Measurement)**. Set the value of the TAUJnCDRm register to FFFF FFFF$_H$.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUJTTINm start edge.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the start edge. When a start edge is detected, TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF$_H$).
During operation	The TAUJnCNTm register can be read at any time.	TAUJnCNTm counts down. When the counter reaches 0000 0000$_H$: • INTTAUJnIm is generated. When a reverse edge of TAUJTTINm is detected during count operation: • TAUJnCNTm stops counting and waits for a trigger. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

30.12.8 Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

30.12.8.1 Overview

Summary

This function measures the cumulative width of a TAUJTTINm input signal. An interrupt is generated if the cumulative TAUJTTINm input width is longer than FFFF FFFF$_H$, and an overflow interrupt can be output.

Prerequisites

- TAUJTTOUTm is not used for this function.
- The value of TAUJnCDRm must be set to FFFF FFFF$_H$.

Description

The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation.

The counter starts when a valid TAUJTTINm input start edge is detected. FFFF FFFF$_H$ is loaded to TAUJnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUJTTINm input start edge and then continues to count down from the current value.

When the counter reaches 0000 0000$_H$ an interrupt is generated. FFFF FFFF$_H$ is loaded to TAUJnCNTm and the counter continues to count down until a TAUJTTINm input stop edge is detected.

Conditions

The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits.

- If TAUJnCMURm.TAUJnTIS[1:0] = 10$_B$, the TAUJTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11$_B$, the TAUJTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

NOTE

The counter cannot be restarted during operation.

30.12.8.2 Block Diagram and General Timing Diagram

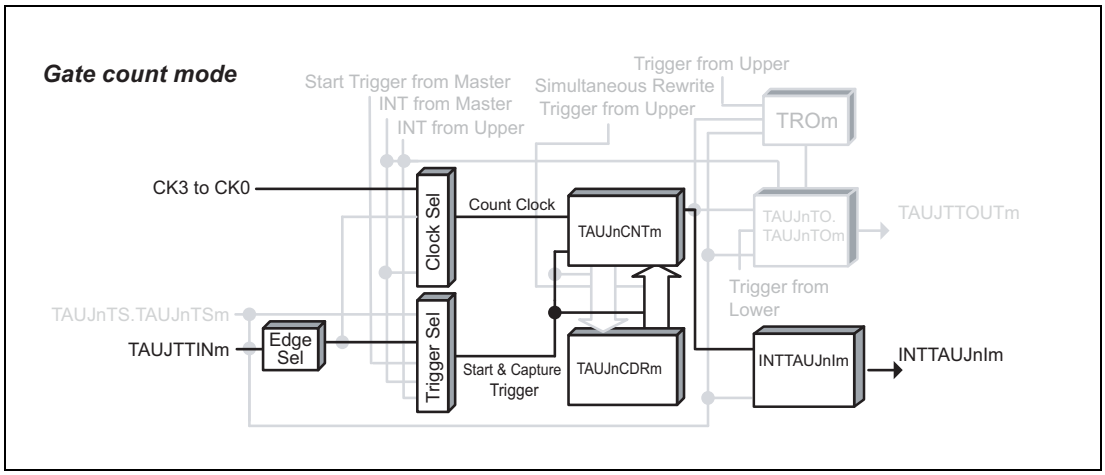


Figure 30.41 Block Diagram for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

The following settings apply to the general timing diagram.

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

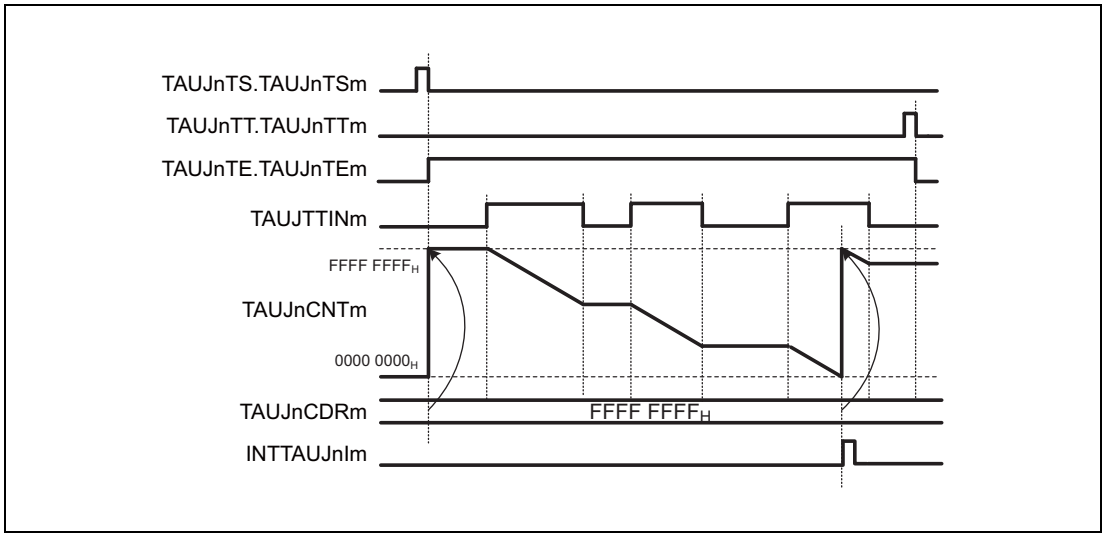


Figure 30.42 General Timing Diagram For Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

30.12.8.3 Register Settings

(1) TAUJnCMORm

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 30.65 Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 010_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 1100_B.
0	TAUJnMD0	Write 0_B.

(2) TAUJnCMURm

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.66 Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(3) Channel output mode

TAUJnTOE.TAUJnTOEm is set to 0 because the channel output mode is not used with this function.

(4) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the Overflow Interrupt Output Function (During TAUJTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 30.67 Simultaneous Rewrite Settings for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Bit Name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0.

30.12.8.4 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

Table 30.68 Operating Procedure for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in **Table 30.65, Contents of the TAUJnCMORm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)** and **Table 30.66, Contents of the TAUJnCMURm Register for Overflow Interrupt Output Function (during TAUJTTINm Input Period Count Detection)**. Set the value of the TAUJnCDRm register to FFFF FFFF$_H$.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUJTTINm start edge.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the start edge. When a start edge is detected, TAUJnCNTm the TAUJnCDRm value (FFFF FFFF$_H$).
During operation	The TAUJnCNTm register can be read at all times	TAUJnCNTm counts down. When the counter reaches 0000 0000$_H$: • INTTAUJnIm is generated. • TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF$_H$) and continues to count down. When a reverse edge of TAUJTTINm is detected during count operation: • TAUJnCNTm stops and retains the stop value. When a TAUJTTINm valid edge is detected while the counter is stopped: • TAUJnCNTm counts down from the stop value. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart operation

30.13 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by the TAUJ. For a general overview of synchronous channel operation, see Section 30.2, Overview.

30.13.1 PWM Output Function

30.13.1.1 Overview

Summary

This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty of the TAUJTTOUTm to be set. The pulse cycle is set in the master channel. The duty is set in the slave channel.

Prerequisites

- Two channels
- The operation mode for the master channel should be set to the interval timer mode. (See Table 30.69, Contents of the TAUJnCMORm Register for the Master Channel of the PWM Output Function.)
- The operation mode for the slave channel should be set to the one-count mode. (See **Table 30.72, Contents of the TAUJnCMORm Register for the Slave Channel of the PWM Output Function**.)
- TAUJTTOUTm is not used for the master channel of this function.
- The channel output mode for the slave channels should be set to synchronous channel output mode 1. (See 30.7, Channel Output Modes.)

Description

The counters are enabled by setting the channel trigger bits (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is loaded to TAUJnCNTm and the counters start to count down from these values. INTTAUJnIm is generated on the master channel and TAUJTTOUTm (slave) is set or reset to realize the PWM output.

- Master channel:

  When the counter of the master channel reaches 0000 0000_H and pulse cycle time has elapsed, INTTAUJnIm is generated. The TAUJnCDRm value is loaded to TAUJnCNTm, and the counter counts down.

- Slave channel(s):

  The INTTAUJnIm of the master channel triggers the counter of the slave channel(s). The current value of TAUJnCDRm (slave) is loaded to TAUJnCNTm (slave) and the counter starts to count down from this value. The TAUJTTOUTm signal is set, to the active level.
  When the counter reaches 0000 0000_H, i.e. duty time has elapsed, INTTAUJnIm is generated and the TAUJTTOUTm signal is set to the inactive level. The counter returns to FFFF $FFFF_H$ and awaits the next INTTAUJnIm of the master channel, and thus the start of the next pulse cycle.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1 for the master and slave channel(s), which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJTTOUTm of master and slave

channel(s) stop but retain their values. The counters can be restarted by setting TAUJnTS.TAUJnTSm to 1.

Conditions

Simultaneous rewrite can be used with this function. Please refer to Section 30.6, Simultaneous Rewrite.

30.13.1.2 Equations

Pulse cycle = (TAUJnCDRm (master) + 1) × count clock cycle

Duty cycle [%] = (TAUJnCDRm (slave)/(TAUJnCDRm (master) +1)) × 100

– Duty cycle = 0%
  TAUJnCDRm (slave) = 0000 0000_H
– Duty cycle = 100%
  TAUJnCDRm (slave) ≥ TAUJnCDRm (master) + 1

30.13.1.3 Block Diagram and General Timing Diagram

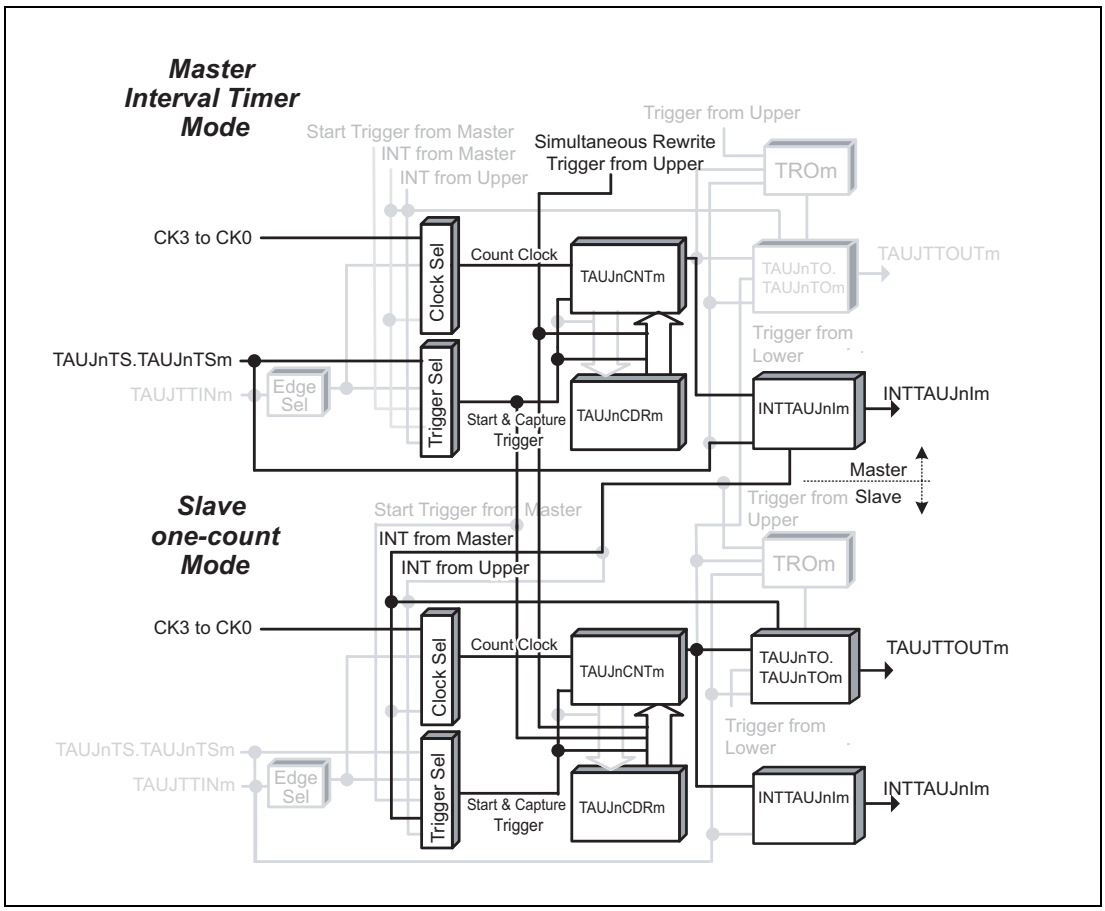


Figure 30.43 Block Diagram for PWM Output Function

The following settings apply to the general timing diagram.

- Slave channel: Positive logic (TAUJnTOL.TAUJnTOLm = 0)

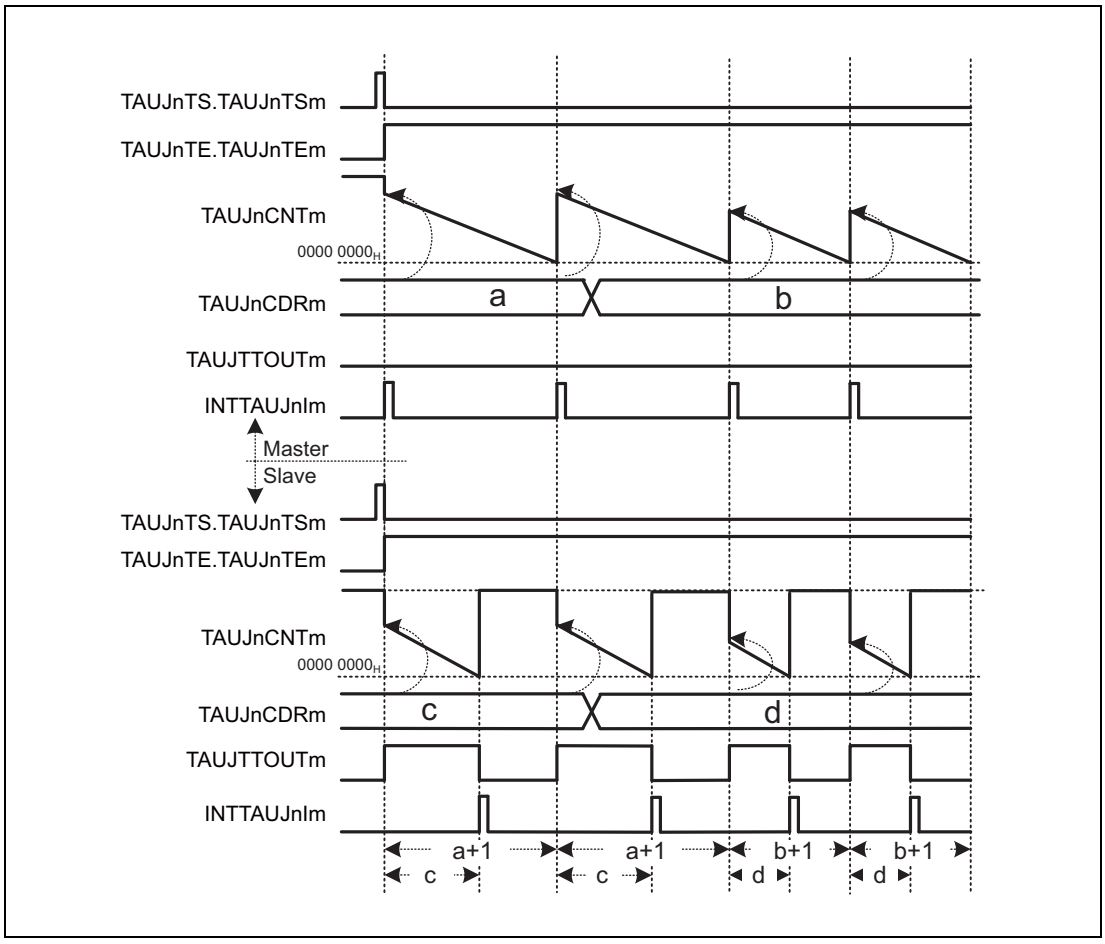


Figure 30.44 General Timing Diagram for PWM Output Function

NOTE

- The interval between the slave channel starting to count and an interrupt being generated is the value of corresponding TAUJnCDRm + 1.
- The slave channel TAUJTTOUTm will rise with a delay of one count clock after the rising of the master channel INTTAUJnIm.

30.13.1.4 Register Settings for the Master Channel

(1) TAUJnCMORm for the master channel

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 30.69 Contents of the TAUJnCMORm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bits of the master and slave channel(s) must be identical.
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 1_B.
10 to 8	TAUJnSTS[2:0]	Write 000_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0000_B.
0	TAUJnMD0	Write 1_B.

(2) TAUJnCMURm for the master channel

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.70 Contents of the TAUJnCMURm Register for the Master Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the master channel

The channel output mode is not used by this function.

(4) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 30.71 Simultaneous Rewrite Settings for the Master Channel of the PWM Output Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting.

30.13.1.5 Register Settings for the Slave Channel(s)

(1) TAUJnCMORm for the slave channel(s)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TAUJnCKS[1:0]		TAUJnCCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJnCOS[1:0]		—	TAUJnMD[4:1]				TAUJn MD0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 30.72 Contents of the TAUJnCMORm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
15, 14	TAUJnCKS[1:0]	Operation Clock Selection 00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bits of the master and slave channel(s) must be identical.
13, 12	TAUJnCCS[1:0]	Write 00_B.
11	TAUJnMAS	Write 0_B.
10 to 8	TAUJnSTS[2:0]	Write 100_B.
7, 6	TAUJnCOS[1:0]	Write 00_B.
5	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
4 to 1	TAUJnMD[4:1]	Write 0100_B.
0	TAUJnMD0	Write 1_B.

(2) TAUJnCMURm for the slave channel(s)

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	TAUJnTIS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 30.73 Contents of the TAUJnCMURm Register for the Slave Channel of the PWM Output Function

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	TAUJnTIS[1:0]	00: Not used, so set to 00.

(3) Channel output mode for the slave channel(s)

Table 30.74 Control Bit Settings for Synchronous Channel Output Mode 1

Bit Name	Setting
TAUJnTOE.TAUJnTOEm	Write 1_B.
TAUJnTOM.TAUJnTOMm	Write 1_B.
TAUJnTOC.TAUJnTOCm	Write 0_B.
TAUJnTOL.TAUJnTOLm	0: Positive logic 1: Negative logic

(4) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 30.75 Simultaneous Rewrite Settings for the Slave Channel of the PWM Output Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite.
TAUJnRDM.TAUJnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting.

30.13.1.6 Operating Procedure for PWM Output Function

Table 30.76 Operating Procedure for PWM Output Function

	Operation	Status of TAUJn
Initial channel setting	Master channel: Set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in **Section 30.13.1.4, Register Settings for the Master Channel**. Slave channel: Set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in **Section 30.13.1.5, Register Settings for the Slave Channel(s)**. Set the values of the TAUJnCDRm registers of all channels.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm of the master and slave channels to 1 simultaneously. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUJnIm is generated on the master channel and TAUJTTOUTm (slave) is set.
During operation	TAUJnCDRm can be changed at any time. TAUJnCNTm and TAUJnRSF.TAUJnRSFm can be read at any time. TAUJnRDT.TAUJnRDTm can be changed during operation.	TAUJnCNTm of the master channel loads TAUJnCDRm and counts down. When the counter reaches 0000 0000_H: • INTTAUJnIm (master) is generated. • TAUJnCNTm (master) loads the TAUJnCDRm value and continues count operation. • TAUJnCNTm (slave) loads the TAUJnCDRm value and counts down. • TAUJTTOUTm (slave) is set to the active level. When TAUJnCNTm (slave) reaches 0000 0000_H: • INTTAUJnIm (slave) is generated. • TAUJTTOUTm (slave) is set to the inactive level.
Stop operation	Set TAUJnTT.TAUJnTTm of the master and slave channels to 1 simultaneously. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJTTOUTm stop and retain their current values.

Restart operation

30.13.1.7 Specific Timing Diagrams

(1) Duty cycle = 0%

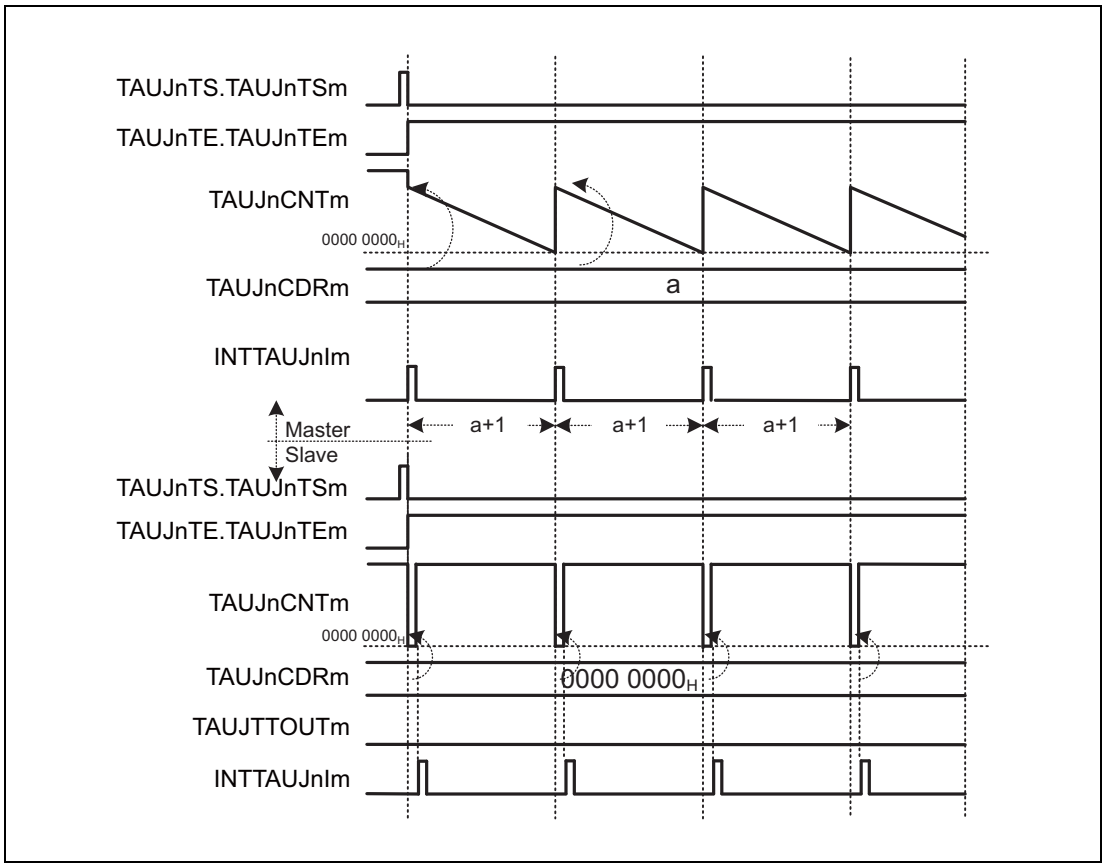


Figure 30.45 TAUJnCDRm (slave) = 0000 0000$_H$, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

Every time the master channel generates an interrupt (INTTAUJnIm), 0000 0000$_H$ is loaded to TAUJnCNTm (slave). As a result, a slave channel interrupt (INTTAUJnIm) is generated at the same time and TAUJTTOUTm remains inactive.

- The value of TAUJnCDRm is loaded into TAUJnCNTm (slave) to generate an interrupt.

(2) Duty cycle = 100%

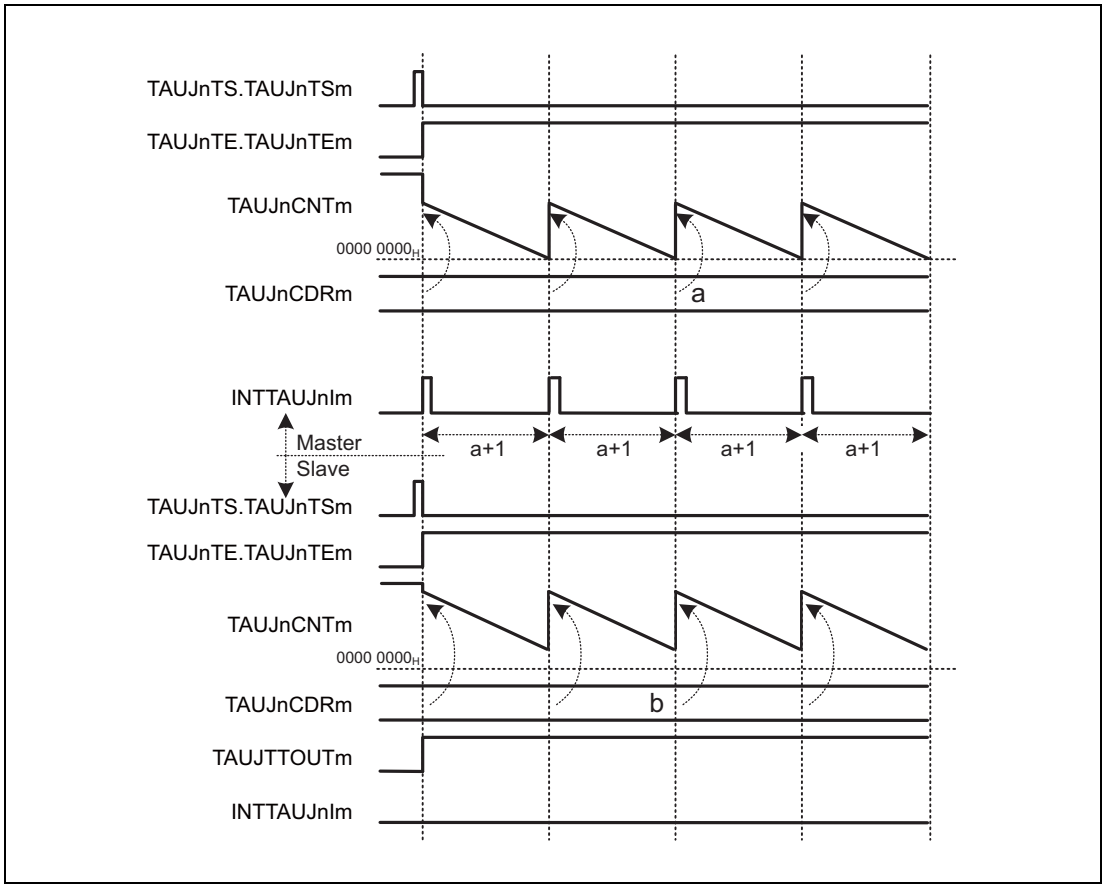


Figure 30.46 TAUJnCDRm (slave) ≥ TAUJnCDRm (master) + 1, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

If the TAUJnCDRm (slave) value is greater than the TAUJnCDRm (master) value, no interrupt occurs because the counter of the slave channel does not reach $0000\ 0000_H$. TAUJTTOUTm remains active.

(3) Operation stop and restart

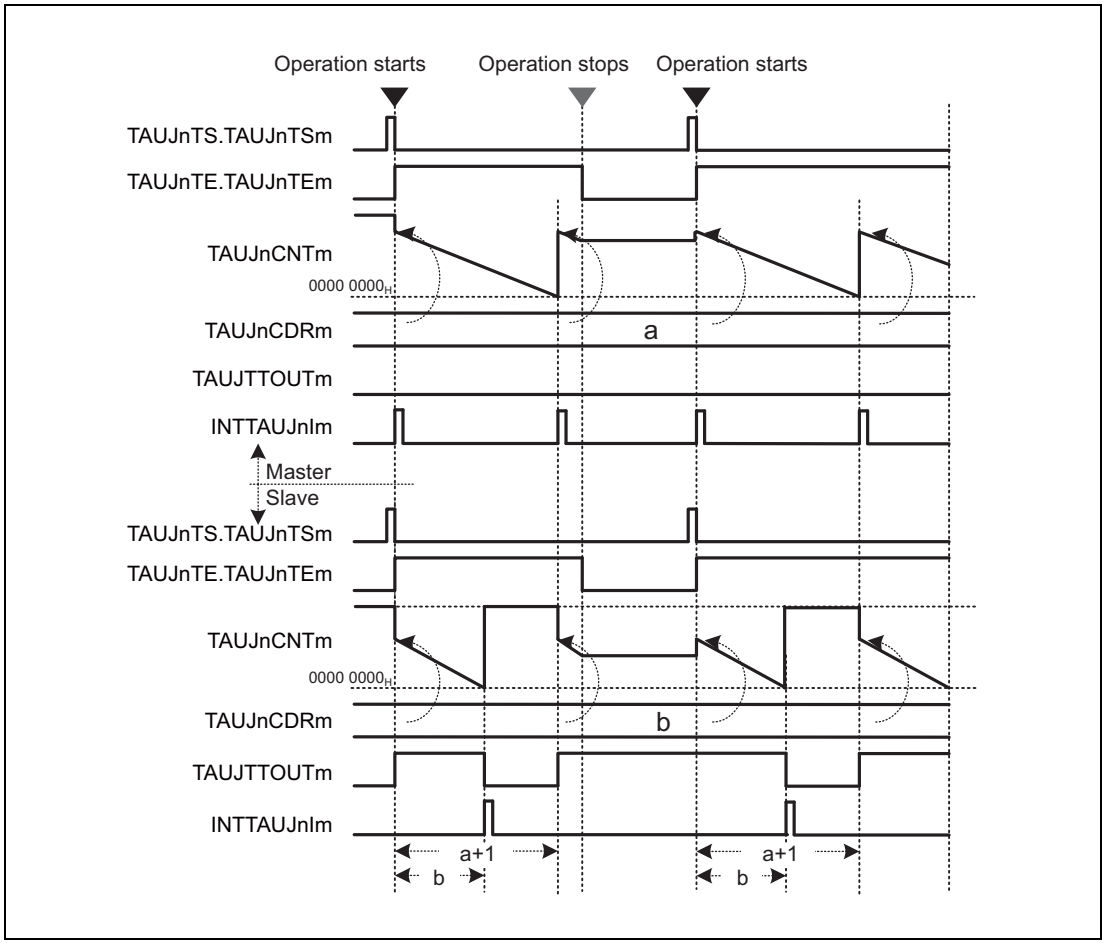


Figure 30.47 Stop and Restart Operation, Positive Logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm of master and slave channels to 1. This sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJTTOUTm of every channel stop and retain their current values. No interrupt occurs.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm of master and slave channels to 1. The TAUJnCDRm value of master and slave channels is loaded into TAUJnCNTm. The counter starts to count down from this value.

Section 31 Real-Time Clock (RTCA)

This section contains a generic description of the Real-Time Clock (RTCA).

The first part of this section describes all RH850/D1L/D1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the RTCA.

31.1 Features of RH850/D1L/D1M RTCA

31.1.1 Number of Units and Channels

This microcontroller has the following number of RTCA units.

Each RTCA unit has one channel RTCA. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 31.1 Number of Units

Product Name	All Products
Number of Units	1
Name	RTCAn (n = 0)

Table 31.2 Index

Index	Meaning
n	Throughout this section, the individual RTCA units are identified by the index “n” (n = 0); for example, RTCAnCTL0 is the RTCAn control register 0.

31.1.2 Register Base Address

RTCAn base address is listed in the following table.

RTCAn register addresses are given as offsets from the base address in general.

Table 31.3 Register Base Address

Base Address Name	Base Address
<RTCA0_base>	FFE4 1000_H

31.1.3 Clock Supply

The RTCAn clock supply is shown in the following table.

Table 31.4 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
RTCA0	RTCATCKI	C_AWO_RTCA
	PCLK	• C_ISO_PCLK in RUN mode • EMCLK in DEEPSTOP mode

31.1.4 Interrupt Requests

RTCAn interrupt requests are listed in the following table.

Table 31.5 Interrupt Requests

RTCAn signals	Function	Connected to
RTCA0		
RTCATINT1S	1 second interval interrupt	Interrupt Controller INTRTCA01S
RTCATINTAL	Alarm interrupt	Interrupt Controller INTRTCA0AL
RTCATINTR	Fixed interval interrupt	Interrupt Controller INTRTCA0R

All Real-Time Clock interrupts can be used as wake-up factors for terminating the DEEPSTOP mode.

NOTE

In all devices except for D1M1A and D1M1-V2, when CPU is operating under EMCLK, there are possibility to miss interrupts' request of INTRTCA01S, INTRTCA0AL and INTRTCA0R. Use wake-up flags instead of interrupt flags, when CPU is running by EMCLK. (Wake-up factors for those interrupts (WUF0[13], WUF0[14], WUF0[15]) are not missed.)

31.1.5 Reset Sources

RTCAn reset sources are listed in the following table. RTCAn is initialized by these reset sources.

Table 31.6 Reset Sources

Unit Name	Reset Source
RTCA0	Reset Controller SYSRES

31.1.6 External Input/Output Signals

External input/output signals of RTCAn are listed below.

Table 31.7 External Input/Output Signals

Unit Signal Name	Outline	Alternative Port Pin Signal
RTCA0		
RTCAT1HZ	1-Hz pulse output	RTCA0OUT*1

Note 1. RTCA0OUT is connected to AWOT0. For details, see Section 28, Always-On-Area Timer (AWOT).

31.2 Overview

31.2.1 Functional Overview

The Real-Time Clock RTCA has the following features:

- Count clock selection from 32 kHz to 4 MHz
- Counters for years, months, day of the month, day of the week, hours, minutes, seconds, and a sub-counter. The calendar covers 99 years. Leap years are handled by hardware automatically.
- One Hz pulse output function
- Fixed interval interrupt function
- Alarm interrupt function
- Clock error correction function if a 32.768-kHz count clock is used

31.2.2 Block Diagram

The block diagram shows the main components of the RTCA.

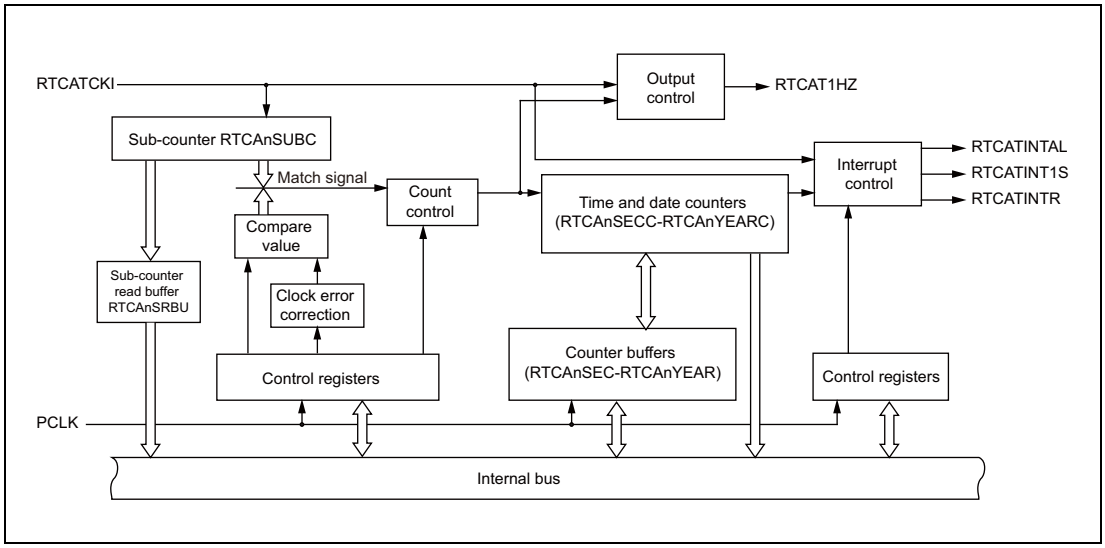


Figure 31.1 Block Diagram of the RTCA

31.2.3 Description of Blocks

The Real-Time Clock RTCA provides information about the present time and date and can generate wake-up signals (interrupts, alarms). This information is derived from the count clock RTCATCKI.

Sub-counter

RTCATCKI is the input to the sub-counter RTCAnSUBC. The sub-counter counts up from zero until it reaches the compare value. The compare value is always defined as the frequency of RTCATCKI – 1 (in Hz). Thus, the sub-counter overflows after one second. It is then reset to zero and triggers the seconds counter RTCAnSECC (and, if desired, the interrupt RTCATINT1S).

The sub-counter can generate a fixed interval interrupt every 0.25 seconds, 0.5seconds, or 1 second, and a 1-Hz output pulse.

Time and date counters

The counters for minutes, hours, day of the week, day of the month, months, and years also count up. They have their own overflow limits. If all the lower counters overflow, the upper counter counts up.

The overflow limit of the counter for the day of the month (RTCAnDAYC) depends on the present month (28, 30, or 31 days) and (in February) on the year counter RTCAnYEARC (years 0, 4, 8, 12, etc. are considered leap years).

The hours counter RTCAnHOURC can be switched between 12- and 24-hour format.

The counters for seconds, minutes, hours, day of the month, and month can generate a fixed interval interrupt upon overflow (RTCATINTR).

The counters for minutes, hours, and day of the week can also generate an alarm interrupt (RTCATINTAL), e.g. every Tuesday and Thursday at 10:32.

Counter buffers

All counters can be read directly at any time. The clock signal used to access the read/write registers and the count clock are usually asynchronous. An overflow of the sub-counter during the read operation can make all read values obsolete. Therefore, reading the counters must be performed using a special procedure. For details, see Section 31.5.3, Reading Clock Counters.

For reasons of synchronization, the counters cannot be written directly.

For reading and writing, all counters are accompanied by buffer registers. The buffer registers provide a synchronized way for reading the counters and for setting time and date. When they are used, the operation of the sub-counter must first be suspended and then re-activated (see also **Section 31.5.3, Reading Clock Counters** and Section 31.5.2, Updating Clock Counters.

The RTCAnTIMEC and RTCAnCALC registers and their corresponding buffer registers can be used to access the time (hours, minutes and seconds) or the date (day of the week, day of the month, month, and year) with one read/write operation.

31.3 Registers

31.3.1 List of Registers

RTCA registers are listed in the following table.

<RTCAn_base> is defined in Section 31.1.2, Register Base Address.

Table 31.8 Registers

Module	Register	Symbol	Address
Control registers			
RTCAn	Control register 0	RTCAnCTL0	<RTCAn_base> + 00_H
RTCAn	Control register 1	RTCAnCTL1	<RTCAn_base> + 04_H
RTCAn	Control register 2	RTCAnCTL2	<RTCAn_base> + 08_H
Sub-counter registers			
RTCAn	Sub-count register	RTCAnSUBC	<RTCAn_base> + $0C_H$
RTCAn	Sub-count register read buffer	RTCAnSRBU	<RTCAn_base> + 10_H
RTCAn	Clock error correction register	RTCAnSUBU	<RTCAn_base> + 38_H
RTCAn	Sub-counter compare register	RTCAnSCMP	<RTCAn_base> + $3C_H$
Clock counter and buffer registers			
RTCAn	Seconds count register	RTCAnSECC	<RTCAn_base> + $4C_H$
RTCAn	Seconds count buffer register	RTCAnSEC	<RTCAn_base> + 14_H
RTCAn	Minute count register	RTCAnMINC	<RTCAn_base> + 50_H
RTCAn	Minute count buffer register	RTCAnMIN	<RTCAn_base> + 18_H
RTCAn	Hour count register	RTCAnHOURC	<RTCAn_base> + 54_H
RTCAn	Hour count buffer register	RTCAnHOUR	<RTCAn_base> + $1C_H$
RTCAn	Day of the week count register	RTCAnWEEKC	<RTCAn_base> + 58_H
RTCAn	Day of the week count buffer register	RTCAnWEEK	<RTCAn_base> + 20_H
RTCAn	Day count register	RTCAnDAYC	<RTCAn_base> + $5C_H$
RTCAn	Day count buffer register	RTCAnDAY	<RTCAn_base> + 24_H
RTCAn	Month count register	RTCAnMONC	<RTCAn_base> + 60_H
RTCAn	Month count buffer register	RTCAnMONTH	<RTCAn_base> + 28_H
RTCAn	Year count register	RTCAnYEARC	<RTCAn_base> + 64_H
RTCAn	Year count buffer register	RTCAnYEAR	<RTCAn_base> + $2C_H$
Special counter and buffer registers			
RTCAn	Time count register	RTCAnTIMEC	<RTCAn_base> + 68_H
RTCAn	Time count buffer register	RTCAnTIME	<RTCAn_base> + 30_H
RTCAn	Calendar count register	RTCAnCALC	<RTCAn_base> + $6C_H$
RTCAn	Calendar count buffer register	RTCAnCAL	<RTCAn_base> + 34_H
Alarm setting registers			
RTCAn	Alarm minute setting register	RTCAnALM	<RTCAn_base> + 40_H
RTCAn	Alarm hour setting register	RTCAnALH	<RTCAn_base> + 44_H
RTCAn	Alarm day of the week setting register	RTCAnALW	<RTCAn_base> + 48_H
Emulation register			
RTCAn	Emulation register	RTCAnEMU	<RTCAn_base> + 74_H

31.3.2 Details of RTCA Control Registers

31.3.2.1 RTCAnCTL0 — RTCA Control Register 0

This register controls the count operation of the sub-counter RTCAnSUBC, the format (12-hour/24-hour) of the hours counter RTCAnHOURC and the alarm hour setting register RTCAnALH, and the operation mode.

Access: This register can be read/written in 8-bit or 1-bit units.

Address: <RTCAn_base> + 00_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnCE	RTCAnCEST	RTCAnAMPM	RTCAnSLSB	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R/W	R/W	R	R	R	R

Table 31.9 RTCAnCTL0 Register Contents

Bit Position	Bit Name	Function
7	RTCAnCE	Starts/stops the sub-counter RTCAnSUBC operation. 0: Stops the sub-counter operation. All output pins and all status flags in control register RTCAnCTL2 are cleared. 1: Starts the sub-counter operation. The sub-counter counts up.
6	RTCAnCEST	Indicates the operation enabled/stopped status of the sub-counter: 0: Operation stopped status 1: Operation enabled status For details on how to use this status flag, see Section 31.5.1, Initial Setting of the RTCA.
5	RTCAnAMPM	Selects the format of the hours counter RTCAnHOURC and the alarm hour setting register RTCAnALH: 0: 12-hour format (1 to 12, am/pm) 1: 24-hour format (0 to 23, military time) For details on the format, see Table 31.21, 12- and 24-Hour Format.
4	RTCAnSLSB	Selects the operation mode: 0: 32.768 kHz mode 1: Frequency selection mode For details on the operation modes, see Section 31.4, Operation. The operation mode must not be changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1). For details on the initialization of RTCAn, see Section 31.5.1, Initial Setting of the RTCA.
3 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

31.3.2.2 RTCAnCTL1 — RTCA Control Register 1

This register controls the interrupt request generation and the 1-Hz pulse output.

Access: This register can be read/written in 8-bit or 1-bit units.

Address: <RTCAn_base> + 04_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnEN1HZ	RTCAnENALM	RTCAnEN1S	RTCAnCT2	RTCAnCT1	RTCAnCT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.10 RTCAnCTL1 Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	RTCAnEN1HZ	Enables/stops 1-Hz pulse output (RTCAT1HZ): 0: RTCAT1HZ disabled (RTCAT1HZ is fixed to 0) 1: RTCAT1HZ enabled
4	RTCAnENALM	Enables/disables alarm interrupt request generation (RTCATINTAL): 0: RTCATINTAL disabled 1: RTCATINTAL enabled
3	RTCAnEN1S	Enables/disables 1-second interrupt request generation (RTCATINT1S): 0: RTCATINT1S disabled 1: RTCATINT1S enabled
2 to 0	RTCAnCT[2:0]	Specifies the fixed interval interrupt request (RTCATINTR) setting:

	RTCATINTR Interrupt Request Generation	
RTCAnCT[2:0]	**Interval**	**Timing**
000	No interrupt request generation	
001	Every 0.25 seconds	Every 0.25, 0.5, 0.75 and 1 second
010	Every 0.5 seconds	Every 0.5 and 1 second
011	Every second	Every 1 second
100	Every minute	Every 1 minute 00 seconds
101	Every hour	Every 1 hour 0 minutes 0 seconds
110	Every day	Every 1 day 0 hours 0 minutes 0 seconds (i.e., every midnight)
111	Every month	Every 1 month first day 0 hours 0 minutes 0 seconds (i.e., every first midnight of a month)

If the settings of RTCAnCT[2:0] are changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), a glitch may be output to RTCATINTR. Implement appropriate interrupt mask processing procedures.

31.3.2.3 RTCAnCTL2 — RTCA Control Register 2

This register contains status information and controls the data transfer from the sub-counter RTCAnSUBC to the dedicated sub-counter read buffer RTCAnSRBU and the operation setting of the clock counters (RTCAnSECC to RTCAnYEARC).

Access: This register can be read/written in 8-bit or 1-bit units.

Address: <RTCAn_base> + 08_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnWUST	RTCAnWSST	RTCAnRSST	RTCAnRSUB	RTCAnWST	RTCAnWAIT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R	R/W

Table 31.11 RTCAnCTL2 Register Contents (1/2)

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5	RTCAnWUST	Indicates whether RTCAnSUBU write operation has been completed: 0: RTCAnSUBU write completed 1: RTCAnSUBU write in progress The write operation ends with the next sub-counter overflow. While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1) and if write operation to RTCAnSUBU is completed, this bit is set to 1. See Section 31.5.5, Writing to RTCAnSUBU, for details.
4	RTCAnWSST	Indicates whether RTCAnSCMP write operation has been completed: 0: RTCAnSCMP write completed 1: RTCAnSCMP write in progress The write operation ends with the next sub-counter overflow. While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1) and if write operation to RTCAnSCMP is completed, this bit is set to 1. See Section 31.5.6, Writing to RTCAnSCMP, for details.
3	RTCAnRSST	Indicates whether the value of the sub-counter (RTCAnSUBC) has been transferred to the sub-count register read buffer (RTCAnSRBU): 0: Transfer in progress, or waiting for a transfer trigger 1: Transfer completed This bit is cleared (transfer is triggered) by RTCAnRSUB=1.This bit is automatically set when the transfer is completed. See Section 31.5.4, Reading RTCAnSRBU, for details.
2	RTCAnRSUB	Triggers transfer of the value of the sub-counter (RTCAnSUBC) to the dedicated read buffer (RTCAnSRBU) or clears the transfer state of the sub-counter: 0: Transfer status (RTCAnRSST) is cleared. 1: Transfer is triggered. This bit is used to read the value of RTCAnSRBU when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1). The value of RTCAnSUBC is synchronized with RTCATCKI and loaded to RTCAnSRBU. For details, see Section 31.5.4, Reading RTCAnSRBU.

Table 31.11 RTCAnCTL2 Register Contents (2/2)

Bit Position	Bit Name	Function
1	RTCAnWST	Indicates the status of all clock counters (RTCAnSECC to RTCAnYEARC): 0: All clock counters are running. 1: All clock counters are stopped The sub-counter is still running. The clock counters must be stopped before reading or writing clock counter values during sub-counter operation (RTCAnCTL0.RTCAnCE = 1). To stop the clock counters, set RTCAnWAIT = 1.
0	RTCAnWAIT	Restarts/stops all clock counters (RTCAnSECC to RTCAnYEARC): 0: Restarts all clock counters either immediately or immediately after the clock counter write operation finishes. 1: Stops all clock counters temporarily. The sub-counter is still running. The clock counters must be stopped before reading or writing counter buffers during sub-counter operation (RTCAnCTL0.RTCAnCE = 1). **CAUTION** Only one overflow can be held internally. When two overflows occur, the seconds counter is incremented only by one when it is restarted. Thus, the procedure must be completed within one second.

31.3.3 Details of RTCA Sub-Counter Registers

31.3.3.1 RTCAnSUBC — RTCA Sub-Count Register

This counter counts the 1-second reference time. It operates using the count clock RTCATCKI.

Access: This register can only be read in 32-bit units.

Address: <RTCAn_base> + $0C_H$

Value after reset: $0000\ 0000_H$

This register is initialized:

– When write operation is performed to the seconds count buffer register (RTCAnSEC) or to the time count buffer register (RTCAnTIME) and the value is reflected to the seconds count register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnSUBC[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAnSUBC[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.12 RTCAnSUBC Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 0	RTCAnSUBC [21:0]	Sub-counter value The sub-counter only operates while RTCAnCTL0.RTCAnCEST = 1.

NOTES

1. This sub-counter operates with RTCATCKI while the read operation is clocked by PCLK. Reading this sub-counter during operation (RTCAnCTL0.RTCAnCEST = 1) is asynchronous to RTCATCKI and can lead to wrong results.
   Use the sub-count register read buffer (RTCAnSRBU) to read the sub-counter value during operation.
   For details, see Section 31.5.4, Reading RTCAnSRBU.
2. The count-operation of this sub-counter depends on the selected operation mode. See Section 31.4, Operation, for details.

31.3.3.2 RTCAnSRBU — RTCA Sub-Count Register Read Buffer

This register is the read buffer for the sub-counter RTCAnSUBC.

Access: This register can only be read in 32-bit units.

Address: <RTCAn_base> + 10_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnSRBU[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAnSRBU[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.13 RTCAnSRBU Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 0	RTCAnSRBU [21:0]	Sub-counter value at the time of the last RTCAnSUBC read. When RTCAnCTL2.RTCAnRSUB is set to 1, the value of the RTCAnSUBC is loaded to the read buffer in synchronization with RTCATCKI.

NOTE

Perform RTCAnSRBU read according to the flow described in Section 31.5.4, Reading RTCAnSRBU.

31.3.3.3 RTCAnSUBU — RTCA Clock Error Correction Register

This register enables and specifies clock error correction. This register only applies in 32.768-kHz mode (RTCAnCTL0.RTCAnSLSB = 0).

For details on clock error correction, see Section 31.4.4, Clock Error Correction.

Access: This register can be read/written in 8-bit units.
Note the following when writing this register during sub-counter operation:
- Previous RTCAnSUBU write must be completed (RTCAnCTL2.RTCAnWUST = 0).
- The write operation ends with the next sub-counter overflow.

Address: <RTCAn_base> + 38_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnDEV	RTCAnF6	RTCAnF[5:0]					
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.14 RTCAnSUBU Register Contents

Bit Position	Bit Name	Function
7	RTCAnDEV	Specifies how often clock error correction is performed per minute: 0: Three times every minute (when RTCAnSECC equals 00, 20, and 40) 1: Once every minute (when RTCAnSECC equals 00)
6	RTCAnF6	Specifies whether the sub-counter value is incremented or decremented: 0: Incremented (+ correction) Incrementation value = (RTCAnF[5:0] value – 1) × 2 1: Decremented (– correction) Decrementation value = (inverted data of RTCAnF[5:0] value + 1) × 2
5 to 0	RTCAnF[5:0]	Error correction value

NOTES

1. When RTCAnF[5:1] = 00000_B, clock error correction is not performed.
2. Perform RTCAnSUBU write as described in
   - Section 31.5.1, Initial Setting of the RTCA, and
   - Section 31.5.5, Writing to RTCAnSUBU.

31.3.3.4 RTCAnSCMP — RTCA Sub-Counter Compare Register

This register sets the compare value of the sub-counter RTCAnSUBC in frequency selection mode (RTCAnCTL0.RTCAnSLSB = 1).

When the sub-counter values matches the value of this register, an overflow signal is output to the seconds counter RTCAnSECC and the sub-counter is cleared.

Set the value for this register according to the frequency of the input clock RTCATCKI.

Access: This register can be read/written in 32-bit units.
Note the following when writing this register during sub-counter operation:
- Previous RTCAnSCMP write must be completed (RTCAnCTL2.RTCAnWSST = 0).
- The write operation ends with the next sub-counter overflow.

Address: <RTCAn_base> + $3C_H$

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnSCMP[21:16]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTCAnSCMP[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.15 RTCAnSCMP Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
21 to 0	RTCAnSCMP [21:0]	Sub-counter compare value in frequency selection mode.

Example

The following example illustrates the setting of RTCAnSCMP:

- RTCATCKI = 4 MHz = 4,000,000 Hz
- RTCAnSCMP[21:0] = 4,000,000 – 1 = 3,999,999 (decimal code) = $3D08FF_H$
- The seconds counter RTCAnSECC is triggered when the sub-counter value changes from $3D08FF_H$ to 0_H.

NOTES

1. The operation of the RTCA cannot be guaranteed if a value of 3198 (decimal code) or lower is set in this register.
2. Perform RTCAnSCMP write as described in Section 31.5.1, Initial Setting of the RTCA and Section 31.5.6, Writing to RTCAnSCMP.

31.3.4 Details of RTCA Clock Counter and Buffer Registers

31.3.4.1 RTCAnSECC — RTCA Seconds Count Register

This register is the seconds counter. It counts seconds from 00 to 59 in BCD.

This register counts as follows.

- It is triggered by every overflow of the sub-counter RTCAnSUBC.

  If the sub-counter overflows while the seconds counter is stopped (RTCAnCTL2.RTCAnWST = 1), the seconds counter behaves as follows:

  – If one sub-counter overflow occurs while the seconds counter is stopped, the overflow is held internally.
  The seconds counter is incremented by one when it is restarted.

  – If two or more overflows occur while the seconds counter is stopped, the overflow count cannot be held internally.
  The seconds counter is incremented by one when it is restarted.

  – If the seconds counter was updated while the seconds counter is stopped, the sub-counter overflow(s) are ignored.

- It outputs an overflow signal when the value changes from 59 to 00. The overflow signal triggers the minutes counter (RTCAnMINC).

Access: This register can only be read in 8-bit units.

Address: <RTCAn_base> + $4C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnSECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 31.16 RTCAnSECC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnSECC [6:0]	Seconds in BCD

NOTES

1. Perform RTCAnSECC read according to the flow described in Section 31.5.3, Reading Clock Counters.
2. A start value can be assigned to this register by writing to the seconds count buffer register RTCAnSEC or to the clock time setting register RTCAnTIME. See
   - **Section 31.5.1, Initial Setting of the RTCA**, and
   - **Section 31.5.2, Updating Clock Counters**

31.3.4.2 RTCAnSEC — RTCA Seconds Count Buffer Register

This register is a buffer register to read/write the seconds counter RTCAnSECC.

Access: This register can be read/written in 8-bit units.

Address: <RTCAn_base> + 14_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnSEC[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.17 RTCAnSEC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6 to 0	RTCAnSEC [6:0]	Seconds in BCD

NOTES

1. When writing this register, only decimal values between 00 and 59 in BCD are allowed.
2. Perform RTCAnSEC read/write as described in
   - **Section 31.5.1, Initial Setting of the RTCA,**
   - **Section 31.5.2, Updating Clock Counters**, and
   - **Section 31.5.3, Reading Clock Counters.**

31.3.4.3 RTCAnMINC — RTCA Minutes Count Register

This register is the minutes counter. It counts minutes from 00 to 59 in BCD.

This register counts as follows.

- It is triggered by every overflow of the seconds counter RTCAnSECC.
- It outputs an overflow signal when the value changes from 59 to 00. The overflow signal triggers the hours counter (RTCAnHOURC).

Access: This register can only be read in 8-bit units.

Address: <RTCAn_base> + 50_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnMINC[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 31.18 RTCAnMINC Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnMINC [6:0]	Minutes in BCD

NOTES

1. Perform RTCAnMINC read according to the flow described in Section 31.5.3, Reading Clock Counters.
2. A start value can be assigned to this register by writing to the minutes count buffer register RTCAnMIN or to the time count buffer register RTCAnTIME. See
   - **Section 31.5.1, Initial Setting of the RTCA**, and
   - **Section 31.5.2, Updating Clock Counters.**

31.3.4.4 RTCAnMIN — RTCA Minutes Count Buffer Register

This register is a buffer register to read/write the minutes counter RTCAnMINC.

Access: This register can be read/written in 8-bit units.

Address: <RTCAn_base> + 18_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnMIN[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.19 RTCAnMIN Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6 to 0	RTCAnMIN [6:0]	Minutes in BCD

NOTES

1. When writing this register, only decimal values between 00 and 59 in BCD are allowed.
2. Perform RTCAnMIN read/write as described in
   - **Section 31.5.1, Initial Setting of the RTCA,**
   - **Section 31.5.2, Updating Clock Counters**, and
   - **Section 31.5.3, Reading Clock Counters.**

31.3.4.5 RTCAnHOURC — RTCA Hours Count Register

This register is the hours counter. It counts the hours in BCD. The count range depends on the selected hour format. See Table 31.21, 12- and 24-Hour Format.

This register counts as follows.

- It is triggered by every overflow of the minutes counter RTCAnMINC.
- It outputs an overflow signal when the value changes from 23 to 00 (in 24-hour format) or from 31 to 12 (in 12-hour format). The overflow signal triggers two counters:
  - Day of the week counter (RTCAnWEEKC)
  - Day of the month counter (RTCAnDAYC)

Access: This register can only be read in 8-bit units.

Address: <RTCAn_base> + 54_H

Value after reset: 12_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnHOURC[5:0]					
Value after reset	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R

Table 31.20 RTCAnHOURC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	RTCAnHOURC [5:0]	Hours in BCD. See Table 31.21, 12- and 24-Hour Format, for details.

NOTES

1. Perform RTCAnHOURC read according to the flow described in Section 31.5.3, Reading Clock Counters.
2. A start value can be assigned to this register by writing to the hours count buffer register RTCAnHOUR or to the time count buffer register RTCAnTIME. See
   - **Section 31.5.1, Initial Setting of the RTCA**, and
   - **Section 31.5.2, Updating Clock Counters.**

12- or 24-hour format

The count values of RTCAnHOURC depend on the selected hour format.

If 12-hour format is selected (RTCAnCTL0.RTCAnAMPM = 0), bit 5 in the RTCAnHOURC register is the am/pm indicator:

- RTCAnHOURC[5] = 0: am
- RTCAnHOURC[5] = 1: pm

The following table shows the count range of RTCAnHOURC in both 12- and 24-hour format.

Table 31.21 12- and 24-Hour Format

12-Hour Format (RTCAnAMPM = 0)			24-Hour Format (RTCAnAMPM = 1)	
Time	RTCAnHOURC		Time	RTCAnHOURC
0 am	12_H		0	00_H
1 am	01_H		1	01_H
2 am	02_H		2	02_H
3 am	03_H		3	03_H
4 am	04_H		4	04_H
5 am	05_H		5	05_H
6 am	06_H		6	06_H
7 am	07_H		7	07_H
8 am	08_H		8	08_H
9 am	09_H		9	09_H
10 am	10_H		10	10_H
11 am	11_H		11	11_H
0 pm	32_H	↓ pm indicator in 12-hour format: RTCAnHOUR.RTCAnHOUR[5] = 1	12	12_H
1 pm	21_H		13	13_H
2 pm	22_H		14	14_H
3 pm	23_H		15	15_H
4 pm	24_H		16	16_H
5 pm	25_H		17	17_H
6 pm	26_H		18	18_H
7 pm	27_H		19	19_H
8 pm	28_H		20	20_H
9 pm	29_H		21	21_H
10 pm	30_H		22	22_H
11 pm	31_H		23	23_H

31.3.4.6 RTCAnHOUR — RTCA Hours Count Buffer Register

This register is a buffer register to read/write the hours counter RTCAnHOURC.

Access: This register can be read/written in 8-bit units.

Address: <RTCAn_base> + $1C_H$

Value after reset: 12_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnHOUR[5:0]					
Value after reset	0	0	0	1	0	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.22 RTCAnHOUR Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5 to 0	RTCAnHOUR [5:0]	Hours in BCD See Table 31.21, 12- and 24-Hour Format, for details.

NOTES

1. When writing this register, only the following decimal values in BCD are allowed:
   - 12-hour format (RTCAnCTL0.RTCAnAMPM = 0):
     01 to 12 or 21 to 32
   - 24-hour format (RTCAnCTL0.RTCAnAMPM = 1):
     00 to 23
2. Perform RTCAnHOUR read/write as described in
   - **Section 31.5.1, Initial Setting of the RTCA,**
   - **Section 31.5.2, Updating Clock Counters**, and
   - **Section 31.5.3, Reading Clock Counters.**

31.3.4.7 RTCAnWEEKC — RTCA Day of the Week Count Register

This register is the day of the week counter. It counts from 0 to 6.

This register counts as follows.

- It is triggered by every overflow of the hours counter RTCAnHOURC.

Access: This register can only be read in 8-bit units.

Address: <RTCAn_base> + 58_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	RTCAnWEEKC[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 31.23 RTCAnWEEKC Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned.
2 to 0	RTCAnWEEKC [2:0]	Day of the week

NOTES

1. Perform RTCAnWEEKC read according to the flow described in Section 31.5.3, Reading Clock Counters.
2. A start value can be assigned to this register by writing to the day of the week count buffer register RTCAnWEEK or to the calendar count buffer register RTCAnCAL. See
   - **Section 31.5.1, Initial Setting of the RTCA**, and
   - **Section 31.5.2, Updating Clock Counters.**

31.3.4.8 RTCAnWEEK — RTCA Day of the Week Count Buffer Register

This register is a buffer register to read/write the day of the week counter RTCAnWEEKC.

There is no particular correspondence between the value of RTCAnWEEK and the day of the week. Set the correspondence according to the application to be used.

Example: 0 = Sunday, 1 = Monday, …, 6 = Saturday

Access: This register can be read/written in 8-bit units.

Address: <RTCAn_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	RTCAnWEEK[2:0]		
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 31.24 RTCAnWEEK Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2 to 0	RTCAnWEEK [2:0]	Day of the week

NOTES

1. When writing this register, only decimal values between 0 and 6 in BCD are allowed.
2. Perform RTCAnWEEK read/write as described in
   - **Section 31.5.1, Initial Setting of the RTCA,**
   - **Section 31.5.2, Updating Clock Counters**, and
   - **Section 31.5.3, Reading Clock Counters.**

31.3.4.9 RTCAnDAYC — RTCA Day of the Month Count Register

This register is the day of the month counter. It counts from 01 to a maximum of 31 in BCD, depending on the value of the month counter (RTCAnMONC) and the year counter (RTCAnYEARC):

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, non-leap year)

Years 0, 4, 8, 12, etc., are considered leap years.

This register counts as follows.

- It is triggered by every overflow of the hours of the day counter RTCAnHOURC.
- It outputs an overflow signal when the value changes from 28, 29, 30, or 31 to 01, depending on the current month and year. The overflow signal triggers the month counter (RTCAnMONC).

Access: This register can only be read in 8-bit units.

Address: <RTCAn_base> + $5C_H$

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnDAYC[5:0]					
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R

Table 31.25 RTCAnDAYC Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	RTCAnDAYC [5:0]	Day of the month in BCD

NOTES

1. Perform RTCAnDAYC read according to the flow described in Section 31.5.3, Reading Clock Counters.
2. A start value can be assigned to this register by writing to the day of the month count buffer register RTCAnDAY or to the calendar count buffer register RTCAnCAL. See
   - **Section 31.5.1, Initial Setting of the RTCA**, and
   - **Section 31.5.2, Updating Clock Counters.**

31.3.4.10 RTCAnDAY — RTCA Day of the Month Count Buffer Register

This register is a buffer register to read/write the day of the month counter RTCAnDAYC.

Access: This register can be read/written in 8-bit units.

Address: <RTCAn_base> + 24_H

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnDAY[5:0]					
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.26 RTCAnDAY Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5 to 0	RTCAnDAY [5:0]	Day of the month in BCD

NOTES

1. When writing this register, only decimal values between 01 and 31 in BCD are allowed:
   - 01 to 31 (January, March, May, July, August, October, December)
   - 01 to 30 (April, June, September, November)
   - 01 to 29 (February, leap year)
   - 01 to 28 (February, non-leap year)
2. Perform RTCAnDAY read/write as described in
   - **Section 31.5.1, Initial Setting of the RTCA,**
   - **Section 31.5.2, Updating Clock Counters**, and
   - **Section 31.5.3, Reading Clock Counters.**

31.3.4.11 RTCAnMONC — RTCA Month Count Register

This register is the month counter. It counts the month of the year, starting from 01 to 12 in BCD.

This register counts as follows.

- It is triggered by every overflow of the counter for the day of the month RTCAnDAYC.
- It outputs an overflow signal when the value changes from 12 to 01. The overflow signal triggers the year counter (RTCAnYEARC).

Access: This register can only be read in 8-bit units.

Address: <RTCAn_base> + 60_H

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	RTCAnMONC[4:0]				
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R

Table 31.27 RTCAnMONC Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned.
4 to 0	RTCAnMONC [4:0]	Month of the year in BCD

NOTES

1. Perform RTCAnMONC read according to the flow described in **Section 31.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the month count buffer register RTCAnMONTH or to the calendar count buffer register RTCAnCAL. See
   - **Section 31.5.1, Initial Setting of the RTCA**, and
   - **Section 31.5.2, Updating Clock Counters.**

31.3.4.12 RTCAnMONTH — RTCA Month Count Buffer Register

This register is a buffer register to read/write the month counter RTCAnMONC.

Access: This register can be read/written in 8-bit units.

Address: <RTCAn_base> + 28_H

Value after reset: 01_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	RTCAnMONTH[4:0]				
Value after reset	0	0	0	0	0	0	0	1
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 31.28 RTCAnMONTH Register Contents

Bit Position	Bit Name	Function
7 to 5	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
4 to 0	RTCAnMONTH [4:0]	Month of the year in BCD

NOTES

1. When writing this register, only decimal values between 01 and 12 in BCD are allowed.
2. Perform RTCAnMONTH read/write as described in
   - **Section 31.5.1, Initial Setting of the RTCA,**
   - **Section 31.5.2, Updating Clock Counters**, and
   - **Section 31.5.3, Reading Clock Counters.**

31.3.4.13 RTCAnYEARC — RTCA Year Count Register

This register is the year counter. It counts years from 00 to a maximum of 99 in BCD.

Years 00, 04, 08, …, 92, and 96 (every four years) are considered leap years.

This register counts as follows.

- It is triggered by every overflow of the month counter RTCAnMONC.

Access: This register can only be read in 8-bit units.

Address: <RTCAn_base> + 64_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnYEARC[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 31.29 RTCAnYEARC Register Contents

Bit Position	Bit Name	Function
7 to 0	RTCAnYEARC [7:0]	Year in BCD

NOTES

1. Perform RTCAnYEARC read according to the flow described in **Section 31.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the year count buffer register RTCAnYEAR or to the calender count buffer register RTCAnCAL. See
   - Section 31.5.1, Initial Setting of the RTCA, and
   - **Section 31.5.2, Updating Clock Counters.**

31.3.4.14 RTCAnYEAR — RTCA Year Count Buffer Register

This register is a buffer register to read/write the year counter RTCAnYEARC.

Access: This register can be read/written in 8-bit units.

Address: <RTCAn_base> + $2C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnYEAR[7:0]							
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.30 RTCAnYEAR Register Contents

Bit Position	Bit Name	Function
7 to 0	RTCAnYEAR [7:0]	Year in BCD

NOTES

1. When writing this register, only decimal values between 00 and 99 in BCD are allowed.
2. Perform RTCAnYEAR read/write as described in
   - **Section 31.5.1, Initial Setting of the RTCA,**
   - Section 31.5.2, Updating Clock Counters, and
   - **Section 31.5.3, Reading Clock Counters.**

31.3.5 Details of RTCA Special Counter and Buffer Registers

31.3.5.1 RTCAnTIMEC — RTCA Time Count Register

This register enables the RTCAnHOURC, RTCAnMINC, and RTCAnSECC counters to be read simultaneously.

Access: This register can only be read in 32-bit units.

Address: <RTCAn_base> + 68_H

Value after reset: 0012 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnHOURC[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RTCAnMINC[6:0]							—	RTCAnSECC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.31 RTCAnTIMEC Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned.
21 to 16	RTCAnHOURC [5:0]	Hours in BCD. See Table 31.21, 12- and 24-Hour Format, for details.
15	Reserved	When read, the value after reset is returned.
14 to 8	RTCAnMINC [6:0]	Minutes in BCD
7	Reserved	When read, the value after reset is returned.
6 to 0	RTCAnSECC [6:0]	Seconds in BCD

NOTES

1. Perform RTCAnTIMEC read according to the flow described in **Section 31.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the time count buffer register RTCAnTIME. See
   - **Section 31.5.1, Initial Setting of the RTCA**, and
   - **Section 31.5.2, Updating Clock Counters.**

31.3.5.2 RTCAnTIME — RTCA Time Count Buffer Register

This register enables the RTCAnHOUR, RTCAnMIN, and RTCAnSEC buffer registers to be read/written simultaneously.

Access: This register can be read/written in 32-bit units.

Address: <RTCAn_base> + 30_H

Value after reset: $0012\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	RTCAnHOUR[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	RTCAnMIN[6:0]							—	RTCAnSEC[6:0]						
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.32 RTCAnTIME Register Contents

Bit Position	Bit Name	Function
31 to 22	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
21 to 16	RTCAnHOUR [5:0]	Hours in BCD See Table 31.21, 12- and 24-Hour Format, for details.
15	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
14 to 8	RTCAnMIN [6:0]	Minutes in BCD
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6 to 0	RTCAnSEC [6:0]	Seconds in BCD

NOTE

Perform RTCAnTIME read/write as described in

- **Section 31.5.1, Initial Setting of the RTCA,**
- Section 31.5.2, Updating Clock Counters, and
- **Section 31.5.3, Reading Clock Counters.**

31.3.5.3 RTCAnCALC — RTCA Calendar Count Register

This register enables the RTCAnYEARC, RTCAnMONC, RTCAnDAYC, and RTCAnWEEKC counters to be read simultaneously.

Access: This register can only be read in 32-bit units.

Address: <RTCAn_base> + $6C_H$

Value after reset: $0001\ 0100_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RTCAnYEARC[7:0]								—	—	—	RTCAnMONC[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RTCAnDAYC[5:0]						—	—	—	—	—	RTCAnWEEKC[2:0]		
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 31.33 RTCAnCALC Register Contents

Bit Position	Bit Name	Function
31 to 24	RTCAnYEARC [7:0]	Year in BCD
23 to 21	Reserved	When read, the value after reset is returned.
20 to 16	RTCAnMONC [4:0]	Month of the year in BCD
15, 14	Reserved	When read, the value after reset is returned.
13 to 8	RTCAnDAYC [5:0]	Day of the month in BCD
7 to 3	Reserved	When read, the value after reset is returned.
2 to 0	RTCAnWEEKC [2:0]	Day of the week in BCD

NOTES

1. Perform RTCAnCALC read according to the flow described in **Section 31.5.3, Reading Clock Counters**.
2. A start value can be assigned to this register by writing to the clock time setting register RTCAnCAL. See
   - **Section 31.5.1, Initial Setting of the RTCA**, and
   - **Section 31.5.2, Updating Clock Counters.**

31.3.5.4 RTCAnCAL — RTCA Calendar Count Buffer Register

This register enables the RTCAnYEAR, RTCAnMONTH, RTCAnDAY, and RTCAnWEEK buffer registers to be read/written simultaneously.

Access: This register can be read/written in 32-bit units.

Address: <RTCAn_base> + 34_H

Value after reset: $0001\ 0100_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RTCAnYEAR[7:0]								—	—	—	RTCAnMONTH[4:0]				
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	RTCAnDAY[5:0]						—	—	—	—	—	RTCAnWEEK[2:0]		
Value after reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 31.34 RTCAnCAL Register Contents

Bit Position	Bit Name	Function
31 to 24	RTCAnYEAR [7:0]	Year in BCD
23 to 21	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
20 to 16	RTCAnMONTH [4:0]	Month of the year in BCD
15, 14	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
13 to 8	RTCAnDAY [5:0]	Day of the month in BCD
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2 to 0	RTCAnWEEK [2:0]	Day of the week in BCD

NOTE

Perform RTCAnCAL read/write as described in

- **Section 31.5.1, Initial Setting of the RTCA.**
- **Section 31.5.2, Updating Clock Counters**, and
- **Section 31.5.3, Reading Clock Counters.**

31.3.6 Details of RTCA Alarm Setting Registers

31.3.6.1 RTCAnALM — RTCA Alarm Minute Setting Register

This register specifies the minute of the alarm interrupt.

For details and example settings, see Section 31.4.3, Alarm Interrupt Function.

Access: This register can be read/written in 8-bit units.

Address: <RTCAn_base> + 40_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnALM[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.35 RTCAnALM Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6 to 0	RTCAnALM [6:0]	Minute of the alarm interrupt in BCD

NOTES

1. If decimal values outside the range of 00 to 59 in BCD are set, no alarm interrupt request will be generated.
2. When the setting of RTCAnALM is changed during sub-counter operation (RTCAnCTL0.RTCAnCEST = 1), a glitch may be output to RTCATINTAL. Implement appropriate interrupt mask processing procedures.

31.3.6.2 RTCAnALH — RTCA Alarm Hour Setting Register

This register specifies the hour of the alarm interrupt.

For details and example settings, see Section 31.4.3, Alarm Interrupt Function.

Access: This register can be read/written in 8-bit units.

Address: <RTCAn_base> + 44_H

Value after reset: 12_H

Bit	7	6	5	4	3	2	1	0
	—	—	RTCAnALH[5:0]					
Value after reset	0	0	0	1	0	0	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.36 RTCAnALH Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
5 to 0	RTCAnALH [5:0]	Hour of the alarm interrupt in BCD

NOTES

1. If decimal values outside the following range are set, no alarm interrupt request will be generated:
   - 12-hour format (RTCAnCTL0.RTCAnAMPM = 0): 01 to 12 or 21 to 32
   - 24-hour format (RTCAnCTL0.RTCAnAMPM = 1): 00 to 23
2. When the setting of RTCAnALH is changed during sub-counter operation (RTCAnCTL0.RTCAnCEST = 1), a glitch may be output to RTCATINTAL. Implement appropriate interrupt mask processing procedures.

31.3.6.3 RTCAnALW — RTCA Alarm Day of the Week Setting Register

This register specifies the day(s) of the week of the alarm interrupt.

Access: This register can be read/written in 8-bit units.

Address: <RTCAn_base> + 48_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	RTCAnALW[6:0]						
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31.37 RTCAnALW Register Contents

Bit Position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6 to 0	RTCAnALW [6:0]	Specifies day of the week m (m = 0 to 6) as a day, when an alarm interrupt request is generated: 0: No alarm interrupt request is generated on day m. 1: Alarm interrupt request is generated on day m at the time set using RTCAnALM and RTCAnALH. The bits of this register correspond to the count value of the day of the week counter (RTCAnWEEKC).

NOTE

When the setting of RTCAnALW is changed during sub-counter operation (RTCAnCTL0.RTCAnCE = 1), a glitch may be output to RTCATINTAL. Implement appropriate interrupt mask processing procedures.

Example

If Sunday is RTCAnWEEK = 0, Monday is RTCAnWEEK = 1, Tuesday is RTCAnWEEK = 2, …, Saturday is RTCAnWEEK = 6:

- To set the alarm for Sunday, set RTCAnALW = $0000\ 0001_B$.
- To set the alarm for Monday and Wednesday, set RTCAnALW = $0000\ 1010_B$.
- To set the alarm for Tuesday, Thursday, and Saturday, set RTCAnALW = $0101\ 0100_B$.

For more examples, see Section 31.4.3, Alarm Interrupt Function.

31.3.7 RTCA Emulation Register

31.3.7.1 RTCAnEMU — RTCA Emulation Register

This register controls operation by SVSTOP.

Access: This register can be read/written in 8- or 1-bit units.
A write should be performed when EPC.SVSTOP = 0.

Address: <RTCAn_base> + 74_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	RTCAnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 31.38 RTCAnEMU Register Contents

Bit Position	Bit Name	Function
7	RTCAnSVSDIS	When the EPC.SVSTOP bit is set to 0: Count clock is supplied when the debugger gains microcontroller control (at a breakpoint, etc.) regardless of the value of this bit. When the EPC.SVSTOP bit is set to 1: 0: Count clock is stopped when the debugger gains microcontroller control (at a breakpoint, etc.). 1: Count clock continues to be supplied when the debugger gains microcontroller control (at a breakpoint, etc.).
6 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

31.4 Operation

The RTCA provides two operation modes:

- Frequency selection mode
- 32.768-kHz mode

The operation mode that can be used depends on the available input clock RTCATCKI. The operation mode specifies the sub-counter compare value that is used to trigger the seconds counter and thus all subsequent counters. Clock error correction is only possible in 32.768-kHz mode.

The following table provides an overview of the properties of the two operation modes.

Table 31.39 RTCA Operation Mode Overview

		32.768-kHz Mode	
	Frequency Selection Mode	**Clock Correction Disabled**	**Clock Correction Enabled**
Allowed input clock RTCATCKI	Any frequency from 32 kHz to 4.194304MHz	32.768 kHz	Any frequency from 32.76180000 kHz to 32.77420000 kHz
Sub-counter RTCAnSUBC operation	• Counter overflow at value of RTCAnSCMP • RTCAnSCMP must be set to RTCATCKI-1 (in Hz)	Counter overflow at $7FFF_H$	Counter overflow at $7FFF_H$ or Every 20 or 60 seconds: $7FFF_H$ ±RTCAnSUBU.RTCAnF[5:0]

The operation mode is selected by control bit RTCAnCTL0.RTCAnSLSB. For details on how to set the operation mode during RTCA initialization, see Section 31.5.1, Initial Setting of the RTCA.

CAUTIONS

1. **The input clock RTCATCKI must not be outside the allowed frequency range.**
2. **The operation mode must not be changed while sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1).**

31.4.1 Clock Counter Format

The clock counters (RTCAnSECC to RTCAnYEARC) operate on binary coded decimals (BCD): Each digit is represented by its own binary sequence.

Depending on the valid data range, the number of bits for a digit differs. For example, the tens digit of the month of the year counter has only one bit (for 0 and 1) whereas the tens digit of the minutes counter has 3 bits (for 0 to 5).

The following table lists the decimals 0 to 59 in binary and BCD.

Table 31.40 Example of BCD Code – Seconds or Minutes Counter (0 to 59)

Decimal	Binary	BCD
0	000000	000 0000
1	000001	000 0001
2	000010	000 0010
3	000011	000 0011
4	000100	000 0100
5	000101	000 0101
6	000110	000 0110
7	000111	000 0111
8	001000	000 1000
9	001001	000 1001
10	001010	001 0000
11	001011	001 0001
12	001100	001 0010
:	:	:
58	111010	101 1000
59	111011	101 1001

31.4.2 Fixed Interval Interrupt Function

Interrupt RTCATINTR can be specified to occur after every 0.25 seconds, 0.5 seconds, 1 (full) second, 1 (full) minute, 1 (full) hour, 1 (full) day, or 1 (full) month.

The fixed interval interrupt function is controlled by bits RTCAnCTL1.RTCAnCT[2:0].

31.4.3 Alarm Interrupt Function

Interrupt RTCATINTAL can be specified to occur at a certain time on one or several days of the week. This interrupt can be used as a wake-up signal.

The alarm interrupt function is enabled and disabled by bit RTCAnCTL1.RTCAnENALM.

The alarm setting is specified by the following control registers:

- RTCAnALW selects the weekday(s).

The allocation of bits to weekdays is defined by the day of the week count buffer register RTCAnWEEK.

- RTCAnALH and RTCAnALM specify the hour and minute in BCD.

Examples

The following tables show some exemplary settings of the alarm control registers for both 12-hour and 24-hour format.

In this example, Sunday is RTCAnWEEK = 0, Monday is RTCAnWEEK = 1, Tuesday is RTCAnWEEK = 2, …, Saturday is RTCAnWEEK = 6:

Table 31.41 Alarm Setting in 12-Hour Format (RTCAnCTL0.RTCAnAMPM = 0)

Alarm Setting Time	RTCAnALW	RTCAnALH	RTCAnALM
Sunday 7:00 am	01_H	07_H	00_H
Sunday, Monday 12:15 pm	03_H	32_H	15_H
Monday, Wednesday, Friday 5:30 pm	$2A_H$	25_H	30_H
Daily, 10:45 pm	$7F_H$	30_H	45_H

Table 31.42 Alarm Setting in 24-Hour Format (RTCAnCTL0.RTCAnAMPM = 1)

Alarm Setting Time	RTCAnALW	RTCAnALH	RTCAnALM
Sunday 7:00	01_H	07_H	00_H
Sunday, Monday 12:15	03_H	12_H	15_H
Monday, Wednesday, Friday 17:30	$2A_H$	17_H	30_H
Daily, 22:45	$7F_H$	22_H	45_H

31.4.4 Clock Error Correction

Clock error correction compensates for deviations of the oscillator from the nominal clock rate. With clock error correction input clock rates from 32.76180 kHz to 32.77420 kHz are possible.

The clock error correction function is only available in 32.768-kHz operation mode. In this operation mode, a nominal clock rate of 32.768 kHz is expected and the sub-counters overflow value is fixed to $7FFF_H$.

The following figures illustrate the clock error when the input clock rate deviates from the nominal clock.

RTCATCKI = 32.768 kHz

Figure 31.2, RTCATCKI = 32.768 kHz, No Clock Error Correction Required shows the timing diagram if RTCATCKI matches the nominal clock rate of 32.768 kHz. No clock error correction is required.

Counting from 0 to 32767 (0 to $7FFF_H$) with a 32.768-kHz clock is exactly equal to one second.

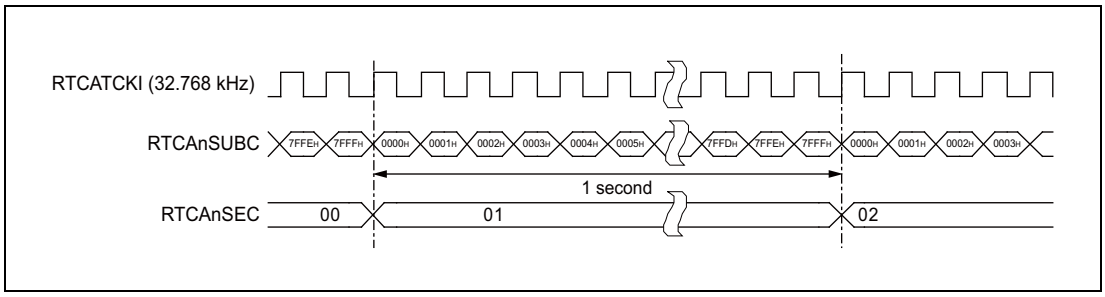


Figure 31.2 RTCATCKI = 32.768 kHz, No Clock Error Correction Required

RTCATCKI = 32.769 kHz

Figure 31.3, RTCATCKI = 32.769 kHz, No Clock Error Correction Enabled shows the timing diagram if RTCATCKI deviates from the nominal clock rate of 32.768 kHz. In this example, RTCATCKI is connected to a 32.769-kHz oscillator. Clock error correction is not enabled.

Counting from 0 to 32767 (0 to $7FFF_H$) with a 32.769-kHz clock is equal to approximately 0.99997 seconds (32768/32769). A "+ error" (faster than 32.768-kHz) occurs. In one month, RTCA deviates approximately –79 seconds from the real time.

Error = (32768/32769 - 1) × 60 (s) × 60 (min) × 24 (h) × 30 (d)

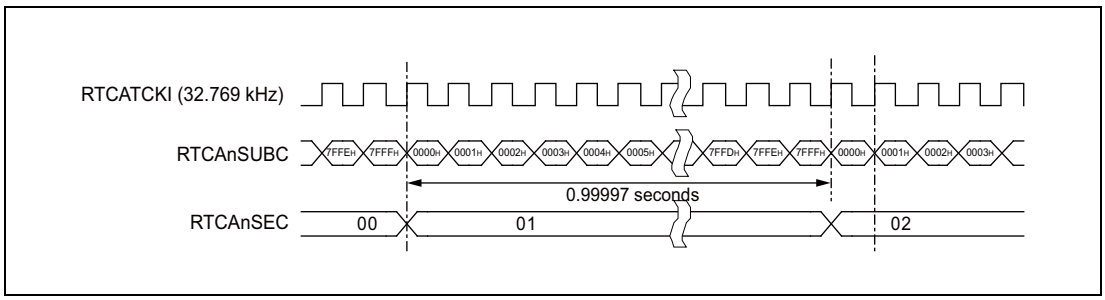


Figure 31.3 RTCATCKI = 32.769 kHz, No Clock Error Correction Enabled

Clock error correction is performed by stretching/reducing the 1-second period of the sub-counter at regular intervals. The sub-counter's upper limit of $7FFF_H$ is increased or decreased by setting the following parameters in register RTCAnSUBU:

- A correction value greater than one
- An operator (add/subtract)
- An interval (20 or 60 seconds)

The corrected overflow value becomes effective every 20 or 60 seconds, so that on the average RTCAnSECC is triggered exactly every second.

31.4.4.1 Setting the Correction Value and the Operator

The correction value and operator are specified by the RTCAnF6, RTCAnF[5:0] bits of the RTCAnSUBU register:

- RTCAnF6 specifies whether the overflow value is incremented or decremented.
- RTCAnF[5:0] specifies the correction value.

The correction values are calculated as follows:

Table 31.43 Correction Value Settings

RTCAnF6	Increment/Decrement	Correction Value
0	Increment	(Value of RTCAnF[5:0] – 1) × 2
1	Decrement	(Inverted value of RTCAnF[5:0] + 1) × 2

Some examples are given in the following table:

Table 31.44 Correction Value Examples

RTCAnF6	RTCAnF[5:0]	Correction Value	Count Limit of RTCAnSUBC
0	15_H	$(15_H - 1) \times 2 = 40$	32768 + 40 = 32808
1	15_H	$(\overline{15}_H + 1) \times 2$ $= (2A_H + 1) \times 2$ $= 86$	32768 – 86 = 32682

31.4.4.2 Impact of the Repetition Interval

The correction value set by RTCAnF6, RTCAnF[5:0] does not change the count limit of RTCAnSUBC every second. The repetition interval at which the correction value becomes effective is specified by bit RTCAnDEV.

This bit also influences the size of the correctable frequency range and the correction accuracy.

The following table summarizes the RTCAnDEV settings.

Table 31.45 Setting of Bit RTCAnSUBU.RTCAnDEV

RTCAn DEV	Count Limit of RTCAnSUBC is Changed	Frequency Range that can be Corrected	Correction Accuracy
0	Every 20 seconds when RTCAnSECC = 00, 20, or 40	32.76180000 to 32.77420000 kHz	
1	Every 60 seconds when RTCAnSECC = 00	32.76593333 to 32.77006667 kHz	Three times higher than for RTCAnDEV = 0

31.4.4.3 Sample Settings

The frequencies that can be corrected, as well as the setting values of bits RTCAnDEV, RTCAnF6, and RTCAnF[5:0], are listed in the following table.

Table 31.46 Correctable Frequency Range when RTCAnDEV = 0

Input Clock Frequency	RTCAnF6	RTCAnF[5:0]	Correction Value of RTCAnSUBC
—	0	000000	No correction
—	0	000001	No correction
32.76810000 kHz	0	000010	Once every 20 s, RTCAnSUBC count value + 2

Table 31.46 Correctable Frequency Range when RTCAnDEV = 0

Input Clock Frequency	RTCAnF6	RTCAnF[5:0]	Correction Value of RTCAnSUBC
32.76820000 kHz	0	000011	Once every 20 s, RTCAnSUBC count value + 4
32.76830000 kHz	0	000100	Once every 20 s, RTCAnSUBC count value + 6
:	:	:	:
32.77400000 kHz	0	111101	Once every 20 s, RTCAnSUBC count value + 120
32.77410000 kHz	0	111110	Once every 20 s, RTCAnSUBC count value + 122
32.77420000 kHz (upper limit)	0	111111	Once every 20 s, RTCAnSUBC count value + 124
—	1	000000	No correction
—	1	000001	No correction
32.76180000 kHz (lower limit)	1	000010	Once every 20 s, RTCAnSUBC count value – 124
32.76190000 kHz	1	000011	Once every 20 s, RTCAnSUBC count value – 122
32.76200000 kHz	1	000100	Once every 20 s, RTCAnSUBC count value – 120
:	:	:	:
32.76770000 kHz	1	111101	Once every 20 s, RTCAnSUBC count value – 6
32.76780000 kHz	1	111110	Once every 20 s, RTCAnSUBC count value – 4
32.76790000 kHz	1	111111	Once every 20 s, RTCAnSUBC count value – 2

Table 31.47 Correctable Frequency Range when RTCAnDEV = 1

Input Clock Frequency	RTCAnF6	RTCAnF[5:0]	Correction Value of RTCAnSUBC
—	0	000000	No correction
—	0	000001	No correction
32.76803333 kHz	0	000010	Once every 60 s, RTCAnSUBC count value + 2
32.76806667 kHz	0	000011	Once every 60 s, RTCAnSUBC count value + 4
32.76810000 kHz	0	000100	Once every 60 s, RTCAnSUBC count value + 6
:	:	:	:
32.77000000 kHz	0	111101	Once every 60 s, RTCAnSUBC count value + 120
32.77003333 kHz	0	111110	Once every 60 s, RTCAnSUBC count value + 122
32.77006667 kHz (upper limit)	0	111111	Once every 60 s, RTCAnSUBC count value + 124
—	1	000000	No correction
—	1	000001	No correction
32.76593333 kHz (lower limit)	1	000010	Once every 60 s, RTCAnSUBC count value – 124
32.76596667 kHz	1	000011	Once every 60 s, RTCAnSUBC count value – 122

Table 31.47 Correctable Frequency Range when RTCAnDEV = 1

Input Clock Frequency	RTCAnF6	RTCAnF[5:0]	Correction Value of RTCAnSUBC
32.76600000 kHz	1	000100	Once every 60 s, RTCAnSUBC count value – 120
:	:	:	:
32.76790000 kHz	1	111101	Once every 60 s, RTCAnSUBC count value – 6
32.76793333 kHz	1	111110	Once every 60 s, RTCAnSUBC count value – 4
32.76796667 kHz	1	111111	Once every 60 s, RTCAnSUBC count value – 2

31.5 Procedures for Setup, Writing and Reading

The following subsections provide flow charts that illustrate the procedures for RTCA setup and for reading and writing the RTCA clock counters.

31.5.1 Initial Setting of the RTCA

The RTCA must be stopped before setting the initial setting value of each counter.

31.5.1.1 RTCA Stop Procedure

Stop the RTCA according to the following flow.

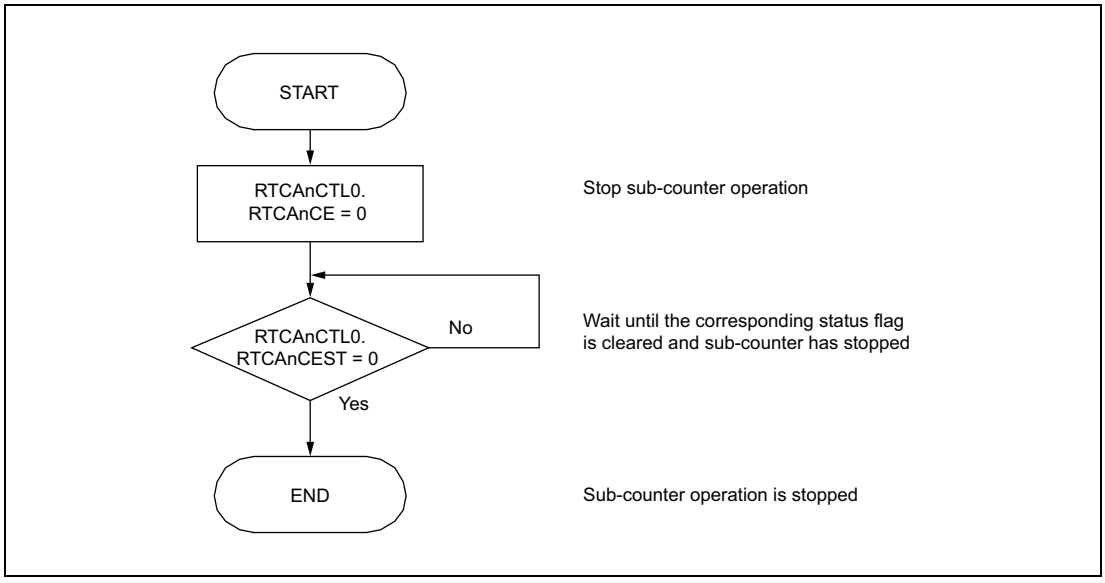


Figure 31.4 RTCA Stop Procedure

31.5.1.2 RTCA Initialization Procedure

Perform the initial setting of the RTCA according to the following flow:

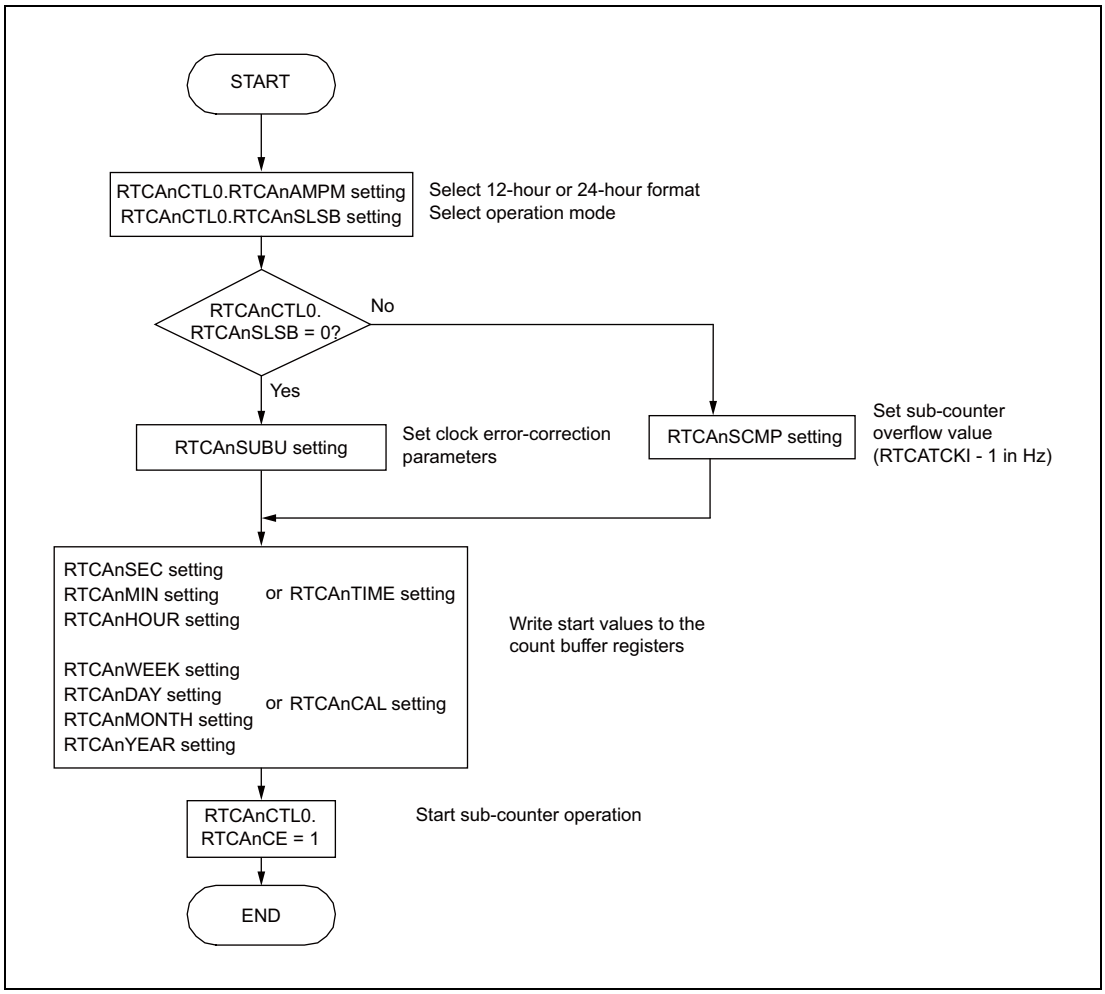


Figure 31.5 RTCA Initial Setup Procedure

CAUTION

The internal clock counter is synchronized with RTCATCKI.

In addition, two RTCATCKI periods are required before the clock counter starting behind END of the above flow.

Therefore, PCLK must be continuously supplied until the completion of the initial setting.

Check that RTCAnCTL0.RTCAnCEST = 1, when the supply of PCLK is stopped after setting the initial setting value of RTCA.

31.5.2 Updating Clock Counters

The clock counters RTCAnSECC to RTCAnYEARC can be stopped and updated while the sub-counter is running.

To update the clock counter when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), follow the flowchart shown below.

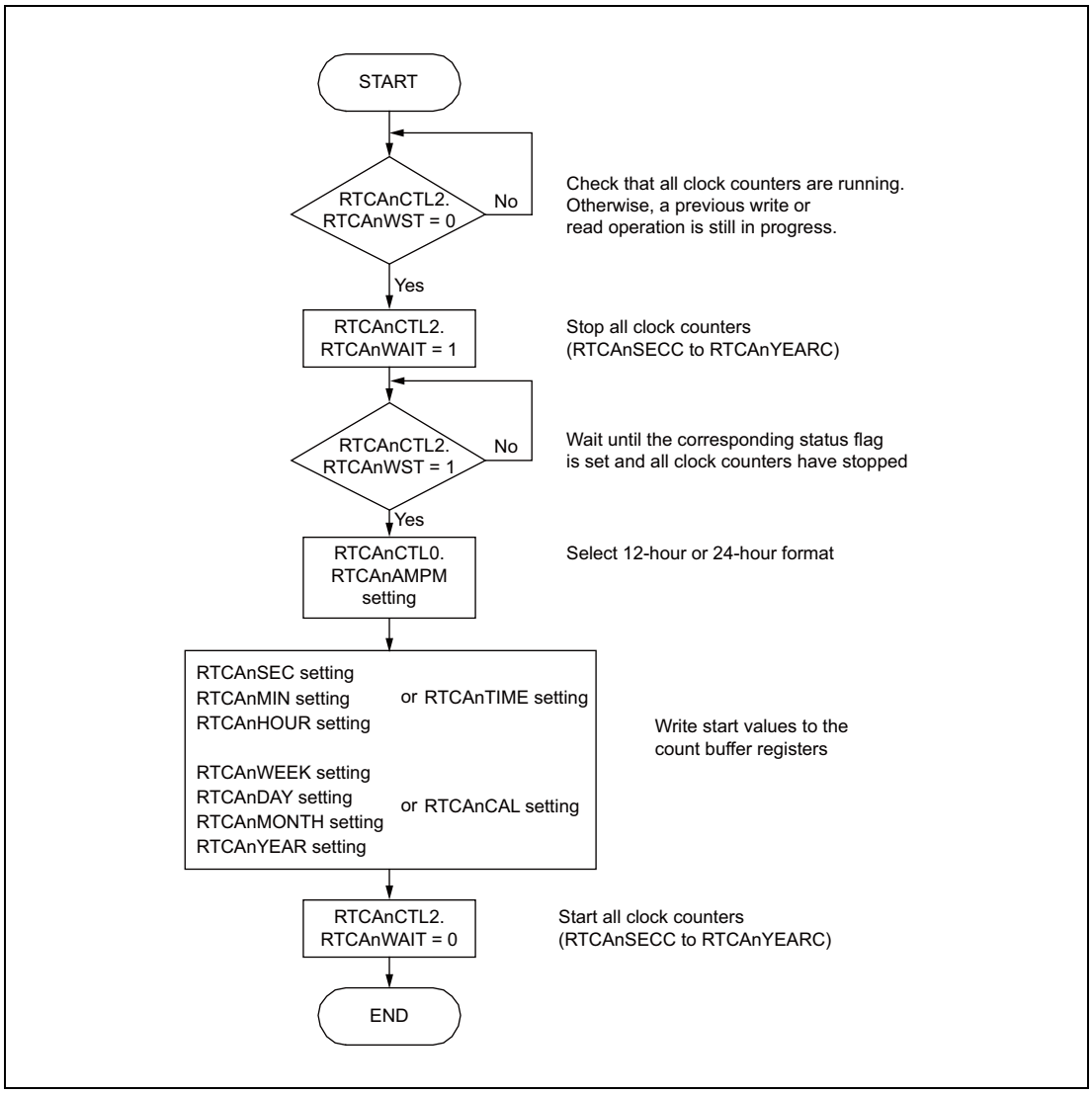


Figure 31.6 Updating Clock Counter Values

CAUTIONS

1. **The internal clock counter is synchronized with RTCATCKI.**
   In addition, two RTCATCKI periods are required before the clock counter updating behind END of the above flow.
   Therefore, PCLK must be continuously supplied until the completion of the clock counter updating.
   Check that RTCAnCTL2.RTCAnWST = 0 before stopping the supply of PCLK after the completion of clock counter updating.
2. **The update procedure must be completed within one second. Otherwise the Real-Time Clock will not count correctly any more:**

3. **Only one sub-counter overflow can be held internally and increment the seconds counter after restarting the clock counters if the value is held.**
4. **If the sub-counter overflows more than once during clock counter stop, the overflow count cannot be held internally. Thus the seconds counter is incremented by one instead of by two when it is restarted.**

31.5.3 Reading Clock Counters

There are two methods to read the clock counters while sub-counter operation is enabled:

- Reading count buffer registers
- Reading counter registers

The advantages and disadvantages of the two methods are summarized in the following table.

Table 31.48 Comparison of the Two Read Methods

	Advantage	Disadvantage
Reading count buffer registers	It is unnecessary to read clock counters several times because the clock counters are read synchronously.	A program wait state occurs between setting RTCAnCTL2.RTCAnWAIT = 1 and completion of data transfer.
Reading count registers	Program wait state does not occur.	If the sub-counter increments, the clock counters must be read several times because they are read asynchronously to RTCATCKI.

31.5.3.1 Procedure for Reading Count Buffer Registers

The following operations are necessary:

1. Stop all clock counters (RTCAnCTL2.RTCAnWAIT = 1). The value of the clock counters is transferred to the corresponding count buffer registers.
2. Read the count buffer registers.

A program wait state occurs between setting RTCAnCTL2.RTCAnWAIT = 1 and completion of data transfer.

The maximum delay is three PCLK periods plus two RTCATCKI periods. For example, if the RTCA operates with PCLK = 40 MHz and RTCATCKI = 32.768 kHz, the delay is about 61 µs.

To read the count buffer register when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCEST = 1), follow the flowchart shown below.

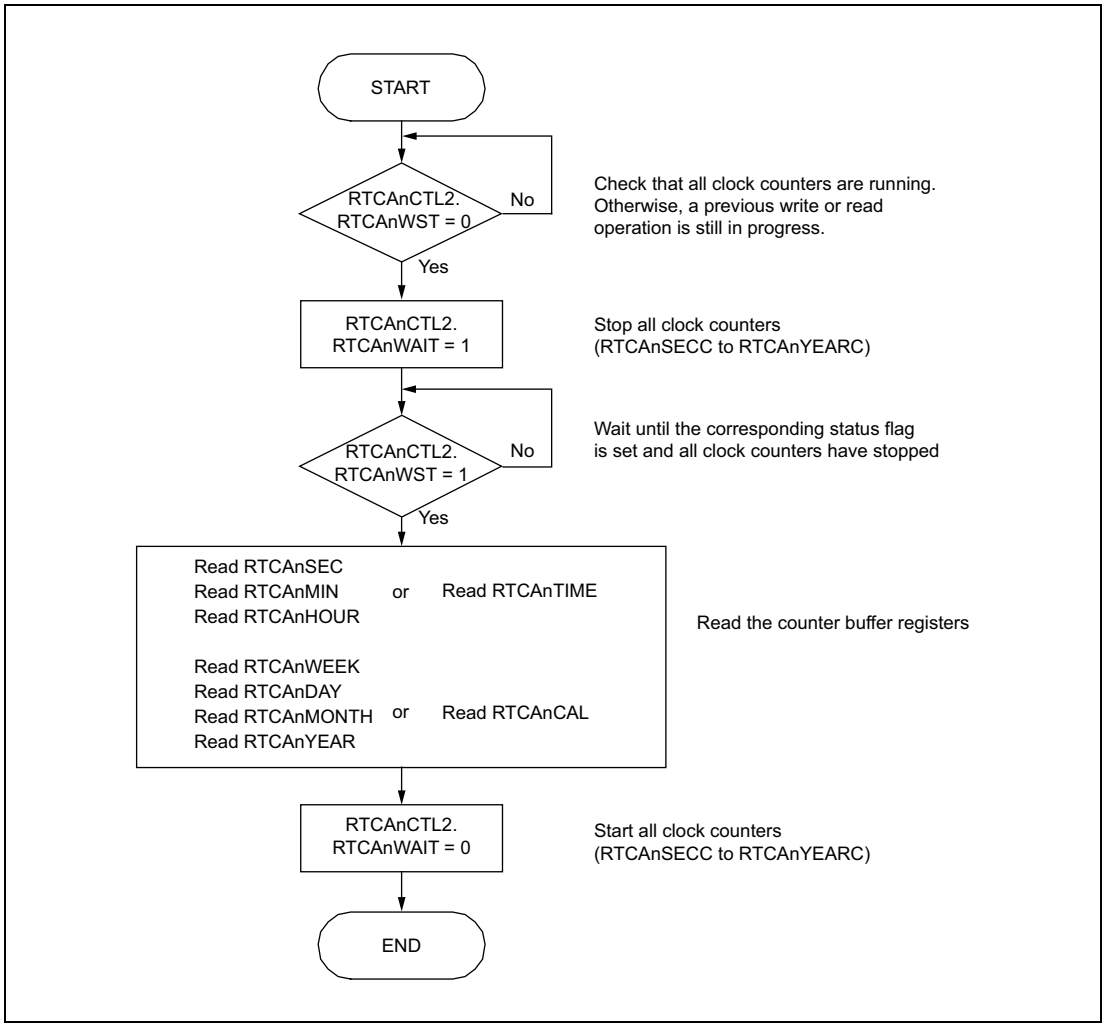


Figure 31.7 Reading Clock Count Buffer Registers

CAUTIONS

1. **The internal clock counter is synchronized with RTCATCKI.**
   In addition, two RTCATCKI periods are required before resuming counter behind END of the above flow.
   Therefore, PCLK must be continuously supplied until the counter resuming.
   Check that RTCAnCTL0.RTCAnCEST= 1 first to stop the supply of PCLK after count buffer register reading.
2. **The reading procedure must be completed within one second. Otherwise the Real-Time Clock will not count correctly any more:**
3. **Only one sub-counter overflow can be held internally. If there is a value held internally when the clock counter restarts, the seconds counter will be incremented by 1.**
4. **If the sub-counter overflows more than once during clock counter stop, the overflow count cannot be held internally. Thus the seconds counter is incremented by one instead of by two when it is restarted.**

31.5.3.2 Procedure for Reading Counter Registers Directly

To ensure that the sub-counter did not overflow while reading the counters, the seconds counter RTCAnSECC must be read twice in the beginning and at the end of the procedure. The first read value is compared with the second read value.

- First read value = second read value:
  No overflow of sub-counter occurred during counter read operation.
- First read value ≠ second read value:
  Overflow of the sub-counter occurred during counter read operation. The counters must be read again to get the current counter values.

To read the counter register directly when the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), follow the flowchart shown below.

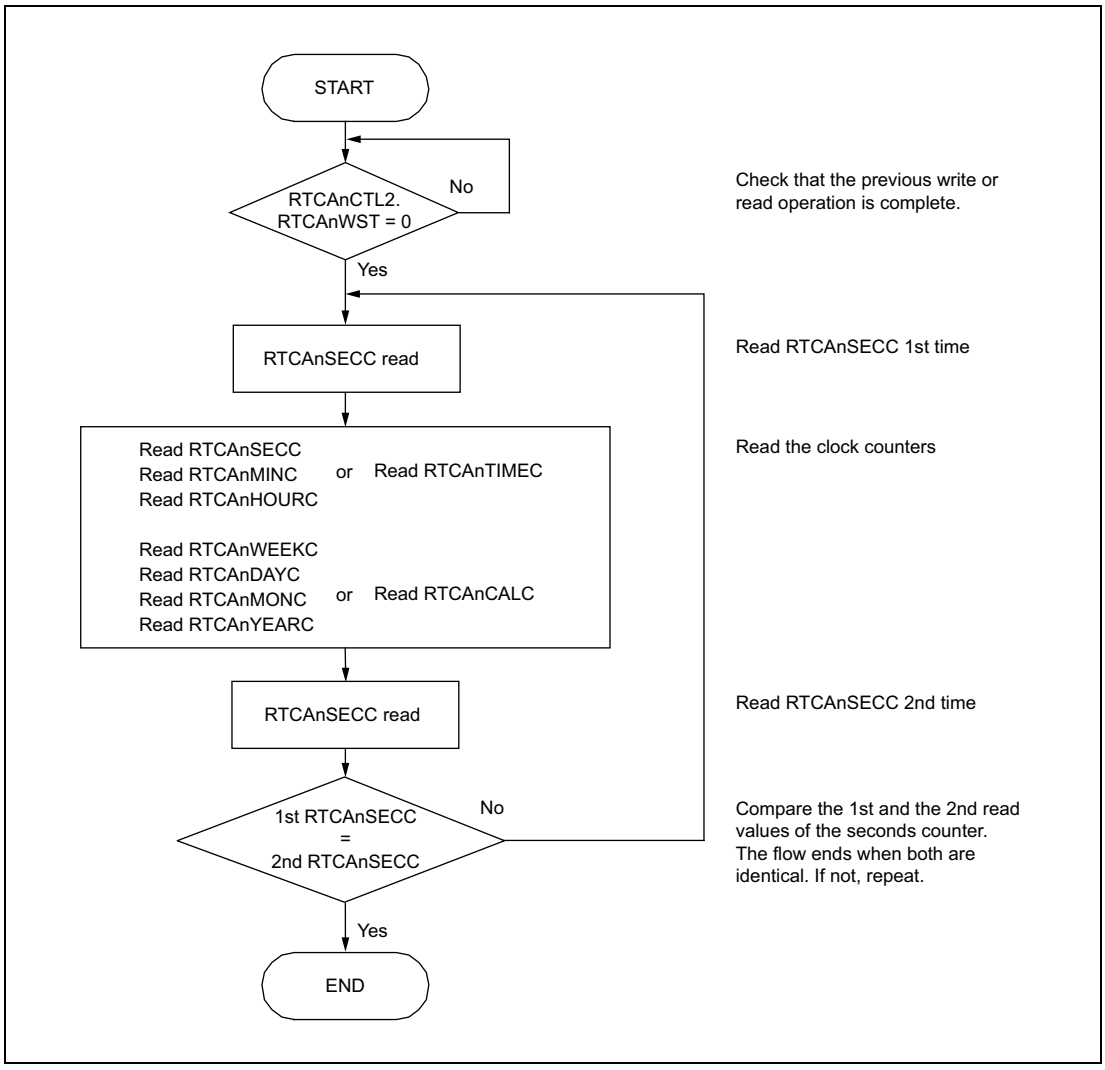


Figure 31.8 Reading Clock Counter Registers

NOTE

The procedure must be completed within one second.

31.5.4 Reading RTCAnSRBU

RTCAnSRBU is the read buffer register for the sub-counter.

When the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), read RTCAnSRBU according to the following flow.

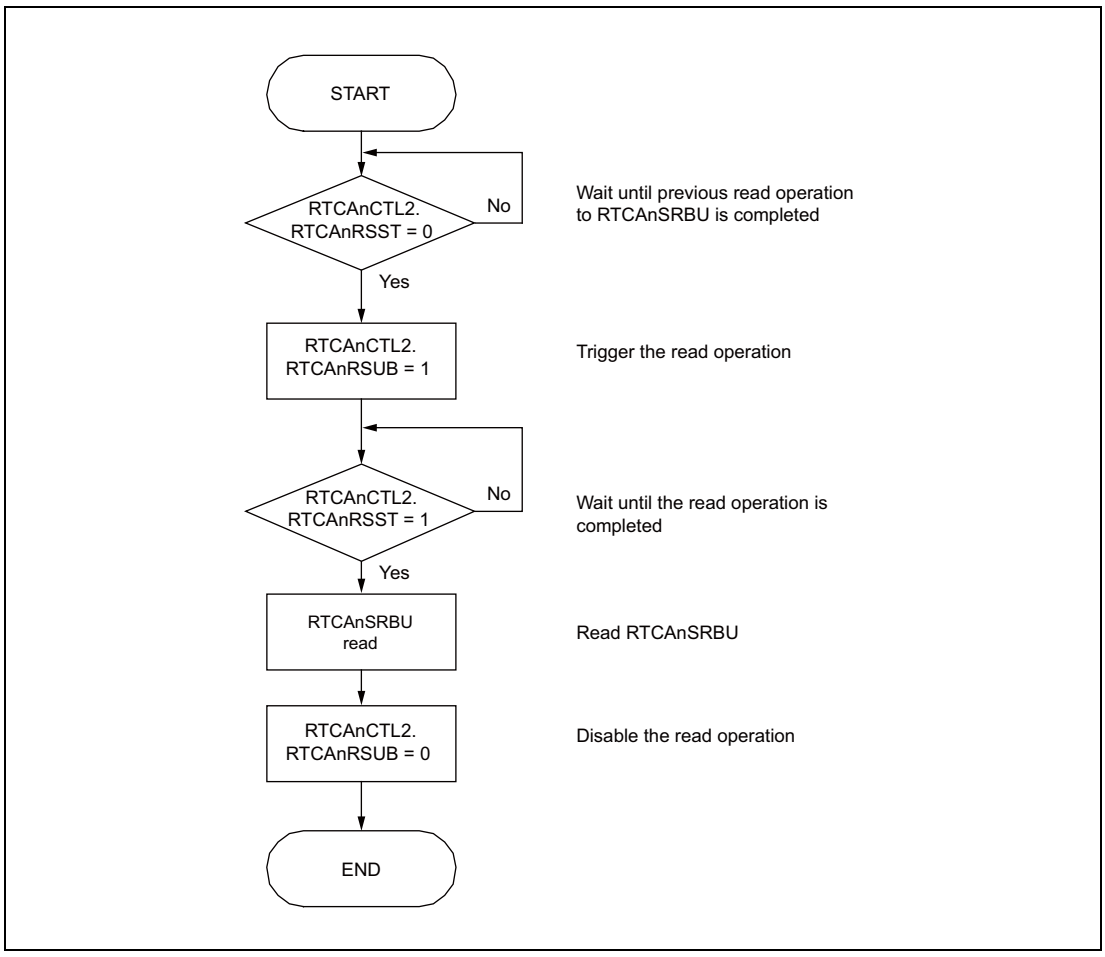


Figure 31.9 Reading the RTCAnSRBU Register

31.5.5 Writing to RTCAnSUBU

RTCAnSUBU is the clock error correction register for the sub-counter.

When the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), write to RTCAnSUBU according to the flow described below.

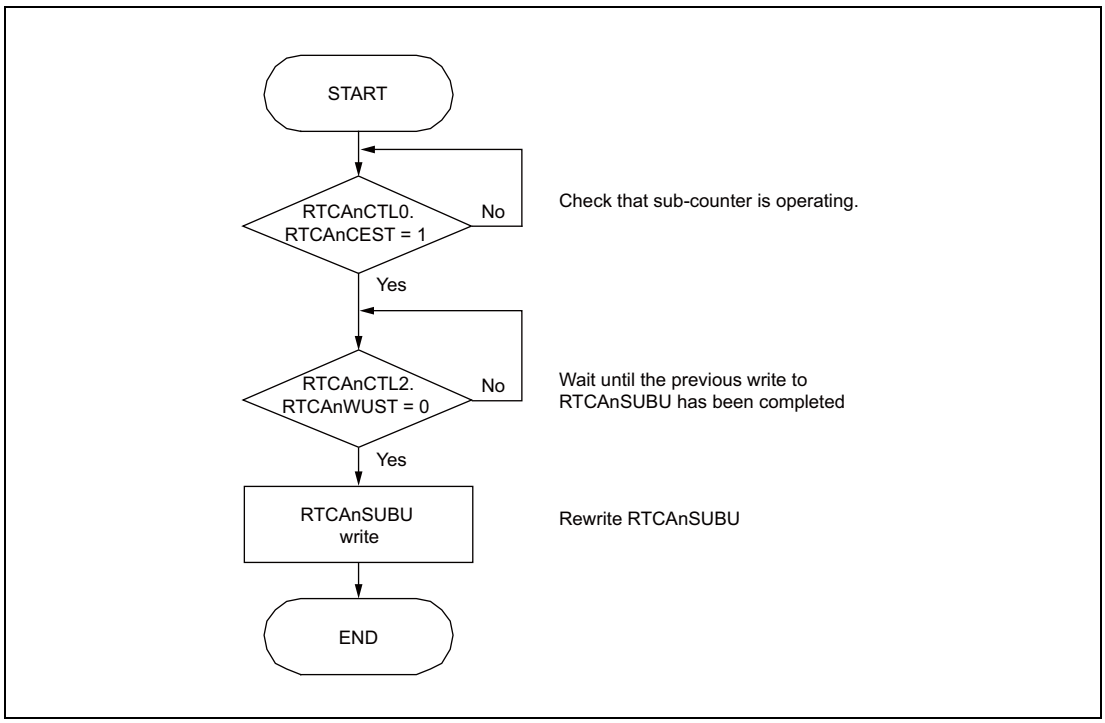


Figure 31.10 Writing to the RTCAnSUBU Register

NOTE

While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), the status flag RTCAnCTL2.RTCAnWUST is set when RTCAnSUBU is written to. It is cleared when the write operation to RTCAnSUBU is completed. This is synchronous with the next RTCAnSUBC overflow.
RTCAnCTL2.RTCAnWUST can be set for up to one second. Be careful when performing polling (checking if RTCAnCTL2.RTCAnWUST = 1 at the beginning of this flow).

31.5.6 Writing to RTCAnSCMP

RTCAnSCMP is the sub-counter compare register.

When the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), write to RTCAnSCMP according to the flow described below.

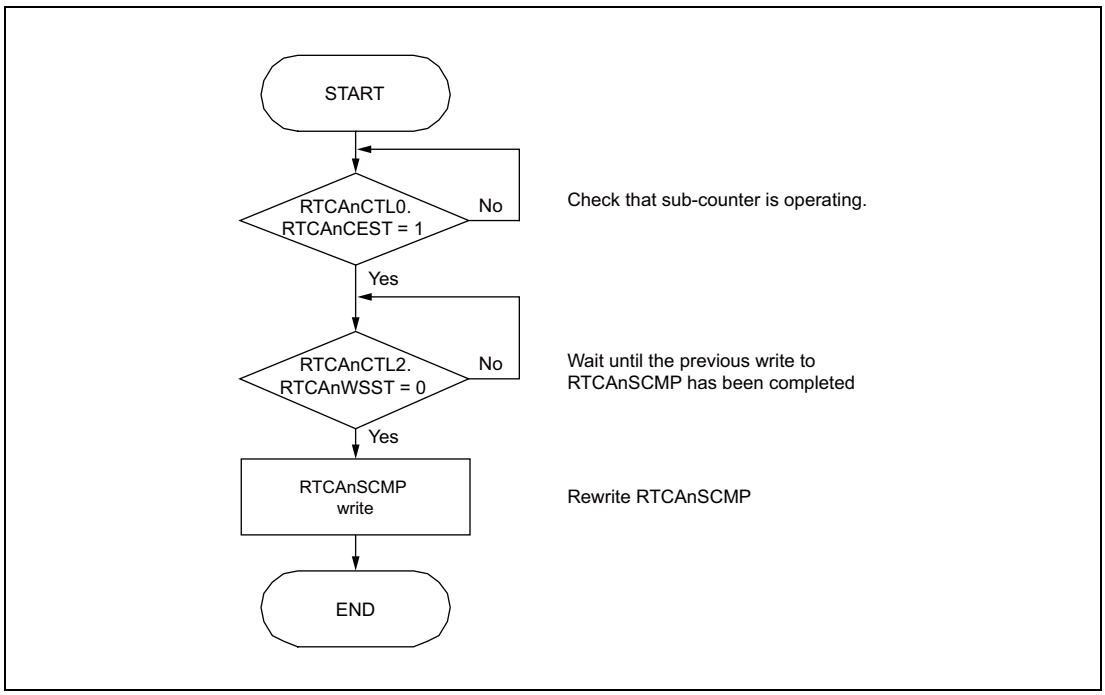


Figure 31.11 Writing the RTCAnSCMP Register

NOTE

While the sub-counter operation is enabled (RTCAnCTL0.RTCAnCE = 1), the status flag RTCAnCTL2.RTCAnWSST is set when RTCAnSCMP is written to. It is cleared when the write operation to RTCAnSCMP is completed. This is synchronous with the next RTCAnSUBC overflow.

RTCAnCTL2.RTCAnWSST can be set for up to one second. Be careful when performing polling (checking if RTCAnCTL2.RTCAnWSST = 1 at the beginning of this flow).

31.6 Timing Diagrams

31.6.1 Timing of Counter Start

The following diagram illustrates the counter start after setting the time in the buffer registers.

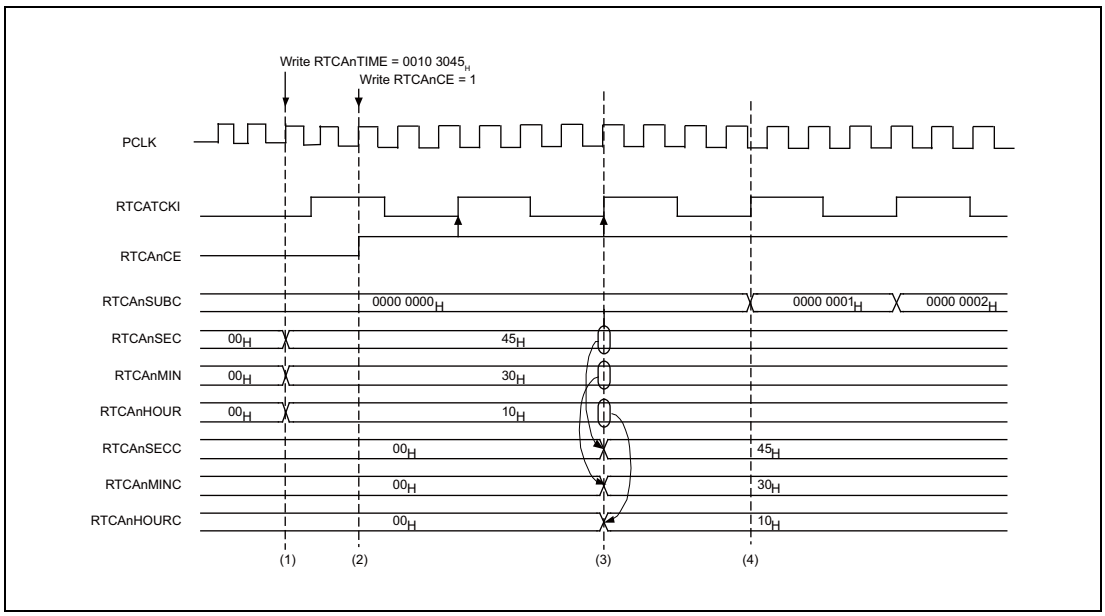


Figure 31.12 Counter Start Timing

The timing diagram above shows the following:

(1) The initial setting value of the time count buffer is set to 10:30:45 by setting RTCAnTIME = 0010 3045_H.
Count buffer registers RTCAnSEC, RTCAnMIN, and RTCAnHOUR are also automatically written.

(2) Sub-counter operation is started by setting RTCAnCTL0.RTCAnCE = 1.

(3) When the second rising edge of RTCATCKI occurs, the buffer register values are loaded to the corresponding count registers.

(4) When the next rising edge of RTCATCKI occurs, count up of the sub-counter starts.

31.6.2 Timing of Clock Counter Update while Counter Is Enabled

The following diagram illustrates the counter restart after setting the time in the buffer registers.

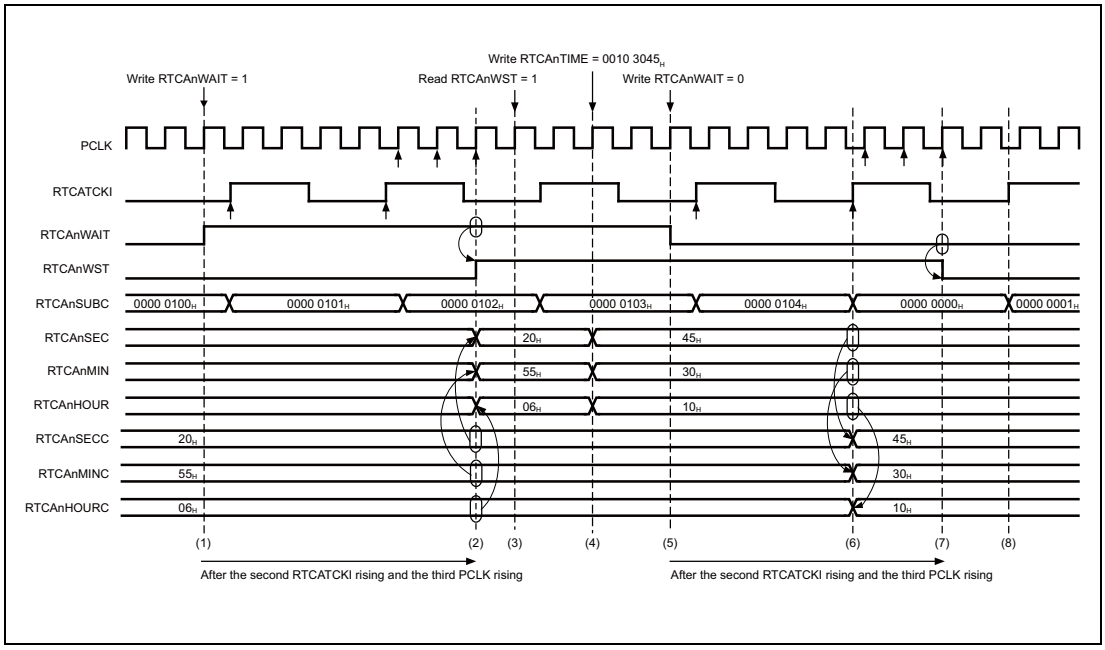


Figure 31.13 Clock Counter Update Timing

The timing diagram above shows the following:

(1) Trigger the clock counters stop (RTCAnCTL2.RTCAnWAIT = 1).

(2) RTCAnCTL2.RTCAnWST is set to 1 after the second rising edge of RTCATCKI and the third rising edge of PCLK, and the counter clock stops. The sub-counter continues counting.

(3) RTCAnCTL2.RTCAnWST = 1 can be readable.

(4) The initial setting value of the time count buffer is set to 10:30:45 by setting RTCAnTIME to 0010 3045_H.
Count buffer registers RTCAnSEC, RTCAnMIN, and RTCAnHOUR are also automatically written.

(5) Trigger the clock counters restart (RTCAnCTL2.RTCAnWAIT = 0).

(6) When the second rising edge of RTCATCKI occurs, the values of the buffer registers are loaded to the corresponding count registers. Write operation to RTCAnSECC is performed and RTCAnSUBC is cleared.

(7) When the third rising edge of PCLK occurs, RTCAnCTL2.RTCAnWST is set to 0.

(8) Clock counter operation is resumed.

31.6.3 Timing of Sub-Counter Read Buffer Reading while Counter is Enabled

The following diagram illustrates the timing when reading the sub-counter read buffer RTCAnSRBU.

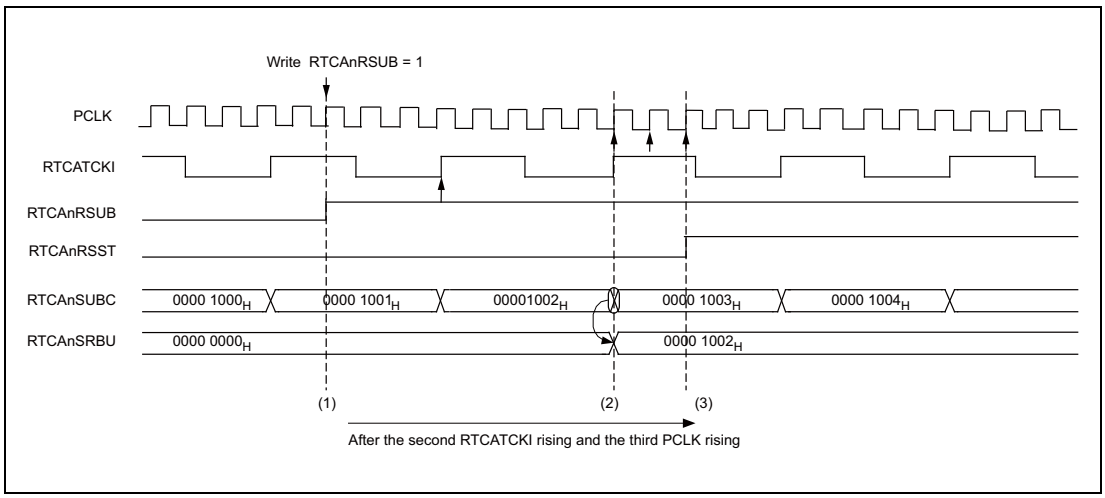


Figure 31.14 Timing when Reading the Sub-Counter Read Buffer Register Value

The timing diagram above shows the following:

(1) Setting RTCAnRSUB = 1 triggers loading of the sub-counter value to RTCAnSRBU.

(2) When the second rising edge of RTCATCKI occurs, the value of RTCAnSUBC is loaded to RTCAnSRBU.

(3) When the third rising edge of PCLK occurs, RTCAnCTL2.RTCAnRSST is set to 1 and RTCAnSRBU can be read.

Section 32 PWM Generators and Diagnostic (PWM-Diag)

This section contains a generic description of the PWM output/diagnostic function (PWM-Diag).

The first part of this section describes all RH850/D1L/D1M specific properties, such as the number of units, register base addresses, etc. The remainder of the section describes the functions and registers of the units constituting PWM-Diag.

32.1 Features of RH850/D1L/D1M PWM-Diag

32.1.1 Number of Units and Channels

The PWM-Diag unit consists of a PWBA block for generating clock signals, PWGA blocks that generate PWM signals, and a PWSA block for generating triggers for A/D conversion. The numbers of individual units are listed below.

Each PWGA unit has one PWM channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 32.1 Number of Units

Product Name	D1L1	D1L2(H)	D1M1(H) D1M1-V2 D1M1A	D1M2	D1M2H
PWBA					
Number of Units			1		
Name			PWBAn (n = 0)		
PWGA					
Number of Units			24		
Name			PWGAn (n = 0 to 23)		
PWSA					
Number of Units			1		
Name			PWSAn (n = 0)		

Table 32.2 Index

Index	Meaning
n	Throughout this section, individual units constituting the PWM-Diag function are identified by the index “n”; for example, PWBAnTE indicates the PWBAn status register.
m	The PWBA generation clock is identified by the index “m” (m = 0 to 3); for example, PWBAnBRSm indicates the PWMCLKm clock cycle configuration register.
x, y	An A/D converter configuration register number corresponding to a PWM-Diag channel is identified by the index “x, y”; for example, PWSAnPVCRx_y (x_y = 00_01, 02_03, ..., 10_11).
j	Registers storing trigger channel numbers (encoded value) from PWGAn are identified by the index “j”; for example, the PWSAnQUEj register (j = 0 to 7).
k	Sets of registers where each has the same function are identified by the index “k”; for example, the SLPWGAk register (k = 0).

The following table shows values indicated by the indexes of each product.

Table 32.3 Indexes of Products

Indexes of each product				
D1L1	D1L2(H)	D1M1(H) D1M1A	D1M2	D1M2H
x = 00, 02, ,,, 10 y = 01, 03, ,,, 11	x = 00, 02, ,,, 10 y = 01, 03, ,,, 11	x = 00, 02, ,,, 10 y = 01, 03, ,,, 11	x = 00, 02, ,,, 10 y = 01, 03, ,,, 11	x = 00, 02, ,,, 10 y = 01, 03, ,,, 11
j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7	j = 0 to 7
k = 0	k = 0	k = 0	k = 0	k = 0

32.1.2 Register Base Address

PWM-Diag base addresses are listed in the following table.

PWM-Diag register addresses are given as offsets from the base addresses in general.

Table 32.4 Register Base Addresses

Base Address Name	Base Address
<PWBA0_base>	FFE4 5800_H
<PWGAn_base>	FFE4 4000_H + 100_H x n
<PWSA0_base>	FFE4 $5A00_H$

32.1.3 Clock supply

Clock supply by and to PWM-Diag is listed in the following table.

Table 32.5 Clock Supply

Unit Name	Unit Clock Name	Internal Clock Signal
PWBA0	PCLK	Clock Controller CLKJIT
PWGAn	PCLK	Clock Controller CLKJIT
PWSA0	PCLK	Clock Controller CLKJIT

32.1.4 Interrupt Requests

PWM-Diag interrupt requests are listed in the following table.

Table 32.6 TAUJn interrupt requests

PWM-Diag signals	Function	Connected to
PWGA_INT0 to PWGA_INT23	Channel 0 to 23 interrupt	Interrupt Controller INTPWGA0 to INTPWGA23
PWSA_INT_QFULL	PWSA queue full interrupt	Interrupt Controller INTQFULL

32.1.5 Hardware Reset

PWM-Diag reset sources are listed in the following table. The individual PWM-Diag units are initialized by these reset sources.

The PWM-Diag and its registers are initialized by the following reset signals

Table 32.7 Reset Sources

Unit Name	Reset Source
PWBA0 PWGAn PWSA0	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

32.1.6 External Input/Output Signals

The external input/output signals of the PWM-Diag are listed below.

Table 32.8 External Input/Output Signals

Unit Signal Name	Outline	Connected to
PWGA_TOUTn (Unit: PWGA)	PWGA unit n output	Port PWGAnO

32.1.7 Internal signals

The signals for connecting two PWM-Diag channels or a PWM-Diag and another function are listed below.

Table 32.9 Internal Input/Output Signals

Unit Signal Name	Outline	Signal
PWBA0		
PWMCLK0	PWGA count clock 0	PWGAn
PWMCLK1	PWGA count clock 1	PWGAn
PWMCLK2	PWGA count clock 2	PWGAn
PWMCLK3	PWGA count clock 3	PWGAn
PWGAn		
PWGA_TRGOUT[11:0]	PWGAn trigger	PWSA0
PWSA0		
PWSA_ADTRG[1:0]	A/D converter unit select signal	ADCE0
PWSA_PVCR_VALUE[11:0] (PWSA0)	A/D converter control signal	ADCE0
ADCE0		
ADC_CONV_END0	A/D conversion completion signal	PWSA0

32.1.8 Outline of Functions

This function is comprised of four types of units: clock divider (PWBA), PWM generator (PWGA), A/D conversion trigger select function (PWSA), and A/D converter (ADCE).

PWBA

- Clock divider
  PWBA generates a PWMCLKm count clock signal by frequency division of PCLK and supplies it to the PWM generator PWGA.
  The cycle of the PWMCLKm count clock signal can be calculated from the setting of the PWBAnBRSm register by the equation below.

  PWMCLKm count clock cycle = (PWBAnBRSm value × 2) × PCLK cycle

  In addition, PWBA can control operation when the on-chip debugger is in use by using the PWBAnEMU register.

PWGA

PWGA outputs PWM waveforms and A/D conversion trigger to PWSA by using the input clock PWMCLKm from PWBA.

- PWM waveform output PWGA_TOUTn
  This generator outputs PWM waveforms from the PWGA_TOUTn pin. The PWM cycle is the full count cycle of the PWGAnCNT register (12-bit free-running counter). Set the high-level period of PWM output in the PWGAnCSDR and PWGAnCRDR registers.

  The PWM waveform cycle and duty can be calculated by the equations below.

  PWM waveform cycle = PWGAnCNT (12-bit full count: FFF_H + 1)
  × Count clock cycle
  = 4096 × PWMCLKm count clock cycle

  When PWGAnCRDR[11:0] > PWGAnCSDR[11:0],

  High-level period of PWM waveform =
  (PWGAnCRDR register value – PWGAnCSDR register value)
  × PWMCLKm count clock cycle

  PWM waveform duty (%) = High-level period of PWM waveform/PWM waveform cycle × 100 = (PWGAnCRDR register value – PWGAnCSDR register value)/4096 × 100

  Note that the PWM output is fixed to the low level when the PWGAnCRDR register value is equal to the PWGAnCSDR register value.

  When $1xxx_H$ is set in the PWGAnCRDR register (i.e. bit 12 is set to 1), the PWM output is fixed to the high level.

- A/D conversion trigger output PWGA_TRGOUTn
  The A/D conversion trigger signal PWGA_TRGOUTn for PWSA is generated when the PWGAnCTDR register value and the PWGAnCNT register value match while the PWM output PWGA_TOUTn is at the high level.
  The timing can be calculated by the equation below.

  A/D conversion trigger signal generation timing = PWGAnCTDR register value
  × PWMCLKm count clock cycle

- PWGA interrupt request signal PWGA_INTn
  PWGA generates the interrupt request signal PWGA_INTn at the falling edge of the PWM output PWGA_TOUTn.
  When the PWM output is fixed to the low level, PWGA_INTn is generated when the PWGAnCRDR register value and the PWGAnCNT register value (free-running counter value) match; when the PWM output is fixed to the high level, PWGA_INTn is generated at the timing of overflow of the PWGAnCNT register.

NOTE

Only the PWM Generators PWGA0 to PWGA11 can trigger the A/D Converter for diagnosis of their PWM output signals PWGA_TOUT0 to PWGA_TOUT11.

PWSA

PWSA transmits the required settings information to the A/D converter and outputs the A/D conversion start trigger, based on the A/D conversion trigger signal PWGA_TRGOUTn from the PWM generator (PWGA).

- A/D conversion control by PWSA
  PWSA outputs the information required for the A/D conversion, which is set in the corresponding PWSAnPVCRx_y register for the channel number of the trigger input from PWGAn, (i.e., information on ADC physical channel, external MPX control, and error detection level selection) is output to the A/D converter.
  At the same timing, A/D conversion trigger (PWSA_ADTRG) is output to ADCE0. (A maximum of eight input trigger signal PWGA_TRGOUTn information received during A/D conversion are stored and kept in PWSAnQUE.)
  The setting information to be output to the A/D converter is kept until the next trigger is generated.
  When the A/D conversion triggered by the PWM-Diag function is completed in the A/D converter, PWSA triggers the next A/D conversion based on the data stored in the PWSAnQUE register.
- Queuing of A/D conversion trigger from PWGA
  The A/D conversion trigger signal (PWGA_TRGOUTn) input from PWGAn is stored in the PWSAnQUEj register as a channel number. The PWSAnQUEj register stores a maximum of eight channel numbers of the A/D conversion trigger signal PWGA_TRGOUTn received during A/D conversion in a queue structure.
  A PWSA queue full interrupt occurs in the following states, when the queue of the PWSAnQUEj register becomes full:
  - A trigger number is written to PWSAnQUE7
  - A trigger number has already been written to PWSAnQUE7 and cannot be written when PWGA_TRGOUTn is input.

ADCE

A/D conversion is executed upon receipt of information required for A/D conversion and A/D conversion trigger from PWSA.

A/D conversion is executed using the PWM-Diag-dedicated scan group; on completion of the A/D conversion, it is reported to the PWSA.

For the basic operation of the A/D converter, see section **A/D Converter (ADCE)**.

For the A/D converter operation with the PWM-Diag function, see subsection **A/D Conversion with PWM-Diag Enabled**.

32.1.9 Block Diagram

The following figure shows an example of connecting the LED control circuit combining the PWM-Diag and the A/D converter.

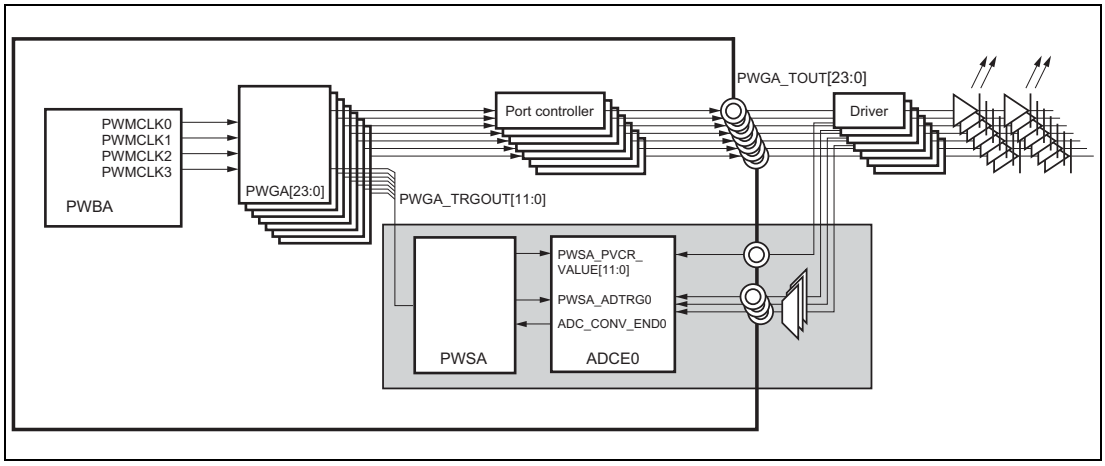


Figure 32.1 Example of Connecting the LED Control Circuit using the PWM-Diag and the A/D Converter

32.2 Registers

32.2.1 List of Registers

PWM-Diag registers are listed in the following table.

Table 32.10 List of PMW-Diag Registers

Module name	Register name	Symbol	Address
PWBAn	PWMCLKm cycle configuration register	PWBAnBRSm	<PWBAn_base> + 0004_H × m
	PWMCLKm enable status register	PWBAnTE	<PWBAn_base> + 0010_H
	PWMCLKm start trigger register	PWBAnTS	<PWBAn_base> + 0014_H
	PWMCLKm stop trigger register	PWBAnTT	<PWBAn_base> + 0018_H
	PWBA emulation register	PWBAnEMU	<PWBAn_base> + $001C_H$
PWGAn	PWM output set condition register	PWGAnCSDR	<PWGAn_base> + 0000_H
	PWM output reset condition register	PWGAnCRDR	<PWGAn_base> + 0004_H
	PWGA_TRGOUTn generation condition register	PWGAnCTDR	<PWGAn_base> + 0008_H
	Buffer register reload trigger register	PWGAnRDT	<PWGAn_base> + $000C_H$
	Buffer register reload status register	PWGAnRSF	<PWGAn_base> + 0010_H
	PWM cycle count register	PWGAnCNT	<PWGAn_base> + 0014_H
	PWGA control register	PWGAnCTL	<PWGAn_base> + 0020_H
	PWGAnCSDR buffer register	PWGAnCSBR	<PWGAn_base> + 0024_H
	PWGAnCRDR buffer register	PWGAnCRBR	<PWGAn_base> + 0028_H
	PWGAnCTDR buffer register	PWGAnCTBR	<PWGAn_base> + $002C_H$
SELB	PWGA synchronous trigger register	SLPWGAk	FFE4 5900_H + k × 4_H
PWSAn	PWSA control register	PWSAnCTL	<PWSAn_base> + 0000_H
	Trigger queue status register	PWSAnSTR	<PWSAn_base> + 0004_H
	Trigger queue status clear register	PWSAnSTC	<PWSAn_base> + 0008_H
	Trigger queue register	PWSAnQUEj	<PWSAn_base> + 0020_H + j × 4_H
	PWM-Diag mode A/D setting register	PWSAnPVCRx_y	<PWSAn_base> + 0040_H + x × 2_H
	PWSA emulation control register	PWSAnEMU	<PWSAn_base> + $000C_H$

The base addresses <PWBAn_base>, <PWGAn_base> and <PWSAn_base> are defined in the first subsection of this section under the key word “Register base addresses”.

32.2.1.1 PWBAnBRSm Register

This register sets the clock cycle of PWMCLKm.

Access: This register can be read/written in 16-bit units.

Address: <PWBAn_base> + 0004_H × m

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	PWBAnBRSm[10:0]										
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.11 PWBAnBRSm Register Contents

Bit Position	Bit Name	Function
15 to 11	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
10 to 0	PWBAnBRSm [10:0]	Register for setting the clock cycle of PWMCLKm. – PWBAnBRSm = 0: PWMCLKm stop – PWBAnBRSm = 1: PWMCLKm = PCLK / 2 × 1 – PWBAnBRSm = 2: PWMCLKm = PCLK / 2 × 2 ... – PWBAnBRSm = n: PWMCLKm = PCLK / 2 × n (n = 1 to 2047) These bits can only be rewritten when all counters using PWMCLKm are stopped (PWBAnTE.PWBATEm = 0).

32.2.1.2 PWBAnTE Register

This is a status register that indicates the output status of PWMCLKm (m = 0 to 3).

Access: This register can be read in 8-bit units.

Address: <PWBAn_base> + 0010_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTE3	PWBAnTE2	PWBAnTE1	PWBAnTE0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 32.12 PWBAnTE Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When read, the value after reset is returned.
3	PWBAnTE3	A status flag indicating the operation status of PWMCLK3 0: Not operating 1: Operating
2	PWBAnTE2	A status flag indicating the operation status of PWMCLK2 0: Not operating 1: Operating
1	PWBAnTE1	A status flag indicating the operation status of PWMCLK1 0: Not operating 1: Operating
0	PWBAnTE0	A status flag indicating the operation status of PWMCLK0 0: Not operating 1: Operating

32.2.1.3 PWBAnTS Register

This register is a start trigger register for PWMCLKm (m = 0 to 3).

Access: This register can be written in 8-bit units.

Address: <PWBAn_base> + 0014_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTS3	PWBAnTS2	PWBAnTS1	PWBAnTS0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 32.13 PWBAnTS Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing to these bits, write the value after reset.
3	PWBAnTS3	Start trigger bit for PWMCLK3 0: Writing 0 does not work as a function. 1: Starts the output of PWMCLK3.
2	PWBAnTS2	Start trigger bit for PWMCLK2 0: Writing 0 does not work as a function. 1: Starts the output of PWMCLK2.
1	PWBAnTS1	Start trigger bit for PWMCLK1 0: Writing 0 does not work as a function. 1: Starts the output of PWMCLK1.
0	PWBAnTS0	Start trigger bit for PWMCLK0 0: Writing 0 does not work as a function. 1: Starts the output of PWMCLK0.

32.2.1.4 PWBAnTT Register

This register is a stop trigger register for PWMCLKm (m = 0 to 3).

Access: This register can be written in 8-bit units.

Address: <PWBAn_base> + 0018_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PWBAnTT3	PWBAnTT2	PWBAnTT1	PWBAnTT0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W

Table 32.14 PWBAnTT Register Contents

Bit Position	Bit Name	Function
7 to 4	Reserved	When writing to these bits, write the value after reset.
3	PWBAnTT3	Stop trigger bit for PWMCLK3 0: Writing 0 does not work as a function. 1: Stops the output of PWMCLK3.
2	PWBAnTT2	Stop trigger bit for PWMCLK2 0: Writing 0 does not work as a function. 1: Stops the output of PWMCLK2.
1	PWBAnTT1	Stop trigger bit for PWMCLK1 0: Writing 0 does not work as a function. 1: Stops the output of PWMCLK1.
0	PWBAnTT0	Stop trigger bit for PWMCLK0 0: Writing 0 does not work as a function. 1: Stops the output of PWMCLK0.

32.2.1.5 PWBAnEMU Register

This register sets the operation during emulation.

Access: This register can be read/written in 8-bit units.

Address: <PWBAn_base> + $001C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	PWBAnSVSDIS	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 32.15 PWBAnEMU Register Contents

Bit Position	Bit Name	Function
7	PWBAnSVSDIS	(When the EPC.SVSTOP bit = 0) The count clock is provided continuously when the debugger has control of the microcontroller (by break points, etc.), regardless of the value of this bit (1 or 0). (When the EPC.SVSTOP bit = 1) 0: The count clock is stopped when the debugger has control of the microcontroller (by break points, etc.). 1: The count clock is provided continuously when the debugger has control of the microcontroller (by break points, etc.). This bit can only be rewritten when all counters using PWMCLKm are stopped (PWBAnTE.PWBATEm = 0).
6 to 0	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.

32.2.1.6 PWGAnCTL — PWGA Control Register

PWGAnCTL is used to select the count clock from PWBA.

Access: This register can be read/written in 8-bit units.

Address: <PWGAn_base> + 0020_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWGAnCKS[1:0]	
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W

Table 32.16 PWGAnCTL Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
1, 0	PWGAnCKS [1:0]	Count Clock Enable Input PWMCLK3 to PWMCLK0 Select 00: Uses PWMCLK0 as count clock 01: Uses PWMCLK1 as count clock 10: Uses PWMCLK2 as count clock 11: Uses PWMCLK3 as count clock These bits can only be rewritten when the PWGAn operation is stopped (SLPWGAk.SLPWGA[31:0] = 0).

32.2.1.7 PWGAnCNT — PWM Cycle Count Register

This is a count register.

Access: This register can be read in 16-bit units.

Address: <PWGAn_base> + 0014_H

Value after reset: $0FFF_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCNT[11:0]											
Value after reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 32.17 PWGAnCNT Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned.
11 to 0	PWGAnCNT [11:0]	12-bit counter value

32.2.1.8 PWGAnCSDR — PWM Output Set Condition Register

This register sets the setting condition for PWGA_TOUTn output.

Access: This register can be read/written in 16-bit units.

Address: <PWGAn_base> + 0000_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCSDR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.18 PWGAnCSDR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
11 to 0	PWGAnCSDR [11:0]	These bits set the setting condition for PWM output. The set value is reflected to the PWGAnCSBR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1).

32.2.1.9 PWGAnCRDR — PWM Output Reset Condition Register

This register sets the reset condition for PWGA_TOUTn output.

Access: This register can be read/written in 16-bit units.

Address: <PWGAn_base> + 0004_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PWGAnCRDR[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.19 PWGAnCRDR Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
12 to 0	PWGAnCRDR [12:0]	These bits set the reset condition for PWM output. The set value is reflected to the PWGAnCRBR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1).

32.2.1.10 PWGAnCTDR — PWGA_TRGOUTn Generation Condition Register

This register sets the generation condition for PWGA_TRGOUTn.

Access: This register can be read/written in 16-bit units.

Address: <PWGAn_base> + 0008_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCTDR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.20 PWGAnCTDR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
11 to 0	PWGAnCTDR [11:0]	These bits set the A/D conversion trigger generation condition for PWSAn. The set value is reflected to the PWGAnCTBR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1).

32.2.1.11 PWGAnCSBR — PWGAnCSDR Buffer Register

This is a buffer register for the PWGAnCSDR register.

Access: This register can be read in 16-bit units.

Address: <PWGAn_base> + 0024_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCSBR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 32.21 PWGAnCSBR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned.
11 to 0	PWGAnCSBR [11:0]	The set value is reflected to the PWGAnCSDR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, the pin output is driven high.

32.2.1.12 PWGAnCRBR — PWGAnCRDR Buffer Register

This is a buffer register for the PWGA_TOUTn reset condition.

Access: This register can be read in 16-bit units.

Address: <PWGAn_base> + 0028_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	PWGAnCRBR[12:0]												
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 32.22 PWGAnCRBR Register Contents

Bit Position	Bit Name	Function
15 to 13	Reserved	When read, the value after reset is returned.
12 to 0	PWGAnCRBR [12:0]	The set value is reflected to the PWGAnCRDR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, the pin output is driven low.

32.2.1.13 PWGAnCTBR — PWGAnCTDR Buffer Register

This is a buffer register for the PWGA_TRGOUTn generation condition.

Access: This register can be read in 16-bit units.

Address: <PWGAn_base> + $002C_H$

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PWGAnCTBR[11:0]											
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 32.23 PWGAnCTBR Register Contents

Bit Position	Bit Name	Function
15 to 12	Reserved	When read, the value after reset is returned.
11 to 0	PWGAnCTBR [11:0]	The set value is reflected to the PWGAnCTDR register at the start of PWGAn operation (SLPWGAk.SLPWGA of the corresponding CH = 1) or when a simultaneous rewrite is performed (PWGAnRDT.PWGAnRDT = 1). When the value matches the PWGAnCNT register value, a trigger is transmitted to PWSAn.

32.2.1.14 PWGAnRSF — Buffer Register Reload Status Register

This register is a status register for simultaneous rewrite control.

Access: This register can be read in 8-bit units.

Address: <PWGAn_base> + 0010_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWGAnRSF
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 32.24 PWGAnRSF Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	PWGAnRSF	Simultaneous Rewrite Control Status 0: Simultaneous rewrite is enabled. This value indicates the completion of simultaneous rewrite after the generation of a simultaneous rewrite trigger signal. 1: Simultaneous rewrite is in progress. This value indicates the waiting state for completion.

32.2.1.15 PWGAnRDT — Buffer Register Reload Trigger Register

This is a simultaneous rewrite request trigger register.

Access: This register can be written in 8-bit units.

Address: <PWGAn_base> + $000C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWGAnRDT
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 32.25 PWGAnRDT Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When writing to these bits, write the value after reset.
0	PWGAnRDT	Simultaneous Rewrite Request Trigger 0: Writing 0 does not work as a function. 1: Triggers the simultaneous rewrite request for the compare registers (PWGAnCSDR, PWGAnCRDR, and PWGAnCTDR), and sets PWGAnRSF.PWGAnRSF to 1.

32.2.1.16 SLPWGAk — PWGA Simultaneous Trigger Register (k = 0)

This register triggers start and stop for respective channels simultaneously.

Access: This register can be read/written in 32-bit units.

Address: FFE4 5900_H + k × 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SLPWGA[23:16]							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SLPWGA[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.26 SLPWGAk Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When writing to these bits, write the value after reset.
23 to 0	SLPWGA [23:0]	Trigger start and stop to multiple channels simultaneously. 0: Stops the corresponding channels. 1: Starts the corresponding channels. The bits correspond to the following channels. SLPWGA0.SLPWGA[23:0]: PWGA23 - PWGA0

32.2.1.17 PWSAnCTL Register

This register is used to control operations of PWSA.

Access: This register can be read/written in 8-bit units.

Address: <PWSAn_base> + 0000_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWSAnENBL
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 32.27 PWSAnCTL Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	PWSAnENBL	Operation Permission Control 0: Operation is prohibited (initial state). Writing 0 initializes PWSAnSTR and PWSAnQUEj. 1: Operation is enabled.

32.2.1.18 PWSAnSTR Register

This is a status register that indicates whether the number of a channel for which an A/D conversion trigger has been generated is stored in a PWSAnQUEj register.

Access: This register can be read in 8-bit units.

Address: <PWSAn_base> + 0004_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWSAnQFL	PWSAnQNE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 32.28 PWSAnSTR Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is returned.
1	PWSAnQFL	Indicates the queuing state of the A/D conversion trigger. 0: Some PWSAnQUEj registers do not store the channel number. 1: All of the PWSAnQUEj registers store the channel number.
0	PWSAnQNE	Bit indicating that there is a trigger in the trigger queue 0: The channel number is not stored in the PWSAnQUEj register, or the A/D conversion is in process while only PWSAnQUE0 stores the channel number. 1: The number of the channel waiting for conversion is stored in j = 1 and subsequent PWSAnQUEj registers.

32.2.1.19 PWSAnSTC Register

This register clears the status of the PWSAnSTR register.

Access: This register can be written in 8-bit units.

Address: <PWSAn_base> + 0008_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PWSAnCLFL	PWSAnCLNE
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	W

Table 32.29 PWSAnSTC Register Contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When writing to these bits, write the value after reset.
1	PWSAnCLFL	PWSAnQFL Clear 0: PWSAnQFL retains the status (Writing 0 does not work as a function). 1: PWSAnQFL is cleared to 0.
0	PWSAnCLNE	PWSAnQNE Clear 0: PWSAnQNE retains the status (Writing 0 does not work as a function). 1: PWSAnQNE is cleared to 0.

32.2.1.20 PWSAnQUEj (j = 0 to 7) Register

This register stores the channel number that received the trigger from PWGAn.

Access: This register can be read in 8-bit units.

Address: <PWSAn_base> + $0020_H + j \times 4_H$

Value after reset: $3F_H$

Bit	7	6	5	4	3	2	1	0
	—	—	PWSAnQUEj[5:0]					
Value after reset	0	0	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R

Table 32.30 PWSAnQUEj Register Contents

Bit Position	Bit Name	Function
7, 6	Reserved	When read, the value after reset is returned.
5 to 0	PWSAnQUEj [5:0]	Stores the channel number (0 to 23) of the trigger-generated PWGA in order from PWSAnQUE0 to PWSAnQUE7. After the A/D conversion of PWSAnQUE0 is completed, the values in PWSAnQUE1 to PWSAnQUE07 shift to PWSAnQUE0 to PWSAnQUE6.

NOTE

If a trigger occurs simultaneously in multiple channels, the trigger with the smaller channel number has priority.

32.2.1.21 PWSAnPVCRx_y (x = 00, 02, 04 ... 10, y = 01, 03, 05 ... 11) Register

This register is used to set the corresponding A/D converter for each channel.

Consecutive two channels are set such as PWSA0PVCR02_03, and the 16 higher-order bits of each register correspond to an odd-numbered channel while the 16 lower-order bits correspond to an even-numbered channel.

At the generation of a trigger, the setting value is transmitted to the ADCEnPWDVCR register of the A/D converter.

For the ADCEnPWDVCR register, see. Section 45, A/D Converter (ADCE).

Access: This register can be read/written in 32-bit units.

Address: <PWSAn_base> + 0040_H + x × 2_H

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	PWSAnVRDTy [7:6]		PWSAnVRDTy[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	PWSAnVRDTx [7:6]		PWSAnVRDTx[5:0]					
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32.31 PWSAnPVCRx_y Register Contents

Bit Position	Bit Name	Function
31 to 24	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
23, 22	PWSAnVRDTy [7:6]	This bit indicates the setting value of the ADCEnPWDVCR.ULS[1:0] bits (odd-numbered channel).
21 to 16	PWSAnVRDTy [5:0]	This bit indicates the setting value of the ADCEnPWDVCR.GCTRL[5:0] bits (odd-numbered channel).
15 to 8	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
7, 6	PWSAnVRDTx [7:6]	These bits indicate the setting value of the ADCEnPWDVCR.ULS[1:0] bits. (even-numbered channel)
5 to 0	PWSAnVRDTx [5:0]	These bits indicate the setting value of the ADCEnPWDVCR.GCTRL[5:0] bits. (even-numbered channel)

32.2.1.22 PWSAnEMU — Emulation Control Register

This register is used to set the operation for emulation.

Access: This register can be read/written in 8-bit units.

Address: <PWSAn_base> + $000C_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWSAnSVSDIS
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 32.32 PWSAnEMU Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	PWSAnSVSDIS	(When the EPC.SVSTOP bit = 0) The operation continues when the debugger has control of the microcontroller (by break points, etc.), regardless of the value of this bit (1 or 0). (When the EPC.SVSTOP bit = 1) 0: When the debugger has control of the microcontroller (by break points, etc.); – The output state to A/D is retained, the ADC_CONV_ENDn input at a break point is internally retained, and PWSAnQUEj is updated after break release. – The PWGA_TRGOUT input is accepted even at a break, and PWSA_INT_QFULL is even output. – Reading and writing to the register is possible. 1: The operation continues when the debugger has control of the microcontroller (by break points, etc.). The above bit can only be rewritten when all counters using PWMCLKm are stopped (PWBAnTE.PWBATEm = 0), the operation of all channels PWGAn has stopped (SLPWGAk.SLPWGA), and no trigger has been generated from any of the channels PWGAn (PWSAnQUE0 is the value after reset).

32.3 Operating Procedure

Procedures for setting when starting and stopping operation of PWM-Diag are illustrated below.

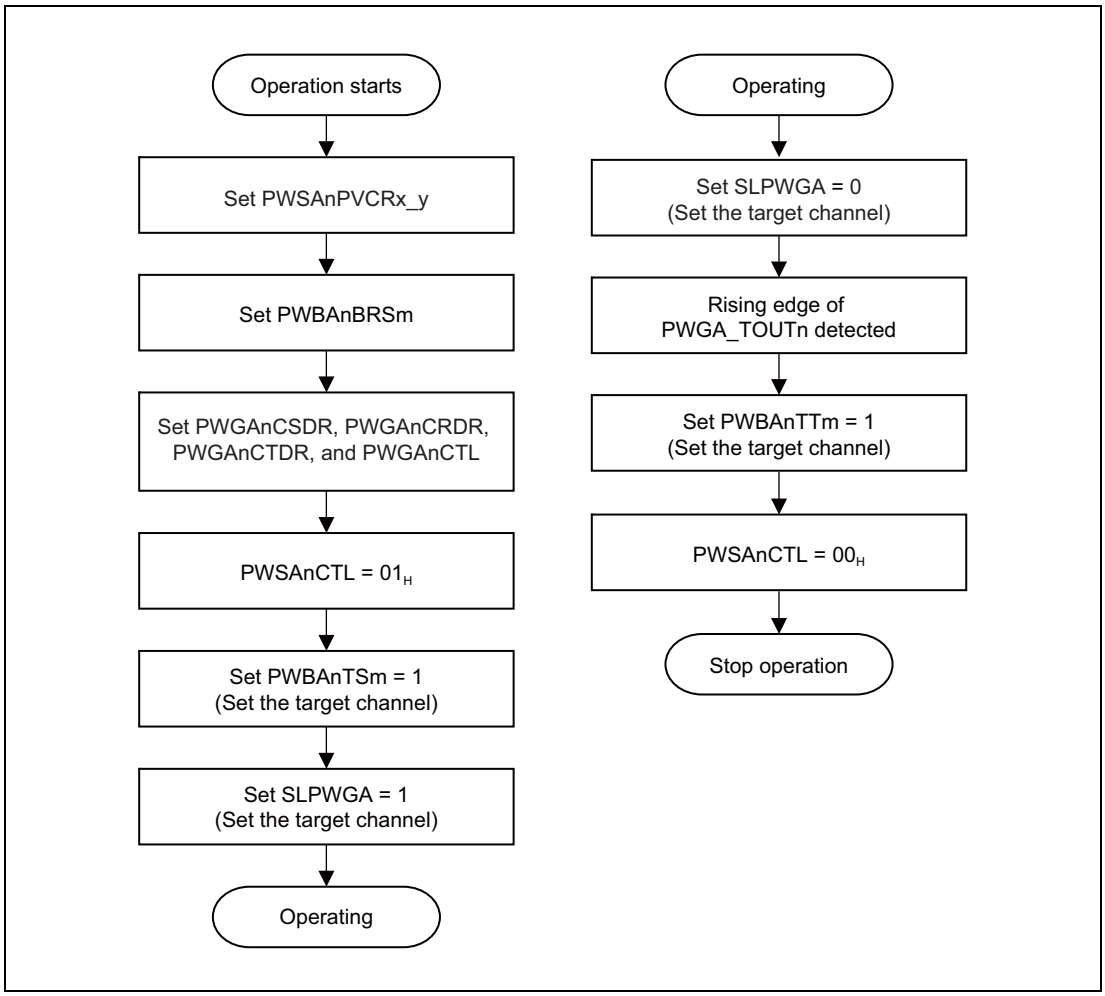
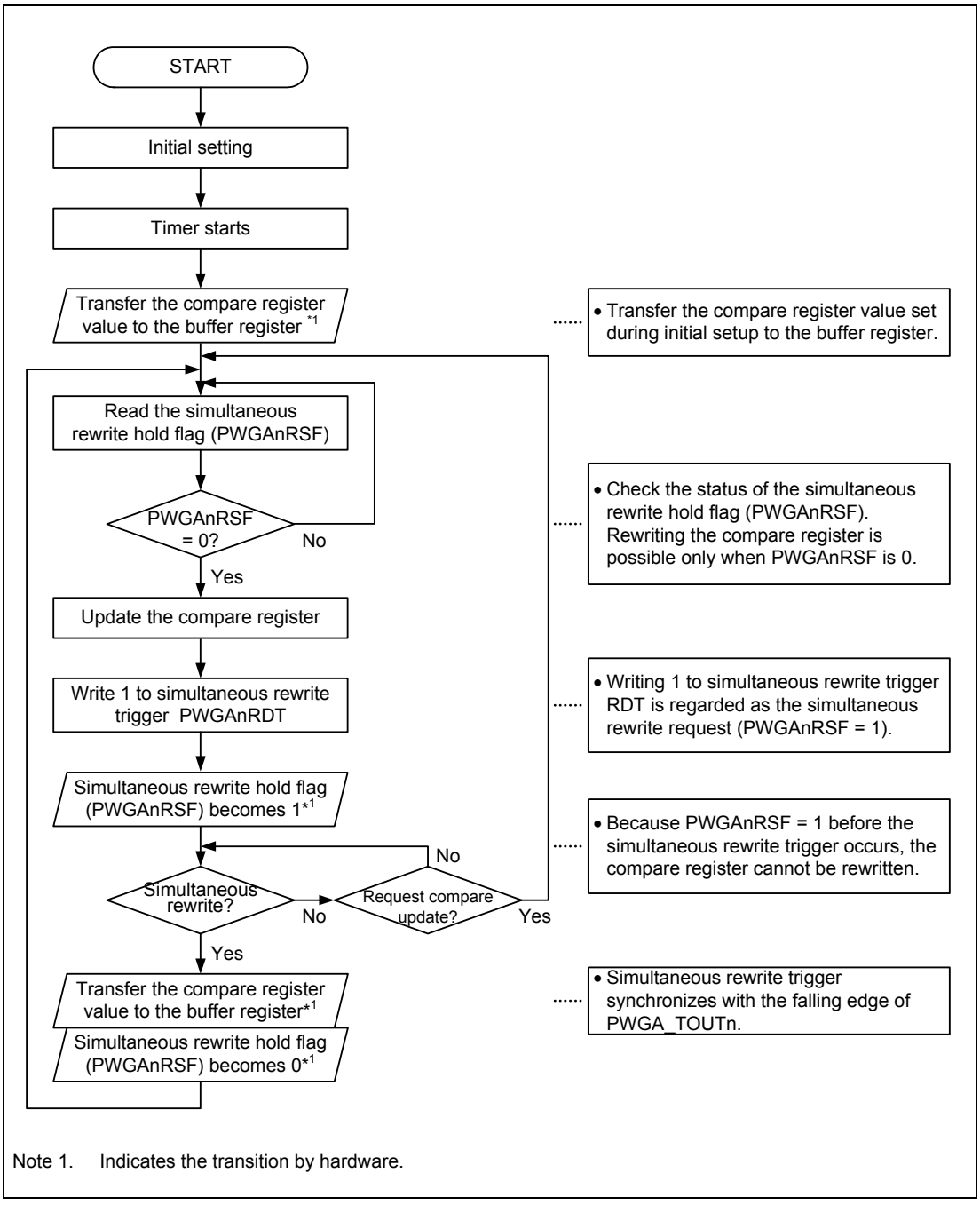


Figure 32.2 PWM-Diag Operating Procedure

Procedures for simultaneous rewrite of PWGA are illustrated below.

The described term “compare register” indicates PWGAnCSDR, PWGAnCRDR, or PWGAnCTDR.

In addition, the described term “buffer register” indicates PWGAnCSBR, PWGAnCRBR, or PWGAnCTBR.



Note 1. Indicates the transition by hardware.

Figure 32.3 Simultaneous Rewrite Procedure

32.4 Operation Waveform of PWM-Diag

32.4.1 PWM Waveform Output by PWGA and Operation Waveform for A/D Conversion Trigger Output

32.4.1.1 Basic Operation Waveform of PWGA

The basic operation waveforms of PWGA are illustrated below.

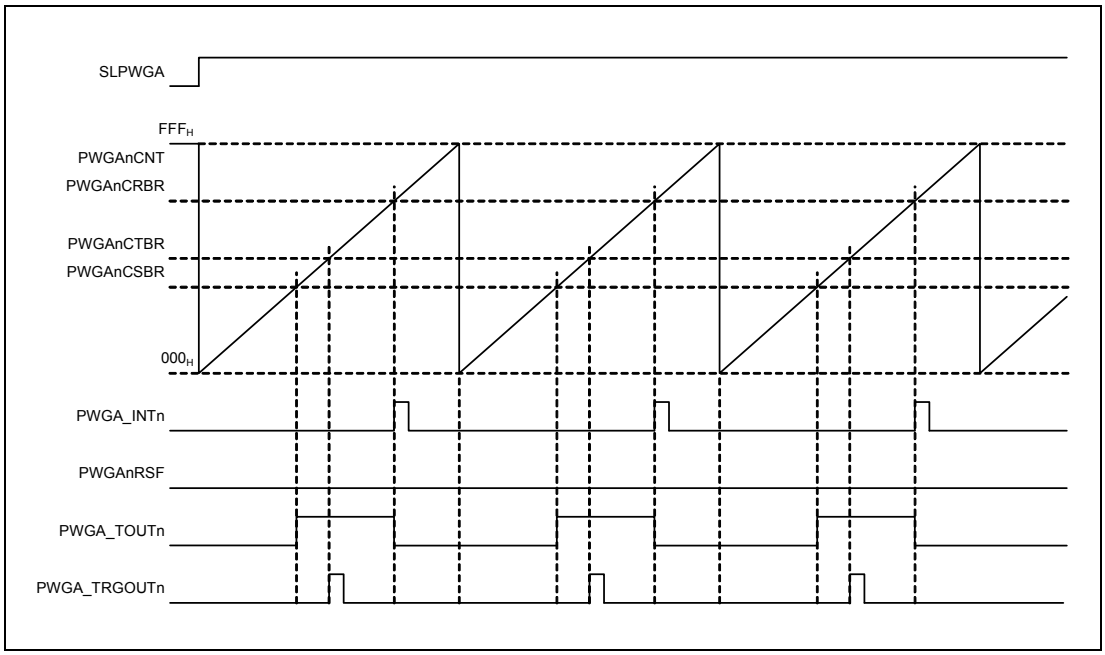


Figure 32.4 Basic Waveform

32.4.1.2 Operation Waveform when Simultaneous Rewrite for PWGA is Executed

The following figure illustrates the operation waveforms when simultaneous rewrite for PWGA is executed.

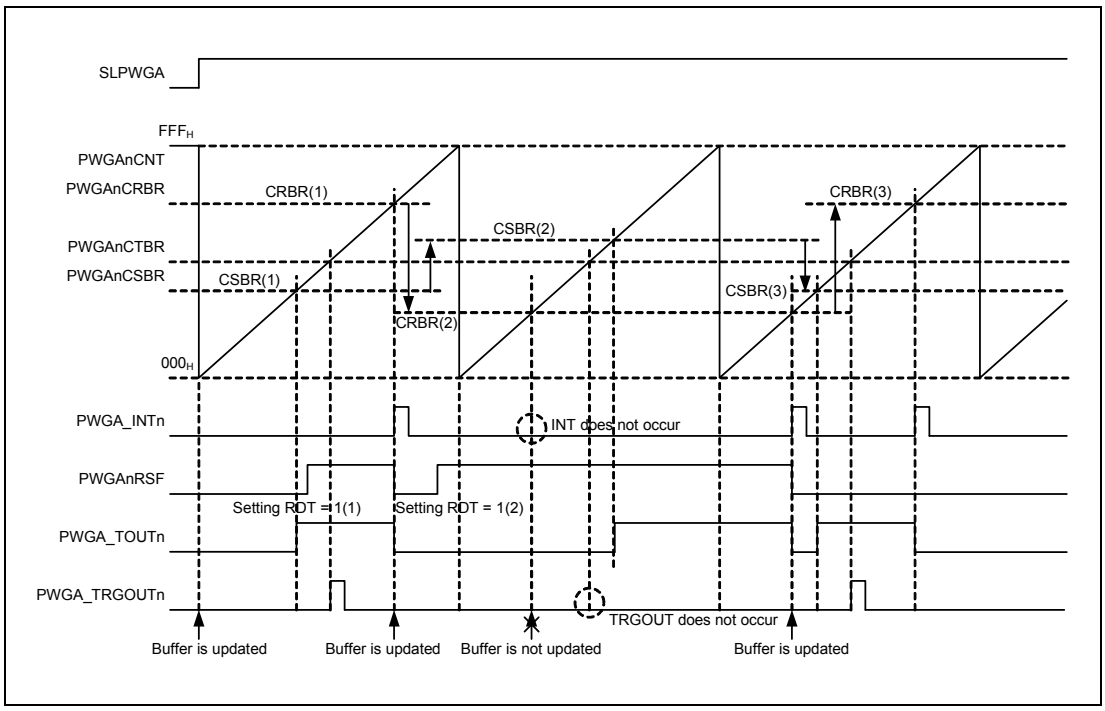


Figure 32.5 Waveform when Simultaneous Rewrite is Executed

Simultaneous rewrite is executed by re-setting the PWGAnCSDR and PWGAnCRDR registers, then setting either the PWGAnRDT or SLPWGAk register.

Moreover, if the relationship between set values in one interval is PWGAnCSDR > PWGAnCRDR, a falling edge in that interval is meaningless, and the falling edge in the next interval is valid.

Moreover, PWGA_TRGOUTn does not become valid unless PWGA_TOUTn is at the high level.

32.4.1.3 Operation Waveform when Stopping and Restarting PWGA Operation

The following figure illustrates the operation waveforms when stopping and restarting PWGA operation.

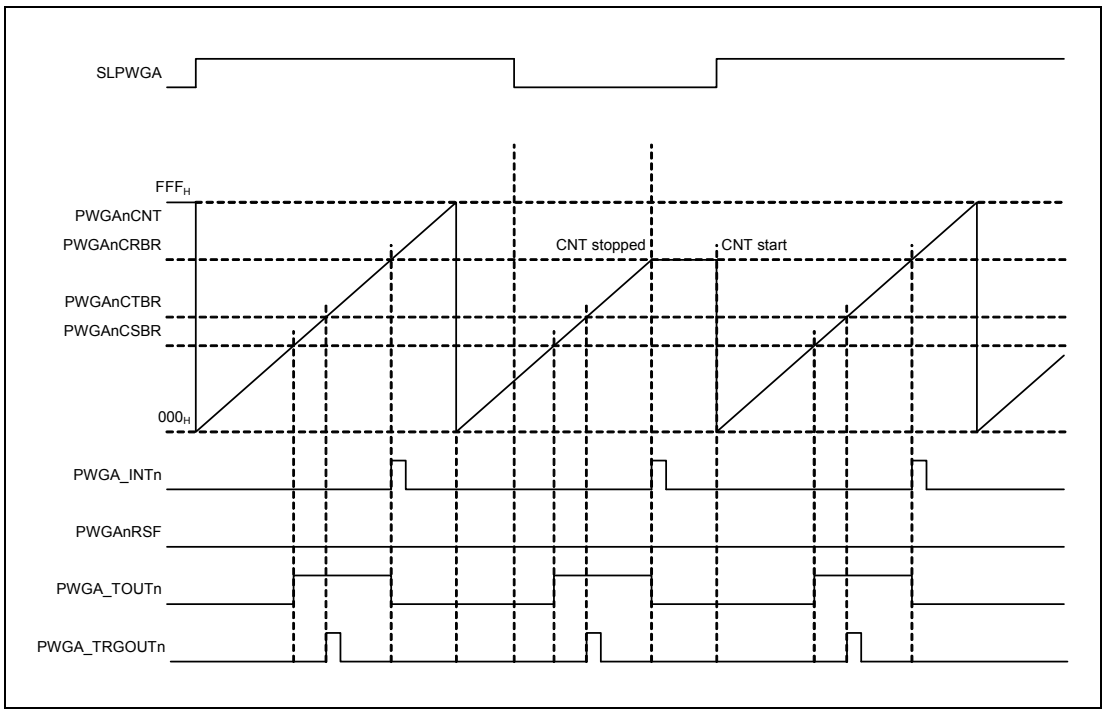


Figure 32.6 Stopping and Resuming Operation (1)

After the setting of SLPWGA has been changed from 1 to 0, PWGAnCNT stops operation because PWGA_INTn is generated.

After PWGA_INTn has been generated, by changing the setting of SLPWGA from 0 to 1, PWGAnCNT resumes counting from 000_H.

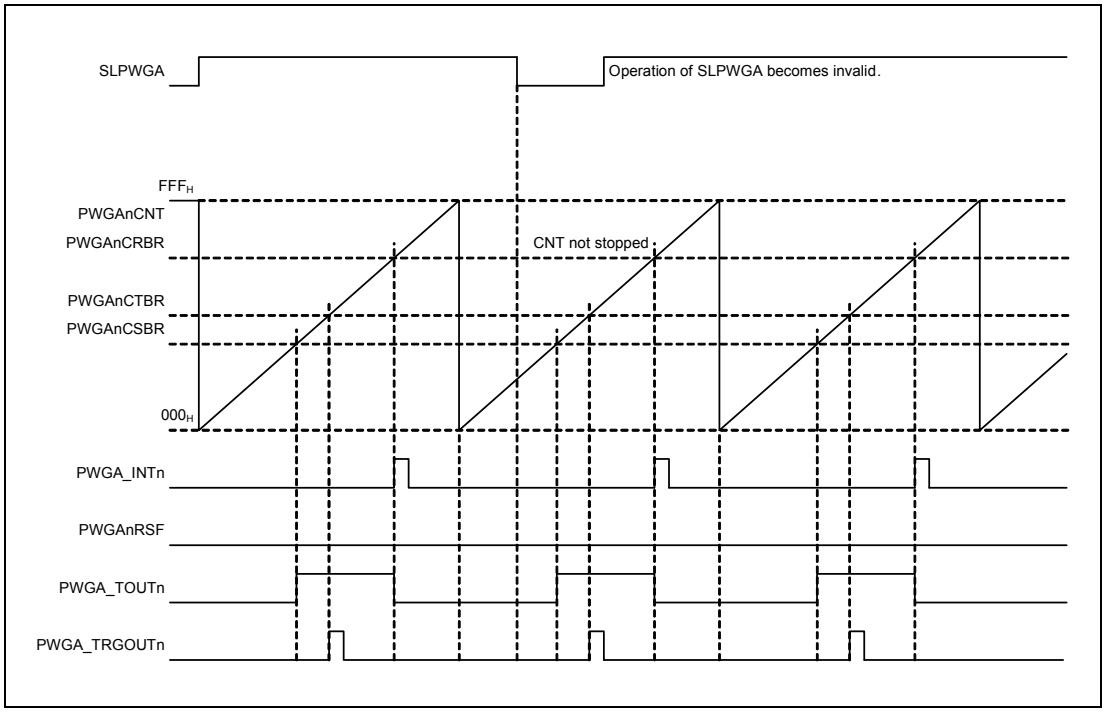


Figure 32.7 Stopping and Resuming Operation (2)

After the setting of SLPWGA has been changed from 1 to 0, if the setting of SLPWGA is changed from 0 to 1 before PWGA_INTn is generated, operations of SLPWGA become invalid, and PWGAnCNT continues counting.

32.4.1.4 Waveforms of PWGA Operation with Specific Settings

The following figures illustrate the waveforms of PWGA operation with specific settings.

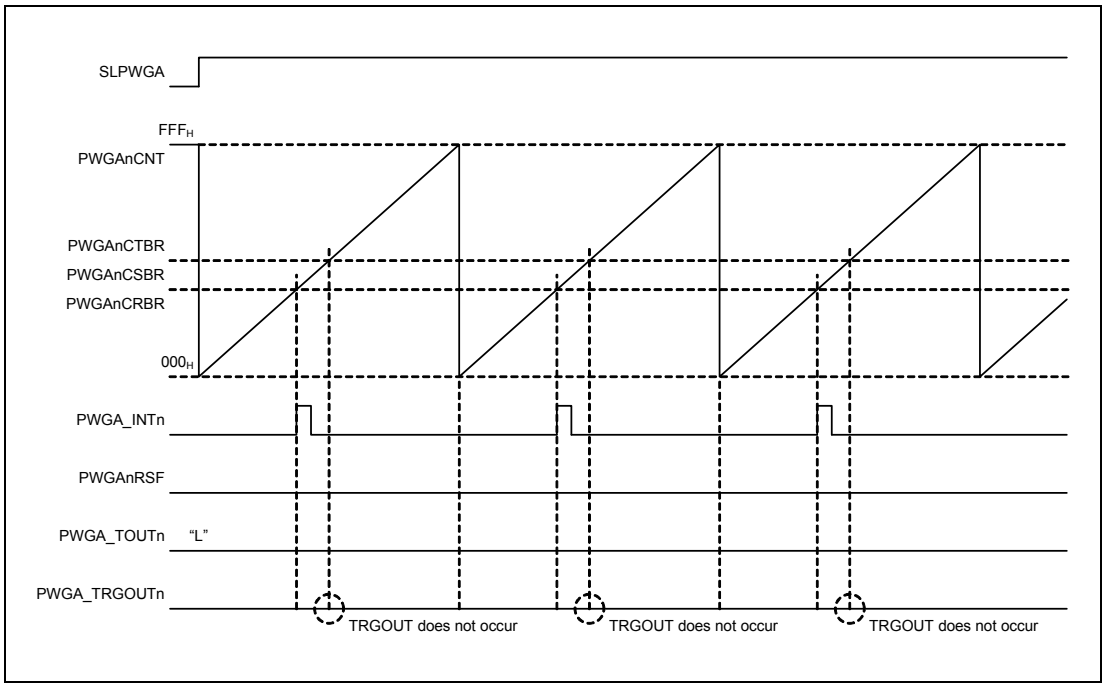


Figure 32.8 PWGA_TOUTn = 0% Output Waveform

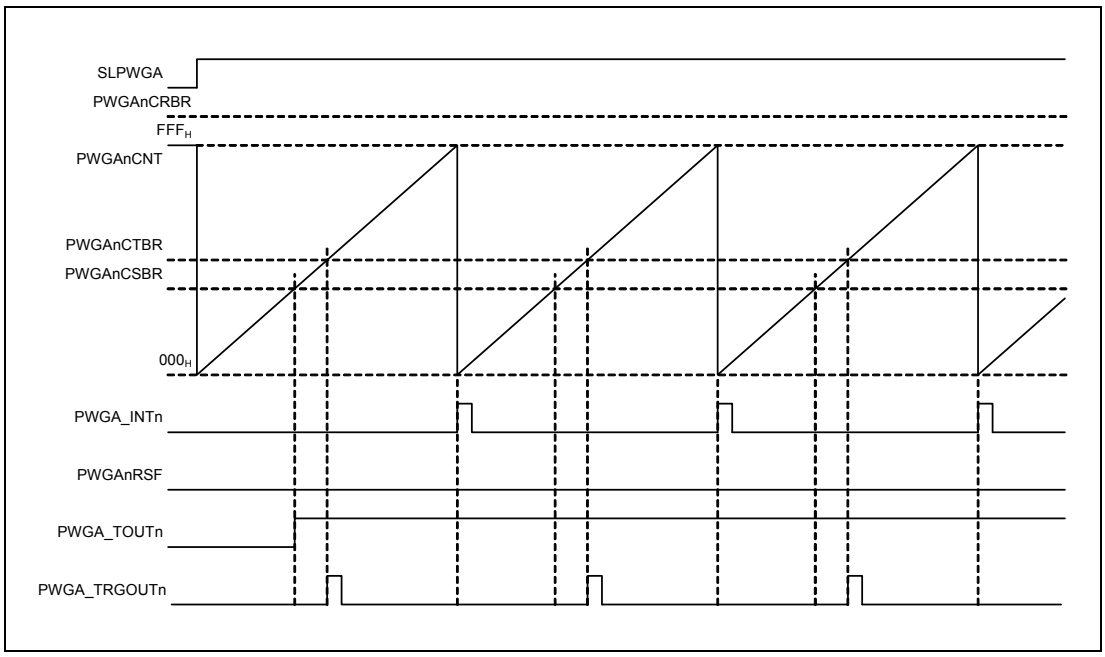
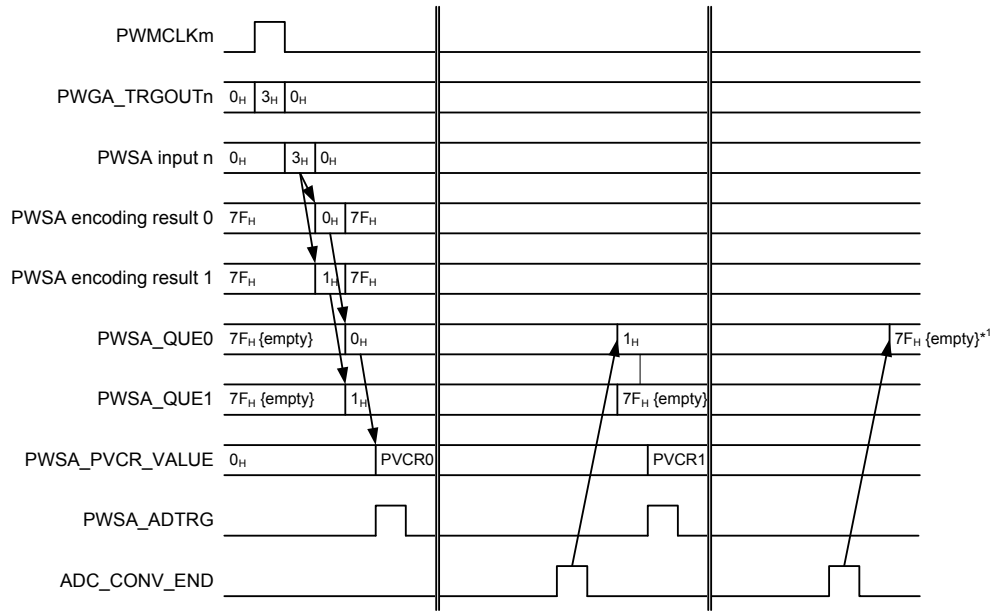


Figure 32.9 PWGA_TOUTn = 100% Output Waveform

32.4.2 Operation Waveform when A/D Conversion Trigger Occurs in PWSA

An example of the PWSA operation is shown below.



Note 1. QUE0[5:0] indicates the PWGA_TRG number while the A/D conversion is executed. The initial value (while the A/D conversion is not executed) is $7F_H$.

Figure 32.10 Example of PWSA Operation

(1) Triggers occur simultaneously in channels 0 and 1 of PWGA. Channel 0 with the smaller channel number is stored in PWSAnQUE0, and channel 1 with the larger channel number is stored in PWSAnQUE1. The lower 16 bits information of PWSAnPVCR00_01 corresponding the value stored in PWSAnQUE0 is transmitted to the A/D converter and a trigger is output to the A/D converter.
At this time, as the A/D conversion for channel 1 is in the waiting state, the PWSAnSTR.PWSAnQNE bit is set.

(2) On completion of A/D conversion executed in step (1), the channel number of PWSAnQUE1 shifts to PWSAnQUE0 and PWSAnQUE1 enters the empty state.
After that, as similar to step (1), the upper 16-bit information of PWSAnPVCR00_01 corresponding to the value stored in PWSAnQUE0 is transmitted to the A/D converter and a trigger is output to the A/D converter.

(3) On completion of A/D conversion executed in step (2), PWSAnQUE0 enters the empty state.

32.5 PWM-Diag Related Function in A/D Converter (ADCE)

This section describes the A/D converter used in the PWM-Diag function.

32.5.1 ADCE registers when the PWM-Diag function is used

- Before starting PWSA operation, the A/D converter must be set using the following register.
  - PWM-Diag scan group control register (ADCEnPWDSGCR)
- When the PWM-Diag is running, the PWSAnPVCRx_y setting value corresponding to the channel under conversion is set in the following registers of the A/D converter.
  - PWM-Diag virtual channel register (ADCEnPWDVCR)
- After completion of A/D conversion, the conversion result can be checked by reading the following registers.
  - PWM-Diag data register (ADCEnPWDTSNDR)
  - PWM-Diag data supplementary information register (ADCEnPWDDIR)
- When A/D conversion result is outside the expected range, it can be confirmed using the upper/lower limit error detection function. The upper/lower limit error detection function is set by the following register.
  - Upper limit/lower limit error register (ADCEnULER)
- The scan end flag of the PWM-Diag scan group can be cleared using the following register.
  - PWM-Diag scan end flag clear register (ADCEnPWDSGSEFCR)

Section 33 Sound Generator (SG)

This section contains a generic description of the Sound Generator.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

33.1 Overview of the RH850/D1L/D1M Sound Generators

33.1.1 Units

This microcontroller has the following number of units of the Sound Generator.

Table 33.1 Units

Sound Generator	
Units	5
Names	SG0 to SG4

Unit index n

Throughout this section, the individual units of the Sound Generators are identified by the index “n” (n = 0 to 4), for example SGnCTL for the SGn control register.

33.1.2 Register addresses

All Sound Generator register addresses are given as address offsets from the individual base addresses <SGn_base>.
The <SGn_base> addresses of each SGn are listed in the following table:

Table 33.2 Register base addresses <SGn_base>

SGn unit	<SGn_base> address
SG0	FFF1 1000_H
SG1	FFF1 1100_H
SG2	FFF1 1200_H
SG3	FFF1 1300_H
SG4	FFF1 1400_H

33.1.3 Clock supply

All Sound Generators provide one clock input.

Table 33.3 Clock supply

SGn unit	SGn clock	Connected to
SG0	PCLK	Clock Controller CLKFIX
SG1	PCLK	Clock Controller CLKFIX
SG2	PCLK	Clock Controller CLKFIX
SG3	PCLK	Clock Controller CLKFIX
SG4	PCLK	Clock Controller CLKFIX

33.1.4 Interrupts and DMA

The Sound Generators can generate the following interrupt and DMA requests:

Table 33.4 SGn interrupt and DMA requests

SGn signals	Function	Connected to
INTSG0TI	Threshold interrupt	Interrupt Controller INTSG0TI DMA Controller trigger ID 118
INTSG1TI	Threshold interrupt	Interrupt Controller INTSG1TI DMA Controller trigger ID 119
INTSG2TI	Threshold interrupt	Interrupt Controller INTSG2TI DMA Controller trigger ID 120
INTSG3TI	Threshold interrupt	Interrupt Controller INTSG3TI DMA Controller trigger ID 121
INTSG4TI	Threshold interrupt	Interrupt Controller INTSG4TI DMA Controller trigger ID 122

33.1.5 Reset sources

The Sound Generators and their registers are initialized by the following reset signal:

Table 33.5 Reset sources

SGn unit	Reset signal
SG0	• Reset Controller SYSRES • Reset Controller SG0RES • reset upon wake-up from DEEPSTOP mode
SG1	• Reset Controller SYSRES • Reset Controller SG1RES • reset upon wake-up from DEEPSTOP mode
SG2	• Reset Controller SYSRES • Reset Controller SG2RES • reset upon wake-up from DEEPSTOP mode
SG3	• Reset Controller SYSRES • Reset Controller SG3RES • reset upon wake-up from DEEPSTOP mode
SG4	• Reset Controller SYSRES • Reset Controller SG4RES • reset upon wake-up from DEEPSTOP mode

CAUTION

By default the SGnRES resets are active.
Thus before accessing an SGn module this reset must be released via the MRSTC register of the Reset Controller. Refer to Section 9, Reset Controller for details.

33.1.6 I/O signals

The following table shows the I/O signals of the Sound Generators.

Table 33.6 I/O signals connections

SGn signal	Function	Connected to
SG0:		
SGTFAO	Tone frequency or composite output	Port SG0FAO
SGTFAOL	Inverted composite output	Port SG0FAOL
SGTAO	PWM volume output	Port SG0AO
SG1:		
SGTFAO	Tone frequency or composite output	Port SG1FAO
SGTFAOL	Inverted composite output	Port SG1FAOL
SGTAO	PWM volume output	Port SG1AO
SG2:		
SGTFAO	Tone frequency or composite output	Port SG2FAO
SGTFAOL	Inverted composite output	Port SG2FAOL
SGTAO	PWM volume output	Port SG2AO
SG3:		
SGTFAO	Tone frequency or composite output	Port SG3FAO
SGTFAOL	Inverted composite output	Port SG3FAOL
SGTAO	PWM volume output	Port SG3AO
SG4:		
SGTFAO	Tone frequency or composite output	Port SG4FAO
SGTFAOL	Inverted composite output	Port SG4FAOL
SGTAO	PWM volume output	Port SG4AO

XOR Compare Unit check of SG output signals

The following SGn output signals can be checked by the XOR Compare Unit:

- SG0: SG0FAO, SG0FAOL at port P3_7, SG0AO at port P3_7.

33.2 Functional Overview

The Sound Generator (SG) consists of a programmable square wave tone generator and a programmable pulse-width modulator (PWM).

It generates an audio-frequency tone signal and a high-frequency pulse-width modulated signal. The duty cycle of the PWM signal defines the volume.

By default, the two signal components are routed to separate outputs. But both signals can also be combined to generate a composite signal that can be used to drive a loudspeaker circuit.

The PWM includes an internal automatic logarithmic decrement unit (ALD). The ALD can be used to reduce the tone volume over time without CPU intervention.

Features summary

Special features of the Sound Generator are:

- Programmable tone frequency (100 Hz to 6 kHz with a minimum step size of 20 Hz)
- Programmable volume level (9 bit resolution)
- Automatic logarithmic volume decrement function (ALD):
  - Volume reduction without CPU interaction
  - Programmable sound duration (256 steps)
  - Sound duration associated with tone frequency (gong effect)
  - Interrupt generation when programmable volume level is reached
- Automatic duration control (ADC)
- Automatic decrement/increment of volume (ADI)
- Wide range of PWM base frequencies (32 kHz to 64 kHz)
- Sound can be stopped or re-triggered (even if the ALD/ADC function is switched on)
- Composite or separated frequency/volume output for external circuitry variation
- Hardware-optimized update of frequency and volume to avoid audible artefacts

33.2.1 Description

The following figure provides a functional block diagram of the Sound Generator.

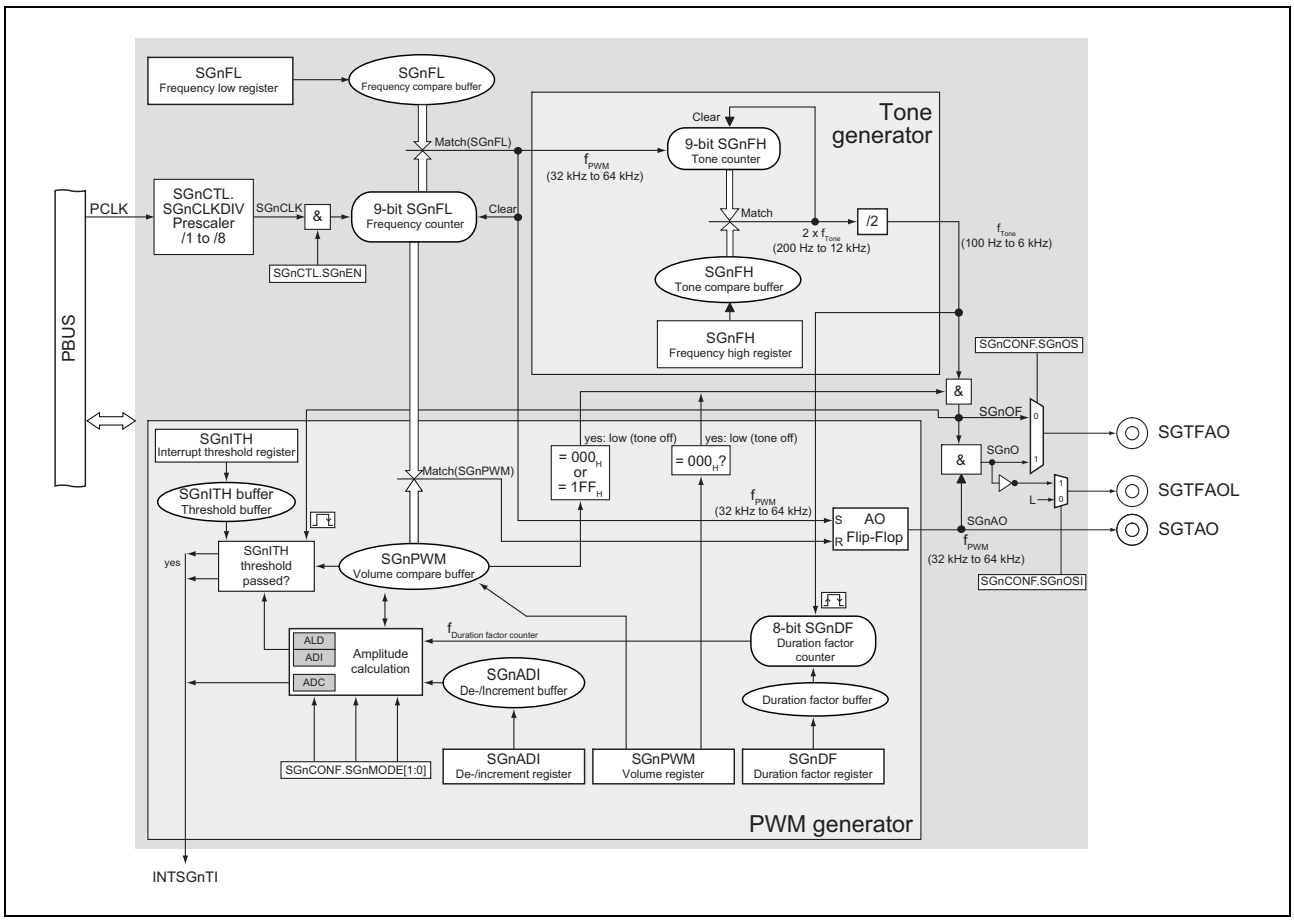


Figure 33.1 Sound Generator block diagram

PCLK

The Sound Generator's clock input PCLK is specified in the first section of this chapter under the key word “Clock supply”.

Tone generator

The tone generator consists of two up-counters with compare registers. The values written to the frequency registers are automatically copied to compare buffers. The counters are reset to zero when their values match the contents of the associated compare buffers.

The 9-bit frequency counter SGnFL generates a clock with a frequency between 32 kHz and 64 kHz. This clock constitutes the PWM base frequency f_{PWM}.

It is also the input of the second 9-bit tone counter SGnFH and a divider by two. The resulting tone frequency signal SGnOF has a frequency between 100 Hz and 6 kHz and a 50 % duty cycle.

PWM generator

The PWM generator generates the PWM volume signal SGnAO with the PWM base frequency f_{PWM}. The composite signal SGnO consists of the tone frequency signal SGnOF that is modulated with the PWM volume signal SGnAO (logical AND combination).
Thus the duty cycle of SGnAO determines the volume of the final audio signal.

The duty cycle of SGnAO is controlled by the volume register SGnPWM. The value written to this register is automatically copied to the associated volume compare buffer.

The PWM continually compares the value of the counter SGnFL with the contents of its volume compare buffer.

The AO Flip-Flop of the PWM generator is set by the pulses generated by the counter SGnFL (PWM base frequency f_{PWM}). It is reset when the SGnFL counter value matches the contents of the volume compare buffer SGnPWM. Thus, the PWM volume signal can have a duty cycle between 0 % (null volume) and 100 % (maximum volume).

The PWM base frequency is above 32 kHz and hence outside the audible range.

Outputs

The Sound Generator outputs following signals:

- SGTAO outputs the PWM volume signal SGnAO that holds the volume information ("amplitude").
- SGTFAO outputs by default the tone frequency signal SGnOF.
  If the SGnCONF.SGnOS bit is set, SGTFAO outputs the composite signal SGnO that can directly control a speaker circuit.
- SGTFAOL outputs by default a low level.
  If the SGnCONF.SGnOSI bit is set, SGTFAOL outputs the inverted composite signal SGnO.

33.2.2 Principle of operation

The software-controlled registers SGnFL, SGnFH, and SGnPWM are equipped with hardware buffers. The Sound Generator operates on these buffers.

This approach eliminates audible artefacts, because the buffers are only updated in synchronization with the generated tone frequency signal.

NOTE

This section provides an overview. For details please refer to Section 33.3, Functional Description.

33.2.2.1 Generation of the tone frequency

The tone frequency is determined by the tone frequency signal SGnOF, a rectangular signal with 50 % duty cycle.

The tone frequency f_{TONE} is generated by two counters and their associated compare register values. Two counters are necessary to keep the pulses of the tone frequency signal and the PWM volume signal synchronized.

The first counter (SGnFL) provides the input to the second (SGnFH) and also to the PWM. It is used to keep the working frequency of connected modules (PWM and tone generation) within the limits of 32 kHz and 64 kHz. Its match value defines also the 100 % volume level.

The second counter (SGnFH) generates the tone frequency (100 Hz to 6 kHz).

NOTE

If the SGnFL counter value is changed to generate a different tone frequency, the volume register SGnPWM has to be adjusted accordingly to keep the same volume.

33.2.2.2 Generation of the volume information (amplitude)

The volume information (the “amplitude” of the audible signal) is provided as the high-frequency PWM volume signal SGnAO. The duty cycle of SGnAO determines the amplitude of the final audio signal.

The composite signal SGnO is a logical AND combination of the PWM volume signal SGnAO and the tone frequency signal SGnOF.

After low-pass filtering, the analog signal amplitude corresponds to the duty cycle of the PWM volume signal. Low-pass filtering (averaging) is an inherent characteristic of a loudspeaker system.

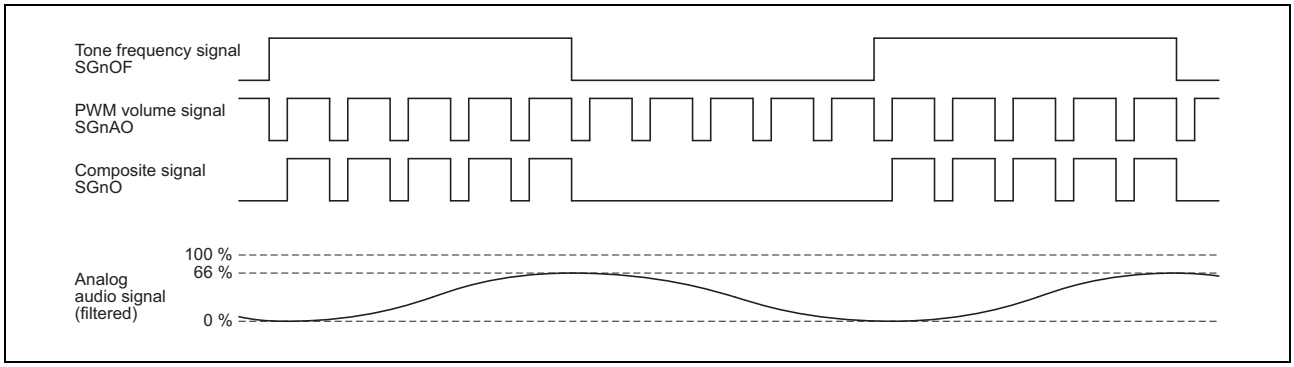


Figure 33.2 Generation of the composite output signal

The duty cycle can vary between 0 % and 100 %. Its generation is controlled by the frequency low register SGnFL and the volume register SGnPWM.

For details on SGnPWM updates, refer to Section 33.3.3, Updating the buffer values.

33.2.2.3 Automatic logarithmic fading (ALD)

The automatic logarithmic decrement function (ALD) provides an automatic volume reduction without CPU interaction.

In regular intervals (related to the tone frequency, selectable via the SGnDF register), the present contents of the volume compare buffer is divided by an automatically calculated division factor. The result is subtracted from the buffer value. The logarithmic reduction creates the impression of a linear volume reduction in the human ear.

The initial volume that is defined by the content of the volume register SGnPWM remains unchanged.

For a detailed description refer to Section 33.3.4, Automatic logarithmic fading (ALD).

33.2.2.4 Automatic decrement/increment of volume (ADI)

The automatic decrement/increment function (ADI) provides fading in and out of the volume without CPU interaction.

Depending on the value of the duration factor SGnDF, the volume is increased or decreased.

The sound is stopped when the volume compare buffer is 000_H or $1FF_H$.

For a detailed description refer to Section 33.3.5, Automatic decrement/increment (ADI) of the volume.

33.2.2.5 Automatic duration control (ADC)

The automatic duration control (ADC) stops the tone after a configurable number of periods of the tone frequency.

For a detailed description refer to Section 33.3.6, Automatic duration control (ADC).

33.2.2.6 Interrupt generation

In ALD, ADI and ADC mode the interrupt INTSGnTI can be generated:

- In ALD mode, the interrupt signals that the tone volume has decreased to a certain level (set in register SGnITH). Because the sound duration depends on the tone frequency and the contents of the sound duration register SGnDF, INTSGnTI can be used to indicate “sound is low” or “sound has ended”. Refer to Section 33.3.4.3, Interrupt in ALD mode for details.
- In ADI mode, the interrupt signals that the tone volume has decreased or increased to a certain level (set in register SGnITH). Refer to Section 33.3.5.3, Interrupt in ADI mode for details.
- In ADC mode, the interrupt signals that the sound has stopped. Refer to **Section 33.3.6.2, Interrupt in ADC mode** for details.

The interrupt is generated only once after the volume level has passed the value written to the SGnITH register.

33.3 Functional Description

This section explains the details of the Sound Generator.

33.3.1 Generating the tone

The tone frequency signal SGnOF is generated by the compare match signal of the tone counter SGnFH value with the value of the tone compare buffer SGnFH, followed by a by-two-divider. At each compare match, the counter is reset to zero.

The tone counter SGnFH is clocked by the output of the frequency counter SGnFL.

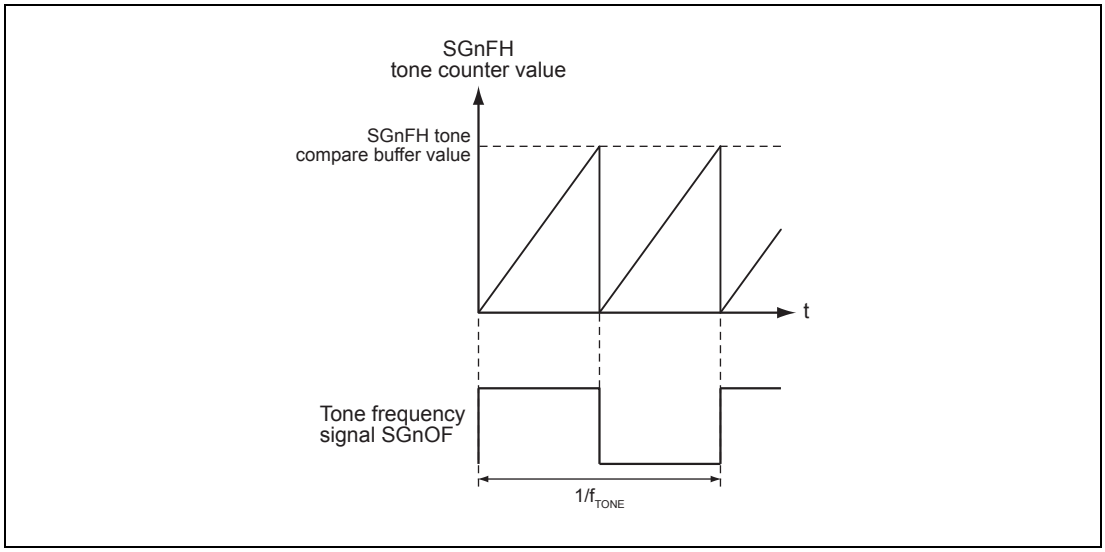


Figure 33.3 Generation of tone frequency

33.3.1.1 Updating the frequency and tone buffer values

The values of the frequency compare buffer and the tone compare buffer can be changed by writing to the associated frequency registers SGnFL and SGnFH.

Changing the value of the SGnFL register would also yield a change of the PWM base frequency, i.e. the tone volume. Therefore it is obligatory to write the correct PWM value to the SGnPWM register after a new SGnFL value is copied to the frequency compare buffer.

For a detailed description, refer to Section 33.3.3, Updating the buffer values.

33.3.1.2 Tone frequency calculation

The tone frequency can be calculated as:

$$f_{TONE} = \frac{f_{SGnCLK}}{2 \bullet (SGnFL + 1) \bullet (SGnFH + 1)}$$

with

$$_{SGnCLK} = \frac{f_{PCLK}}{SGnCLKDIV + 1}$$

where:

f_{PCLK}:	frequency of the Sound Generator input clock PCLK
f_{SGnCLK}:	frequency of the prescaler output clock SGnCLK
SGnFL:	contents of the SGnFL buffer
SGnFH:	contents of the SGnFH buffer
SGnCLKDIV:	register bits SGnCTL.SGnCLKDIV[2:0]

Example

With

- f_{SGnCLK} = 16 MHz
- SGnFL buffer = 255 ($00FF_H$) (this yields a PWM base frequency of 62.5 kHz)
- SGnFH buffer = 32 (0020_H)

a tone frequency f_{TONE} = 947 Hz is generated.

NOTE

The buffer contents can differ from the contents of the associated register until the next compare match.

33.3.2 Generating the volume information

The tone volume information, i.e. the frequency and duty cycle of the PWM volume signal SGnAO, is generated by comparing the SGnFL frequency counter value with the contents of the SGnFL frequency compare buffer and the SGnPWM volume compare buffer.

A Reset-Set Flip-Flop (AO-FF) is

- set if the SGnFL frequency counter value matches the SGnFL frequency compare buffer value (PWM base frequency f_{PWM})
- reset if the SGnFL frequency counter value matches the SGnPWM volume compare buffer value (duty cycle).

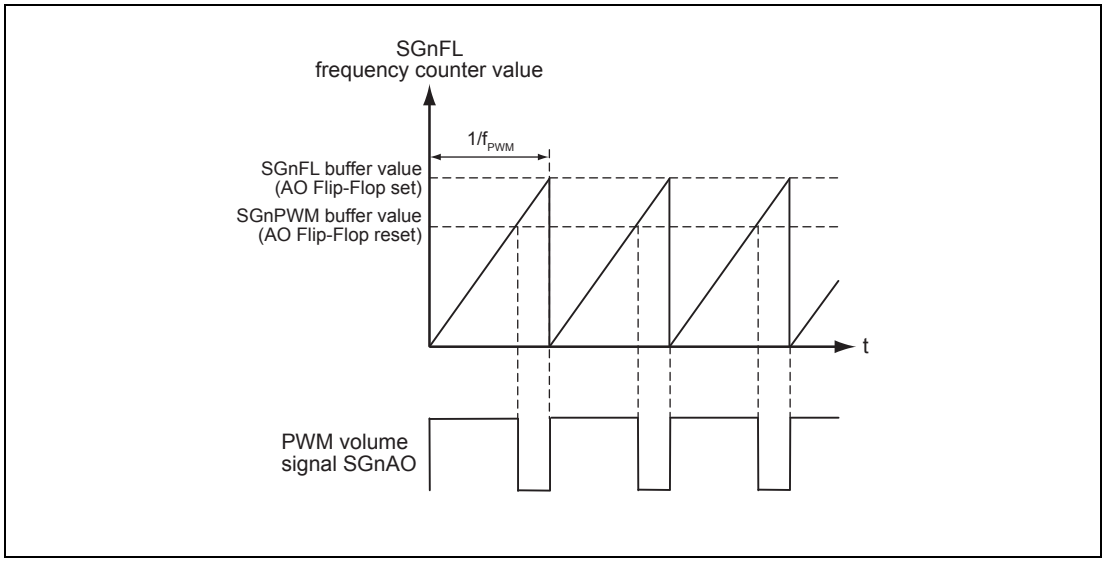


Figure 33.4 PWM signal generation

The duty cycle of the PWM volume signal is determined by the difference between the contents of the SGnFL counter buffer and the content of the SGnPWM volume buffer. A smaller difference (SGnFL - SGnPWM) yields a smaller duty cycle.

NOTES

1. The PWM volume signal SGnAO is continually high (100 % duty cycle) when the value of the SGnPWM volume buffer is higher than the value of the SGnFL frequency compare buffer.
2. The PWM volume signal SGnAO is continually low (0 % duty cycle) when the value of the SGnPWM volume buffer is zero, i.e. tone output has stopped.
3. The frequency low register SGnFL must not be set to a value above $1FE_H$.

33.3.2.1 PWM calculations

PWM base frequency

The PWM base frequency is determined by the frequency compare buffer SGnFL. It can be calculated as:

$$f_{PWM} = \frac{f_{SGnCLK}}{SGnFL + 1}$$

where:

f_{SGnCLK}: frequency of the prescaler output clock SGnCLK

SGnFL: contents of the SGnFL buffer

Duty cycle

The duty cycle of the PWM volume signal is calculated as follows:

- If [SGnPWM buffer] > [SGnFL buffer]:
  Duty cycle = 100 %
- If 0 ≤ [SGnPWM buffer] ≤ [SGnFL buffer]:
  Duty cycle = [SGnPWM buffer] / ([SGnFL buffer] + 1)

where:

[SGnPWM buffer] = contents of SGnPWM buffer

[SGnFL buffer] = contents of SGnFL buffer

Example

If register SGnFL is set to 240 ($00F0_H$), the following table applies:

Table 33.7 Duty cycle calculation example

[SGnPWM]	Calculation	Duty cycle [%]
$01FF_H$		100
...		100
$00F1_H$	241 / 241	100
$00F0_H$	240 / 241	99.6
$00EF_H$	239 / 241	99.2
...	...	...
0001_H	1 / 241	0.41
0000_H	0 / 241	0

The table shows, how the contents of register SGnFL affects the achievable volume resolution.

33.3.3 Updating the buffer values

The Sound Generator operates internally not directly with registers, which can be written by the application program, but with buffers of these registers.
The content of the registers are copied to their buffers (i.e. the buffer update) under specific conditions. This way the register content can be changed by the application program at any time without interfering with an ongoing tone generation.

Following registers have associated buffer registers: SGnPWM, SGnCONF, SGnDF, SGnADI, SGnITH, SGnFL, SGnFH.

33.3.3.1 Start of a new tone

Two events can basically enable an update of all buffers, e.g. for starting generation of a new tone.

It can be selected by the SGnCONF.SGnBE bit which of both is used:

- SGnCONF.SGnBE = 0:
  - buffer update by write to the volume register SGnPWM enabled
  - buffer update by INTSGnTI disabled
- SGnCONF.SGnBE = 1:
  - buffer update by INTSGnTI enabled
  - buffer update by write to the volume register SGnPWM disabled

In both above cases the generation of a new tone is started. Therefore all buffers are updated to specify the parameters of the new tone.

(a) SGnCONF.SGnBE = 0: SGnPWM write synchronization

The figure below shows a buffer update process by writing to SGnPWM.

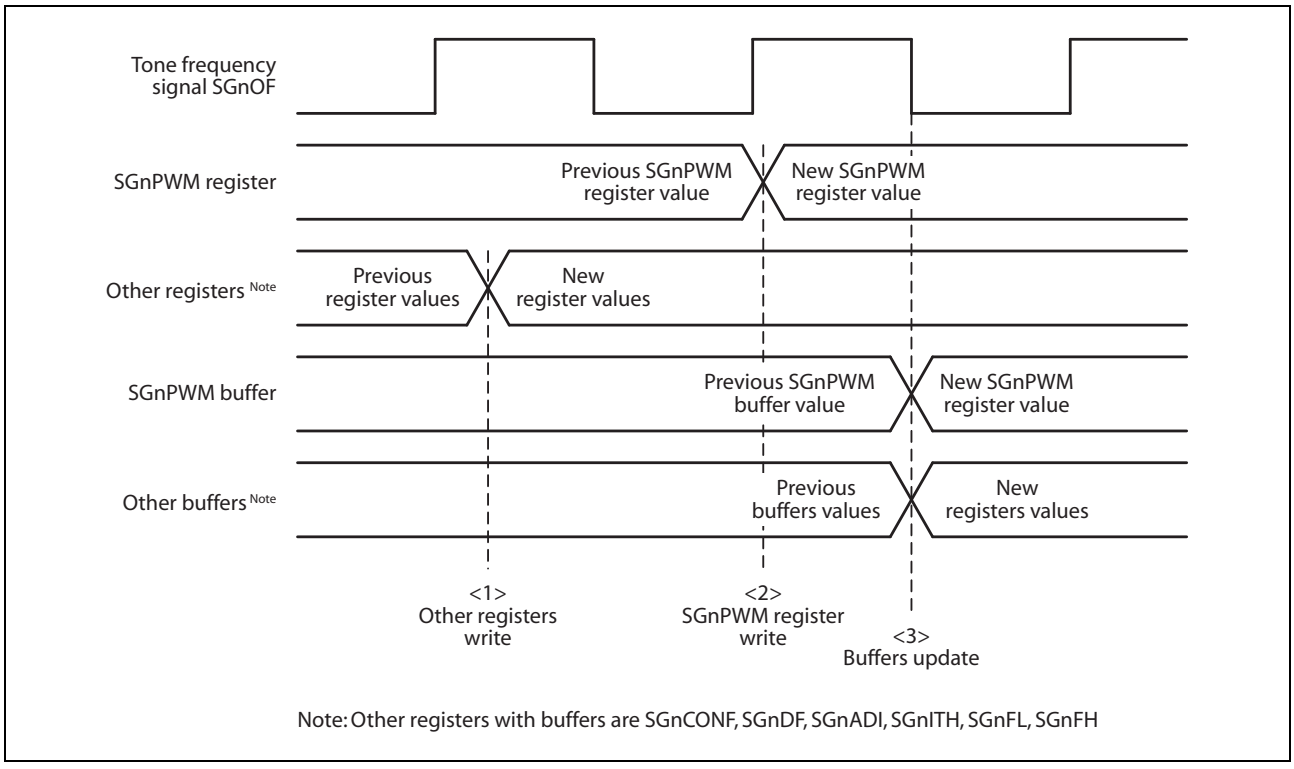


Figure 33.5 Update of buffer values by write to SGnPWM (SGnCONF.SGnBE = 0)

Before writing a new value to SGnPWM, all other buffered registers have to contain their desired values (<1>). Write to SGnPWM (<2>) enables the update of all buffers - including the SGnPWM buffer - with the falling next edge of the tone frequency signal SGnOF (<3>).

(b) SGnCONF.SGnBE = 1: INTSGnTI synchronization

The figure below shows a buffer update process by the INTSGnTI interrupt.
The interrupt is set up to be asserted at volume level D, i.e. the interrupt threshold register SGnITH is set to volume D.

The SGnPWM buffer is updated automatically by the ALD or ADI function each tone frequency signal falling edge.

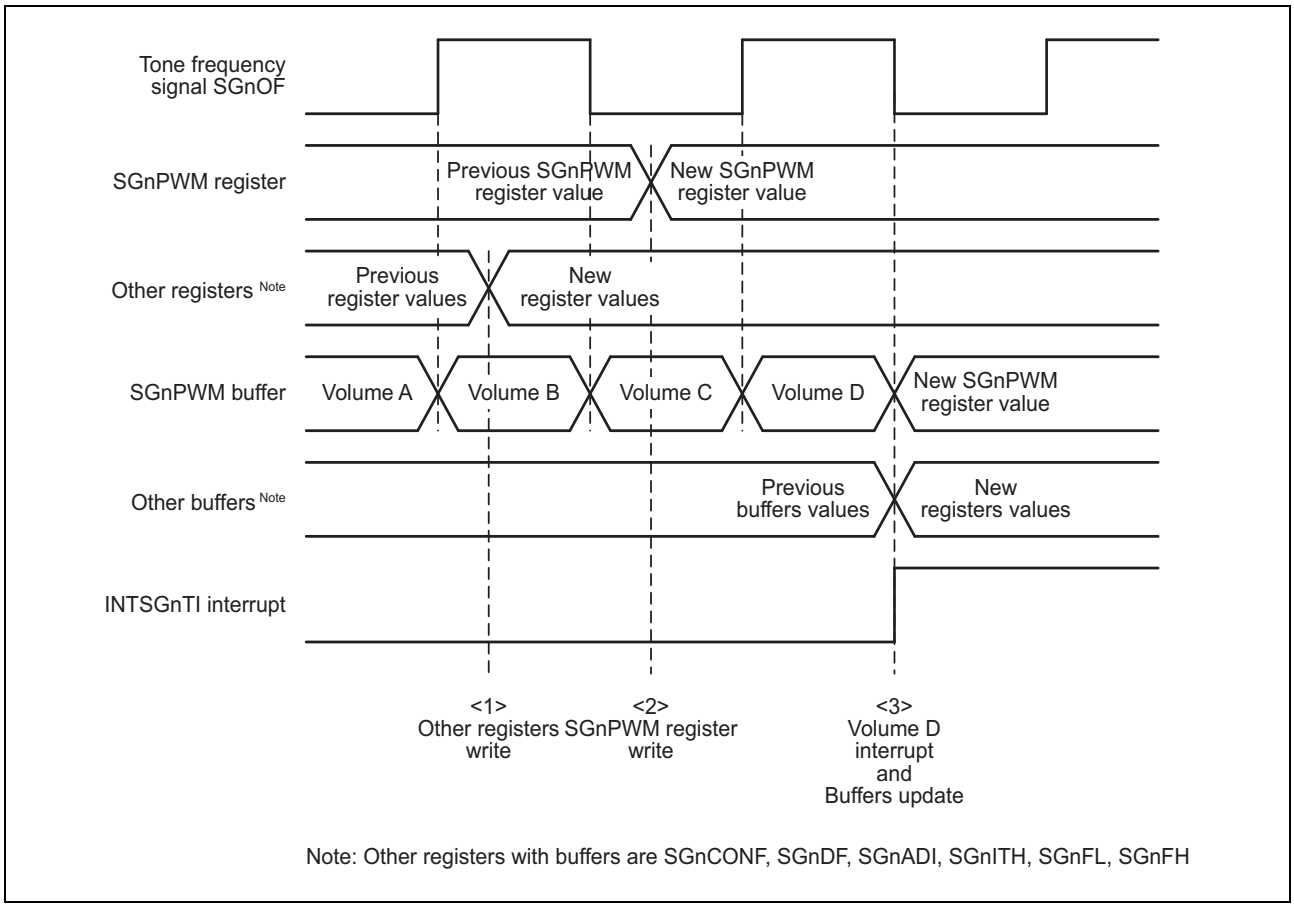


Figure 33.6 Update of buffer values by INTSGnTI (SGnCONF.SGnBE = 1)

Before the SGnPWM buffer reaches the volume D level all buffered registers have to contain their desired values (<1>) - including the SGnPWM buffer (<2>).
The INTSGnTI is asserted at volume D (<3>) and all buffers - including the SGnPWM buffer - are updated.

ADC mode

In automatic duration control mode (ADC) mode, the interrupt INTSGnTI is asserted, when the tone generation stops. However the buffer update is performed also in that case upon occurrence of INTSGnTI.

33.3.3.2 Volume compare buffer SGnPWM in ALD and ADI mode

In automatic volume adjustment modes (logarithmic volume fading (ALD) and volume decrement/increment (ADI)) the SGnPWM volume buffer is updated automatically during the tone generation process for changing the volume.

33.3.3.3 Buffer update conditions overview

The table below shows a summary of the buffers update conditions:

Table 33.8 Buffers update conditions

Buffers update conditions					
Register settings					
SGnCTL	SGnCONF		SGnPWM register write	Description	Updated Buffers
SGnEN	SGnBE	SGnMODE[1:0]			
1	0	X	yes	Tone generation is ongoing. Buffers update at next tone frequency signal (SGnOF) to continue with new tone.	SGnPWM, SGnCONF, SGnDF, SGnADI, SGnITH, SGnFL, SGnFH
	1	• 01_B (ADC) • 10_B (ADI) • 11_B (ALD)	X*[1]	Tone generation is ongoing. Buffers update at generation of INTSGnTI interrupt to continue with new tone.	SGnPWM, SGnCONF, SGnDF, SGnADI, SGnITH, SGnFL, SGnFH
	X	• 10_B (ADI) • 11_B (ALD)	no	Tone generation with automatic volume control is ongoing. Next calculated volume value stored in SGnPWM buffer at specified edge (SGnDF) of the tone frequency signal SGnOF.	SGnPWM
0	X	X	yes	Tone generation is stopped. Buffers are updated immediately.	SGnPWM, SGnCONF, SGnDF, SGnADI, SGnITH, SGnFL, SGnFH

Note 1. A write to the SGnPWM register does *not* enable the buffer update. See Section 33.4.6, Register buffers updating at interrupt generation for details.

33.3.4 Automatic logarithmic fading (ALD)

The automatic logarithmic decrement function (ALD) can be used to reduce the volume gradually to zero without CPU intervention.
The logarithmic decrease matches the sensitivity of the human ear and creates the impression of a linearly decaying tone.

A tone, started in ALD mode (SGnCONF.SGnMODE[1:0] = 11_B), will automatically fade away.

The fading is stopped if

- the volume has decreased to 0, i.e. the volume compare buffer SGnPWM value becomes 0,
- 000_H is written to the SGnPWM register.

The speed of the volume reduction is controlled by the sound duration factor register SGnDF.

A new volume value is calculated at every (SGnDF + 1) rising or falling edge of the tone frequency signal SGnOF.

The sound duration factor SGnDF can be set in the range 0 to 255.

33.3.4.1 Volume reduction

The calculation of the volume reduction uses an approximation of the logarithmic function by reducing the volume by 1/32 of the previous volume at each reduction step m.

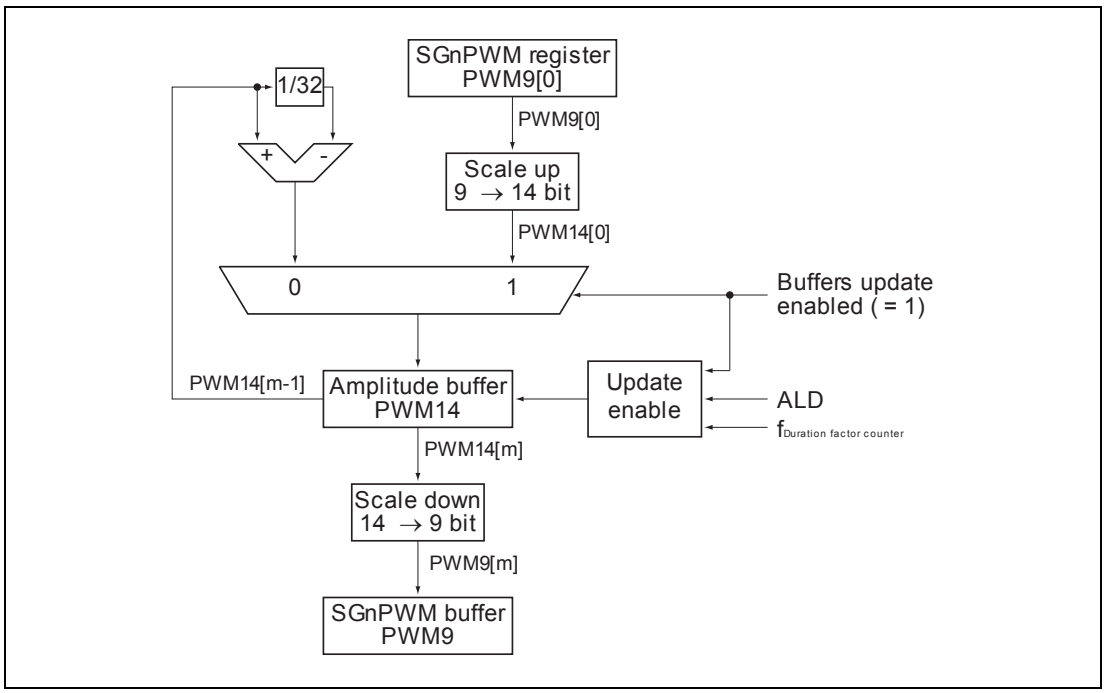


Figure 33.7 Volume calculation in ALD mode

The calculation uses 14-bit arithmetic and follows the procedure:

```
PWM9[0] = SGnPWM;                                  // initial volume
PWM14[0] = PWM9[0] << 5 + 31;                      // scale up to 14 bit
for (m=1, m<M+1; m++){
     PWM14[m] = PWM14[m-1]-(PWM14[m-1]>>5);        // decrement by 1/32
     PWM9[m] = PWM14[m] >> 5;                      // scale down to 9 bit
}
```

where:

PWM9[0]:	initial 9-bit PWM value
PWM14[0]:	initial 14-bit PWM value
PWM9[m]:	new 9-bit PWM value
PWM14[m]:	new 14-bit PWM value
PWM9[m-1]:	previous 9-bit PWM value
PWM14[m-1]:	previous 14-bit PWM value
M:	Number of volume reduction steps until tone stops, i.e. PWM9[M] = 0
m:	volume reduction step number, m = 1 to M

The maximum number of reduction steps M depends on the initial volume PWM9[0].

33.3.4.2 Tone duration

The volume reduction is conducted at the tone frequency signal SGnOF rising or falling edge number m x (SGnDF + 1).

The sound stops when the volume compare buffer SGnPWM value becomes zero.

The maximum number of volume reduction steps M depends on the start volume PWM9[0], i.e. the value set to SGnPWM register. To avoid an initial delay with apparently no effect, the start value shall not exceed the value of register SGnFL by more than 1.

The total duration T_D depends on

- the start volume PWM9[0], i.e. value of the SGnPWM register,
- the sound duration factor set in register SGnDF,
- the tone frequency f_{TONE}.

The diagram shows an example of ALD volume reduction with SGnDF = 2, i.e. every 3rd rising or falling SGnOF edge the tone volume is reduced.

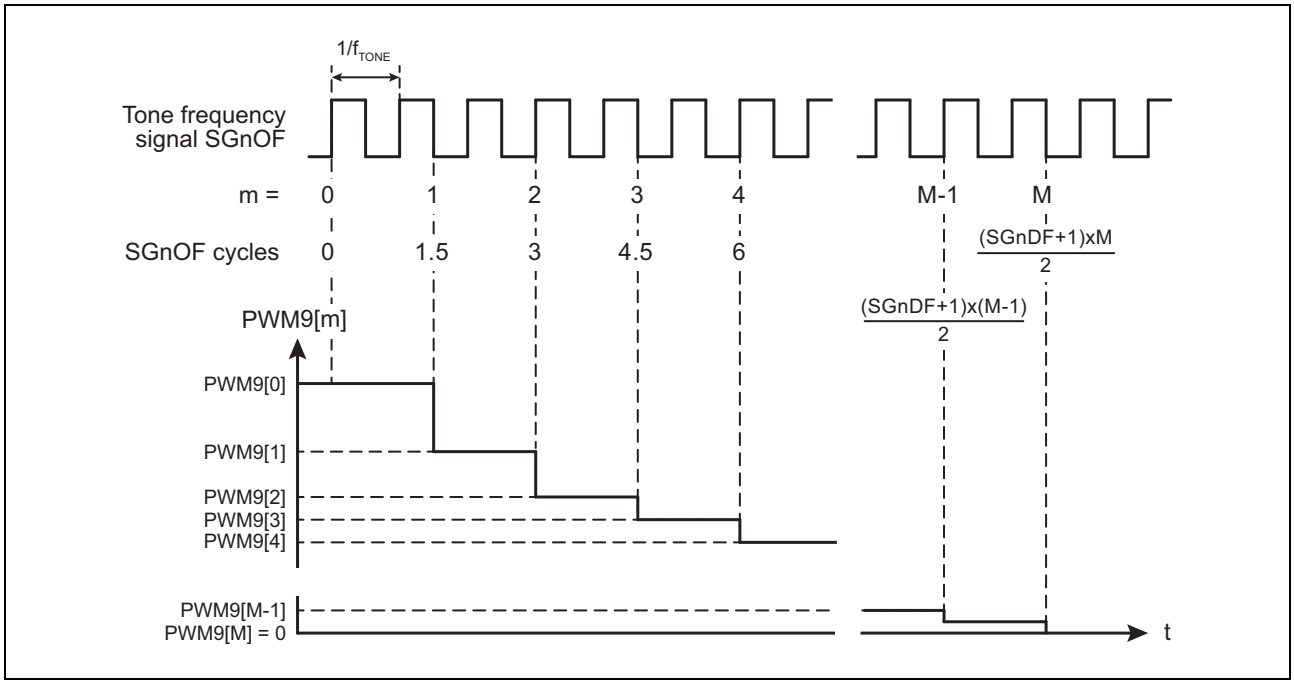


Figure 33.8 ALD mode timing

The tone duration T_D can be calculated as follows:

$$T_D = \frac{1}{2 \bullet f_{TONE}} \bullet (SGnDF + 1) \bullet M$$

where:

T_D:	duration of the tone, i.e. time until tone stops at PWM9[M] = 0
f_{TONE}:	frequency of the tone frequency signal SGnOF
SGnDF:	tone duration factor, i.e. contents of the SGnDF register
M:	Number of volume reduction steps until tone stops, i.e. PWM9[M] = 0

Example

The table below shows two examples of the sound duration for minimum and maximum tone frequency.

The following settings are assumed:

- f_{SGnCLK} = 16 MHz
- SGnFL = 332 (this yields a PWM base frequency of 48.048 kHz)
- SGnPWM = 333 (100 % volume)
- The tone frequency is changed via the SGnFH register:
  - SGnFH = 3 (this yields a tone frequency of 6.006 kHz)
  - SGnFH = 240 (this yields a tone frequency of 99.68 Hz)

With the initial 100 % volume (SGnPWM = 333) it takes M = 202 reduction steps to stop the sound, i.e. M = 202 and PWM9[M] = 0).

Table 33.9 ALD tone duration examples

	Tone duration T_D [sec]			
Tone frequency	SGnDF = 0	SGnDF = 1	...	SGnDF = 255
6006 Hz	0.017	0.034	...	4.305
99.68 Hz	1.013	2.026	...	259.390

NOTE

The present volume value of the SGnPWM buffer can not be read via the SGnPWM register.

33.3.4.3 Interrupt in ALD mode

The interrupt INTSGnTI is generated when the SGnPWM volume buffer reaches or passes below the value of the interrupt threshold register SGnITH at the next falling edge of the tone frequency signal SGnOF.

NOTE

The interrupt will not be generated if the SGnITH buffer value is 000_H.

If buffer update by INTSGnTI is enabled (SGnCONF.SGnBE = 1) the buffers are updated at the generation of the interrupt. See Section 33.3.3, Updating the buffer values for more information about buffer updating.

33.3.5 Automatic decrement/increment (ADI) of the volume

The automatic decrement/increment function (ADI) can be used to reduce or increase the volume without CPU intervention.

A tone, started in ADI mode (SGnCONF.SGnMODE[1:0] = 10_B) will be output with a linear decrementing or incrementing volume.

The volume increment or decrement is stopped if

- the volume has decreased to 0, i.e. the volume compare buffer SGnPWM value becomes 0,

- the volume has increased to the maximum, i.e. the volume compare buffer SGnPWM value becomes $1FF_H$,
- 000_H is written to the SGnPWM register.

The speed of the volume increment or decrement is controlled by the sound duration factor register SGnDF.

A new volume value is calculated at every (SGnDF + 1) rising or falling edge of the tone frequency signal SGnOF.

The sound duration factor SGnDF can be set in the range 0 to 255.

33.3.5.1 Volume reduction

The value to increment respectively decrement the volume buffer is defined by SGnADI.SGnADIL[8:0].
SGnADIL[8:0] can hold positive (increment) and negative (decrement) values. The MSB SGnADIL[8] is the sign bit (1 for negative values) and SGnADIL[7:0] is the absolute value. Thus a increment/decrement value range from -255 ($1FF_H$) to +255 ($0FF_H$) can be specified.

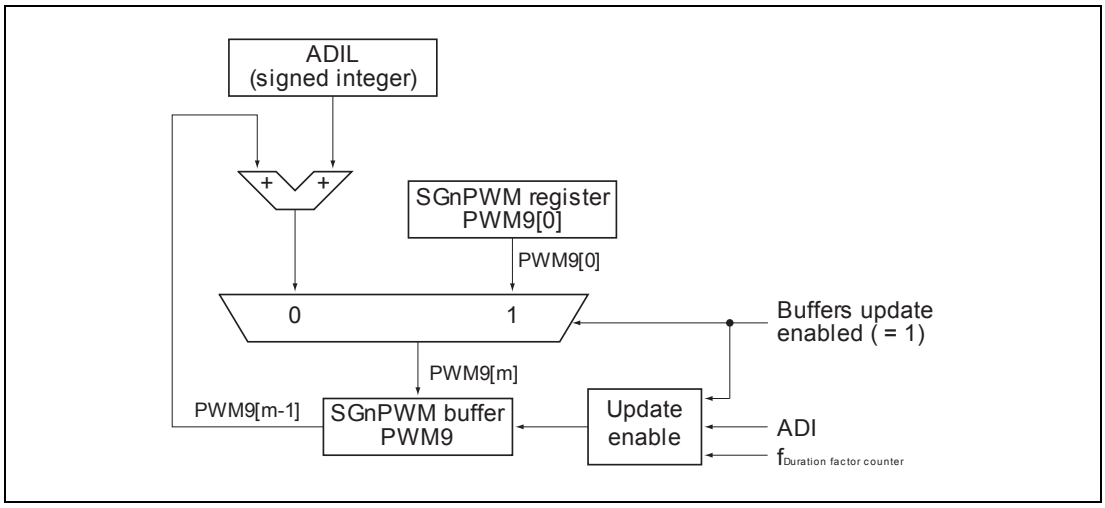


Figure 33.9 Volume calculation in ADI mode

The calculation uses 9-bit arithmetic and follows the procedure:

```
PWM9[0] = SGnPWM;                                   // initial volume
for (m=1, m<M+1; m++){
     PWM9[m] = PWM9[m-1]+ADIL                       // increment/decrement
}
```

where:

PWM9[0]:	initial 9-bit PWM value
PWM9[m]:	new 9-bit PWM value
PWM9[m-1]:	previous 9-bit PWM value
ADIL:	signed integer increment/decrement value SGnADI.SGnADIL[8:0]
M:	Number of volume increment/decrement steps until tone stops, i.e. PWM9[M] = 0 or $1FF_H$
m:	volume reduction step number, m = 1 to M

The maximum number of volume increment/decrement steps M depends on the initial volume PWM9[0]. [M = initial volume(SGnPWM) / decrement values(SGnADI.SGnADIL[8:0])]

33.3.5.2 Tone duration

The volume increment/decrement is conducted at the tone frequency signal SGnOF rising or falling edge number m x (SGnDF + 1).

The sound stops when the volume compare buffer SGnPWM value becomes 000_H or $1FF_H$.

The maximum number of volume increment/decrement steps M depends on the start volume PWM9[0], i.e. the value set to SGnPWM register. To avoid an initial delay with apparently no effect, the start value shall not exceed the value of register SGnFL by more than 1.

The total duration T_D depends on

- the start volume PWM9[0], i.e. value of the SGnPWM register,
- the sound duration factor set in register SGnDF,
- the tone frequency f_{TONE}.

The diagram shows an example of ADI volume reduction with SGnDF = 2, i.e. every 3rd rising or falling SGnOF edge the volume buffer SGnPWM is reduced.

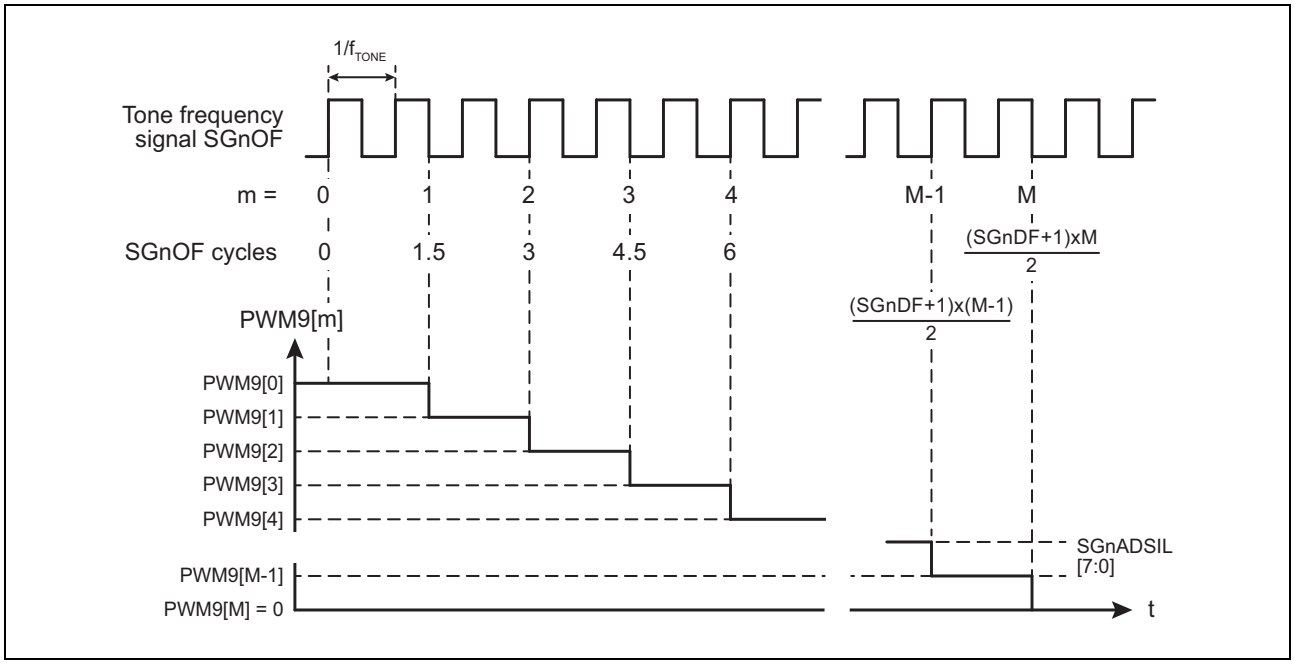


Figure 33.10 ADI mode timing

The tone duration T_D can be calculated as follows:

$$T_D = \frac{1}{2 \bullet f_{TONE}} \bullet (SGnDF + 1) \bullet M$$

where:

T_D:	duration of the tone, i.e. time until tone stops at PWM9[M] = 0
f_{TONE}:	frequency of the tone frequency signal SGnOF
SGnDF:	tone duration factor, i.e. contents of the SGnDF register
M:	Number of volume increment/decrement steps until tone stops, i.e. PWM9[M] = 0

Example

The table below shows two examples of the sound duration for minimum and maximum tone frequency.

The following settings are assumed:

- f_{SGnCLK} = 16 MHz
- SGnFL = 332 (this yields a PWM base frequency of 48.048 kHz)
- SGnPWM = 333 (100 % volume)
- The tone frequency is changed via the SGnFH register:
  - SGnFH = 3 (this yields a tone frequency of 6.006 kHz)
  - SGnFH = 240 (this yields a tone frequency of 99.68 Hz)
- SGnADI.SGnADIL[8:0] = 103_H (decrement by 3)

With the initial 100 % volume (SGnPWM = 333) it takes M = 111 reduction steps to stop the sound, i.e. M = 111 and PWM9[M] = 0). Please refer to the equations in **Section 33.3.5.1** to calculate the "M" value.

Table 33.10 ADI tone duration examples

	Tone duration T_D [sec]			
Tone frequency	SGnDF = 0	SGnDF = 1	...	SGnDF = 255
6006 Hz	0.009	0.018	...	2.366
99.68 Hz	0.557	1.114	...	142.529

NOTE

The present volume value of the SGnPWM buffer can not be read via the SGnPWM register.

33.3.5.3 Interrupt in ADI mode

The interrupt INTSGnTI is generated at the next falling edge of the tone frequency signal SGnOF in the following cases:

- In case of automatic volume increment (positive SGnADI.SGnADIL[8:0] value, i.e. SGnADIL[8] = 0):
  When the SGnPWM volume buffer reaches or goes above the value of the buffer of the interrupt threshold register SGnITH.
- In case of automatic volume decrement (negative SGnADI.SGnADIL[8:0] value, i.e. SGnADIL[8] = 1):
  When the SGnPWM volume buffer reaches or goes below the value of the buffer of the interrupt threshold register SGnITH.

NOTE

The interrupt will not be generated under any of the following conditions:

- if (SGnITH buffer value) = 000_H
- in case of automatic decrement (negative SGnADIL value):
  if ((SGnITH buffer value) + 1) < |(SGnADI buffer value)|
- in case of automatic increment (positive SGnADIL value):
  if ((SGnITH buffer value) - 1) > |(SGnADI buffer value)|
- in case of automatic increment (positive SGnADIL value):
  if (SGnITH buffer value) = $1FF_H$

If buffer update by INTSGnTI is enabled (SGnCONF.SGnBE = 1) the buffers are updated at the generation of the interrupt. See Section 33.3.3, Updating the buffer values for more information about buffer updating.

33.3.5.4 Fade-in with following continuous tone

The maximum volume is defined by the value of the SGnFL register, since SGnPWM ≥ SGnFL yields 100 % duty cycle of the PWM volume signal SGnAO.

If the volume shall increment in ADI mode the tone does not stop until the SGnPWM buffer value reaches its maximum value $1FF_H$, despite of the value of SGnFL. Thus the tone does not stop at maximum volume, defined by SGnFL, but at the maximal SGnPWM register value.

The implementation of a fade-in tone, that keeps a final volume after fade-in completion, can be done by using the INTSGnTI interrupt and the buffer update function, according to the following procedure:

1. Select the ADI mode (SGnCONF.SGnMODE[1:0] = 10_B).
2. Enable register's buffers update by INTSGnTI interrupt (SGnCONF.SGnBE = 1).
3. Set the interrupt trigger level (SGnITH) to the desired final volume level.
   The desired final volume level must not exceed the value of SGnFL.
4. Set the SGnADI and SGnDF registers to define the speed on the volume increment.
5. Start tone by writing start volume to SGnPWM.
6. Set SGnPWM to the desired final volume, i.e. to the same value as in SGnITH. This volume becomes valid after the interrupt INTSGnTI.
7. After the tone fade-in has started set SGnCONF.SGnMODE[1:0] = 00_B in order to change to continuous mode at the next buffers update, i.e. upon occurrence of the first INTSGnTI.
8. If the desired final volume is reached, the interrupt INTSGnTI is asserted, and the SGnPWM volume buffer is updated with the desired volume value from the SGnPWM register.

33.3.6 Automatic duration control (ADC)

The automatic duration control function (ADC) can be used to limit the duration of a constant volume tone.

The ADC mode is selected by SGnCONF.SGnMODE[1:0] = 01_B.

The tone is stopped if

- the specified time has passed,
- 000_H is written to the SGnPWM register.

33.3.6.1 Tone duration

The duration of the tone is determined by

- the initial value of the duration counter value SGnADI.SGnADIL[8:0]
- the duration factor SGnDF.SGnDF[7:0]

The duration counter buffer SGnADI.SGnADIL[8:0] is decrement by 1 at every (SGnDF + 1) rising or falling edge of the tone frequency signal SGnFO.

When the duration counter buffer SGnADI.SGnADIL[8:0] becomes 0, the tone is stopped.

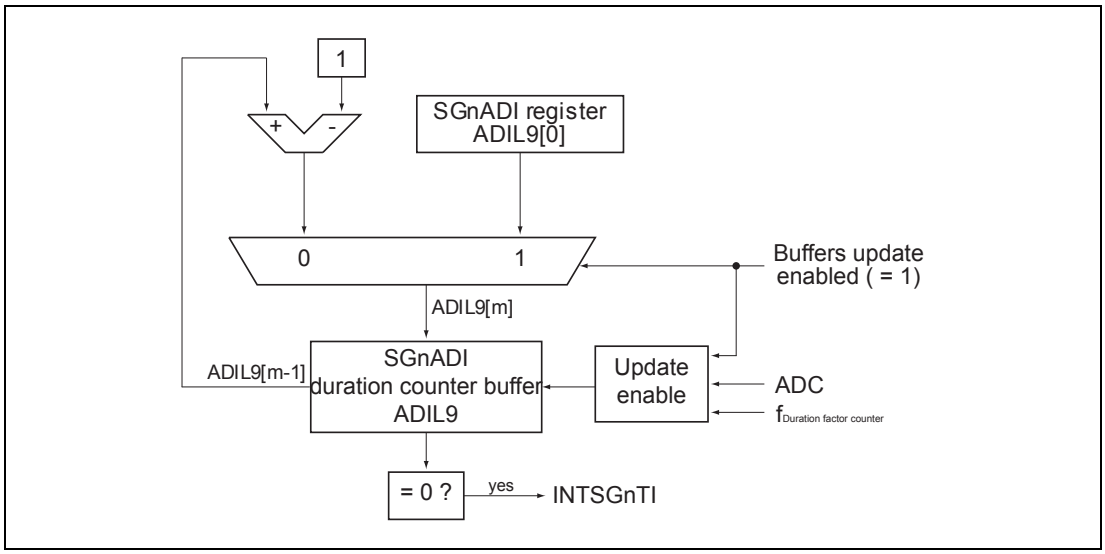


Figure 33.11 Time calculation in ADC mode

The calculation uses 9-bit arithmetic and follows the procedure:

```
ADIL9[0] = SGnADI;                              // initial counter = M
for (m=1, m<M+1; m++){
     ADIL9[m] = ADIL9[m-1]-1                    // decrement by 1
}
```

where:

ADIL9[0]:	initial duration counter value in the SGnADI.SGnADIL[8:0] register
ADIL9[m]:	new duration counter value in the SGnADI.SGnADIL[8:0] buffer
M:	Number of duration counter decrement steps until tone stops, i.e. M = ADIL9[0]
m:	duration counter decrement step number, m = 1 to M

The diagram shows an example of a tone duration control with SGnDF= 2, i.e. every 3rd rising or falling SGnOF edge the duration counter SGnADI.SGnADIL[8:0] is reduced by 1.
ADIL denotes the initial value of the SGnADI buffer, i.e. the SGnADI register contents.

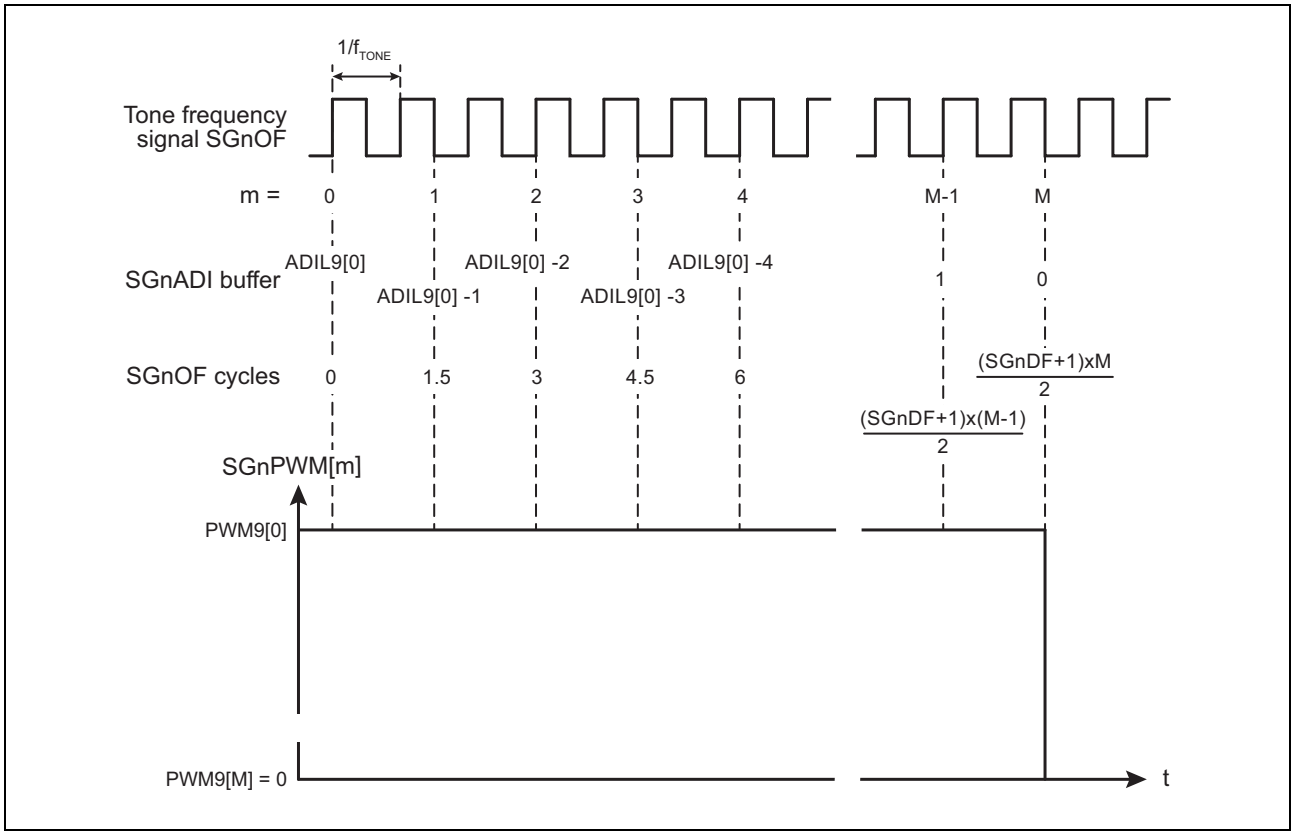


Figure 33.12 ADC mode timing

The tone duration T_D can be calculated as follows:

$$T_D = \frac{1}{2 \bullet f_{TONE}} \bullet (SGnDF + 1) \bullet SGnADIL$$

where:

T_D:	duration of the tone, i.e. time until tone stops at SGnADIL = 0
f_{TONE}:	frequency of the tone frequency signal SGnOF
SGnDF:	tone duration factor, i.e. contents of the SGnDF register
SGnADIL:	initial tone duration counter value SGnADI.SGnADIL[8:0]

Example

The table below shows two examples of the sound duration for minimum and maximum tone frequency.

The following settings are assumed:

- f_{SGnCLK} = 16 MHz
- SGnFL = 332 (this yields a PWM base frequency of 48.048 kHz)
- The tone frequency is changed via the SGnFH register:

  - SGnFH = 3 (this yields a tone frequency of 6.006 kHz)
  - SGnFH = 240 (this yields a tone frequency of 99.68 Hz)
- SGnADI.SGnADIL[8:0] = 64_H = 100 (initial duration counter value)

Table 33.11 ADC tone duration examples

	Tone duration T_D [sec]			
Tone frequency	SGnDF = 0	SGnDF = 1	...	SGnDF = 255
6006 Hz	0.008	0.017	...	2.131
99.68 Hz	0.502	1.003	...	128.405

NOTE

The present tone duration counter value of the SGnADI buffer can not be read via the SGnADI register.

33.3.6.2 Interrupt in ADC mode

The interrupt INTSGnTI is generated when the tone is stopped.

If buffer update by INTSGnTI is enabled (SGnCONF.SGnBE = 1) the buffers are updated at the generation of the interrupt. See Section 33.3.3, Updating the buffer values for more information about buffer updating.

33.3.7 INTSGnTI interrupt

In ALD, ADI and ADC mode the interrupt INTSGnTI can be generated:

- In ALD mode the interrupt INTSGnTI is generated when the SGnPWM volume buffer reaches or passes below the value of the interrupt threshold register SGnITH at next falling edge of the tone frequency signal SGnOF.
- In ADI mode, the interrupt generation depends on the sign of the volume offset:
  - If the volume offset is *positive* (positive value in the SGnADI buffer), the interrupt INTSGnTI is generated when the SGnPWM buffer value is *equal or above* the value of the SGnITH buffer.
  - If the volume offset is *negative* (negative value in SGnADI buffer), the interrupt INTSGnTI is generated when the SGnPWM buffer value is *equal or below* the value of the SGnITH buffer.
- In ADC mode the interrupt INTSGnTI is generated when the sound stops.

The following figure illustrates the interrupt generation in all three modes.

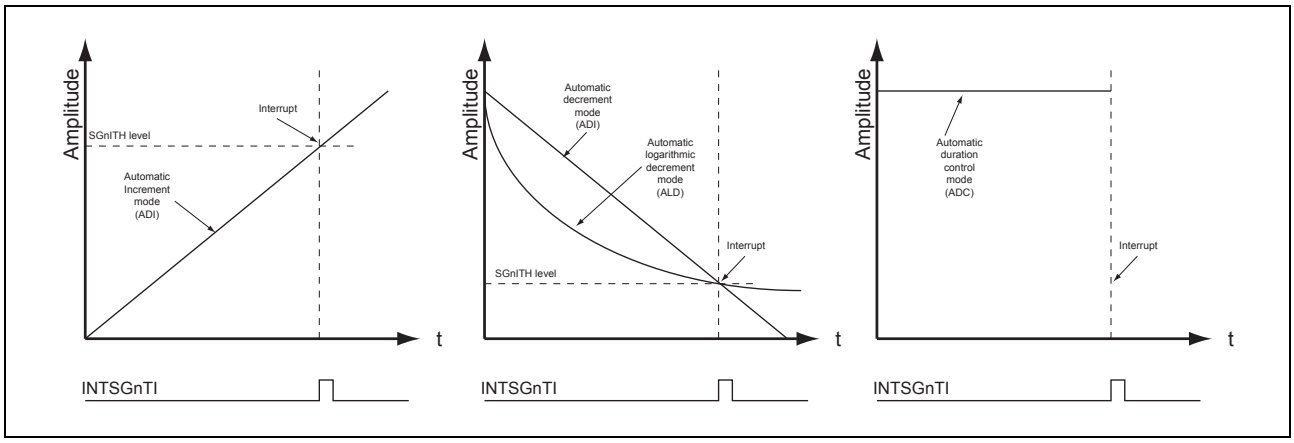


Figure 33.13 Interrupt generation

33.4 Sound Generator Application Hints

This section provides supplementary programming information.

33.4.1 Initialization

- To enable the Sound Generator, set SGnCTL.SGnEN = 1.
  This applies the clock SGnCLK to the Sound Generator.
- Select the desired mode:
  - SGnCONF.SGnMODE[1:0] = 00_B: continuous sound is output.
  - SGnCONF.SGnMODE[1:0] = 01_B: ADC mode is selected.
    The duration of sound output is limited as specified by the sound duration factor register SGnDF.
  - SGnCONF.SGnMODE[1:0] = 10_B: ADI mode is selected.
    The volume is faded in or out. The speed of decrease/increase of the volume is controlled by the register SGnDF.
  - SGnCONF.SGnMODE[1:0] = 11_B: ALD mode is selected.
    The volume is faded out logarithmically. This is detected by the human ear as a linearly fading sound.
- Enable or disable buffer update by interrupt INTSGnTI:
  - SGnCONF.SGnBE = 0: buffers updating is enabled by writing to the SGnPWM register.
  - SGnCONF.SGnBE = 1: buffers updating is performed with the next INTSGnTI interrupt.
- Specify the interrupt threshold level SGnITH.SGnITH[8:0].
- Select the output signal for SGTFAO:
  - SGnCONF.SGnOS = 0: SGTFAO outputs the tone frequency signal SGnOF
  - SGnCONF.SGnOS = 1: SGTFAO outputs the composite signal SGnO
- Select the output signal for SGTFAOL:

  - SGnCONF.SGnOSI = 0: SGTFAOL outputs fixed low level.
  - SGnCONF.SGnOSI = 1: SGTFAOL outputs the inverted composite signal SGnO
- Specify the tone to be generated:
  - Tone frequency: SGnFL and SGnFH
  - Volume: SGnFL and SGnPWM

  Writing the volume value to SGnPWM starts output of the tone.

33.4.2 Start sound

The sound is started by writing a non-zero value to the volume register SGnPWM.

NOTE

If sound was stopped before by writing SGnPWM = 0, the SGnPWM value to restart the sound must be greater than 1.

- If SGnCTL.SGnEN = 1 the sound generation will start immediately when writing to the SGnPWM register.
- If SGnCTL.SGnEN = 0 the sound starts after setting the SGnCTL.SGnEN = 1.
  Refer to Section 33.4.4.1, Avoidance of sound artefacts for details about pause termination.

NOTE

Before starting the sound, all other register settings must be made. See **Section 33.4.1, Initialization**.

33.4.3 Stop sound

33.4.3.1 Stop sound by the application program

The sound is stopped when 000_H is written to volume register SGnPWM, regardless of the setting of the SGnCONF.SGnBE bit.

33.4.3.2 Stop sound by automatic logarithmic decrement function

In ALD mode (SGnCONF.SGnMODE[1:0] = 11_B), the sound is stopped when the SGnPWM volume buffer reaches the value 000_H.

33.4.3.3 Stop sound by automatic decrement/increment function

In ADI mode (SGnCONF.SGnMODE[1:0] = 10_B), the sound is stopped when the SGnPWM volume buffer

- reaches or underruns the value 000_H (with decrement).
- reaches or overruns the value $1FF_H$ (with increment).

33.4.3.4 Stop sound by automatic duration control function

In ADC mode (SGnCONF.SGnMODE[1:0] = 01_B), the sound is stopped when the value stored in the SGnADI buffer register is decremented to 0.

33.4.4 Pause Sound

While sound generation is ongoing the sound can be paused by setting SGnCTL.SGnEN = 0. The sound will be paused with the next edge of the PCLK clock and all output signals remain at their current state.

Note that no synchronization with the tone frequency signal SGnOF is taking place, thus audible noise may be generated.

Sound generation continues after setting SGnCTL.SGnEN = 1.

33.4.4.1 Avoidance of sound artefacts

For avoiding sound artefacts, when the pause mode is terminated, it has to be prevented that the PWM signals will affect the output pins in this case.

Therefore it is recommended to perform following sequence:

1. Activate pause mode by SGnCTL.SGnEN = 0.
2. Read the current PWM buffer value via the volume register SGnPWM and store as a variable.
3. Set SGnPWM = 0.
4. Change the ports, that are used for the Sound Generator output signals, to port mode (PMCn_m = 0) and set the port output to low level (Pn_m = 0).
5. Terminate pause mode by SGnCTL.SGnEN = 1.
6. Wait at least for half of the sound frequency period.
7. Change the ports back to alternative function (PMCn_m = 0), i.e. the ports act again as Sound Generator outputs.
8. Restart the sound by writing the previously stored variable back to the volume register SGnPWM. Note that the SGnPWM value has to be greater than 1.
   The sound will start without sound artefacts.

33.4.5 Change tone volume

The tone volume is changed by writing a new volume value to SGnPWM register.

The new volume becomes effective upon the next buffers update:

If buffers update by INTSGnTI interrupt is enabled (SGnCONF.SGnBE = 1), the new volume becomes effective with the next INTSGnTI interrupt.

If buffers update by INTSGnTI interrupt is disabled (SGnCONF.SGnBE = 0), the new volume becomes effective with the next edge of the tone frequency signal SGnOF.

In either case the new volume becomes effective with a falling edge of the tone frequency signal SGnOF.

33.4.6 Register buffers updating at interrupt generation

If SGnCONF.SGnBE = 1 the buffers will be loaded with the values stored in their related registers at the generation of the INTSGnTI interrupt, that is generated according to the interrupt threshold register (SGnITH) value.

Furthermore the update process of the buffers due to a SGnPWM register write is disabled.

Therefore, when SGnCONF.SGnBE = 1 it is possible to load the next tone parameters into the registers while tone generation is ongoing. The next tone is started automatically when the INTSGnTI interrupts occurs.

For more details on register buffering see Section 33.3.3, Updating the buffer values.

CAUTION

Even if SGnCONF.SGnBE = 1 writing SGnPWM = 00_H stops the sound immediately, i.e. not at the generation of the next INTSGnTI interrupt. See Section 33.4.3, Stop sound for details about stopping of the sound.

33.4.7 Constant sound volume

A sound started in continuous output mode (SGnCONF.SGnMODE[1:0] = 00_B) is output with the volume value written to SGnPWM. The sound is output continually and does not stop automatically. It has to be stopped by writing 000_H to SGnPWM.

33.4.8 Generate special sounds

To generate special sounds (like blinker clicks etc.), frequency and volume can be changed simultaneously.

To change the frequency of a sound that has already started:

1. Write to frequency register SGnFL and/or SGnFH.
2. Write to volume register SGnPWM.

33.4.9 Output signal control

Table 33.12 Sound Generator output signals control

Register settings			Output signals		
SGnCONF.SGnOS	SGnCONF.SGnOSI	SGnPWM	SGTFAO	SGTFAOL	SGTAO
0	0	0	L	L	L
0	1	0	L	H	L
1	0	0	L	L	L
1	1	0	L	H	L
0	0	≠ 0	f_{TONE}	L	f_{PWM}
0	1	≠ 0	f_{TONE}	~(f_{TONE} & f_{PWM})	f_{PWM}
1	0	≠ 0	f_{TONE} & f_{PWM}	L	f_{PWM}
1	1	≠ 0	f_{TONE} & f_{PWM}	~(f_{TONE} & f_{PWM})	f_{PWM}

33.5 Sound Generator Registers

This section contains a description of all registers of the Sound Generator.

33.5.1 SG registers overview

The SG is controlled and operated by the following registers:

Table 33.13 SG register overview

Register name	Shortcut	Address
SGn control register	SGnCTL	<SGn_base> + 00_H
SGn status register	SGnSTAT	<SGn_base> + 04_H
SGn configuration register	SGnCONF	<SGn_base> + 08_H
SGn duration factor register	SGnDF	<SGn_base> + $0C_H$
SGn automatic decrement/increment register	SGnADI	<SGn_base> + 10_H
SGn interrupt threshold register	SGnITH	<SGn_base> + 14_H
SGn frequency low register	SGnFL	<SGn_base> + 18_H
SGn frequency high register	SGnFH	<SGn_base> + $1C_H$
SGn volume register	SGnPWM	<SGn_base> + 20_H
SGn emulation register	SGnEMU	<SGn_base> + 24_H

<SGn_base>

The base addresses <SGn_base> of the SGn is defined in the first section of this chapter under the key word “Register addresses”.

33.5.2 Sound Generator registers details

33.5.2.1 SGnCTL – Control register

This register controls the Sound Generator clock supply.

Access: This register can be read/written in 32-bit units.

Address: <SGn_base> + 00_H

Initial Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	SGnCLKDIV[2:0]			SGnEN
R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 33.14 SGnCTL register contents

Bit position	Bit name	Function
31 to 4	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
3 to 1	SGn CLKDIV[2:0]	PCLK clock divisor for generation of the SGnCLK clock
0	SGnEN	Sound Generator clock enable / disable 0: Sound Generator clock is disabled and does not operate. Access to all registers is still possible. 1: Sound Generator clock is enabled and ready to use.

SGnCLKDIV[2:0]	f_{SGnCLK}
000_B	f_{PCLK}
001_B	f_{PCLK} / 2
010_B	f_{PCLK} / 3
011_B	f_{PCLK} / 4
100_B	f_{PCLK} / 5
101_B	f_{PCLK} / 6
110_B	f_{PCLK} / 7
111_B	f_{PCLK} / 8

Note: The SGnCLKDIV[2:0] bits shall only be changed when SGnEN = 0.

If SGnCTL.SGnEN is set to ’0’ while sound generation is ongoing, the sound will be stopped immediately. No synchronization to the sound signal takes place.

NOTE

Change the contents of this register only when the sound is stopped (register SGnPWM cleared).

33.5.2.2 SGnSTAT – Status register

This register indicates the status of the sound generation.

Access: This register can be read in 32-bit units.

Address: <SGn_base> + 04_H

Initial Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SGn RUN
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 33.15 SGnSTAT register contents

Bit position	Bit name	Function
31 to 1	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
0	SGnRUN	Sound generation status 0: No sound is generated 1: Sound generation is ongoing SGnRUN is 0 in the following cases • SGnEN = 0 (Sound Generator clock is switched off) • SGnPWM buffer = 000_H • SGnPWM buffer = $1FF_H$ in ADI mode (positive saturation)

33.5.2.3 SGnFH – Frequency high register

This register is used to specify the final tone frequency. It holds the target value for the 9-bit counter SGnFH.

Tone compare buffer

The value of this register becomes effective only when it is copied to the SGnFH tone compare buffer. For details on register buffering refer to Section 33.3.3, Updating the buffer values.

Access: This register can be read/written in 32-bit units.

Address: <SGn_base> + $1C_H$

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	Counter SGnFH target value								
R	R	R	R	R	R	R	R/W								

Table 33.16 SGnFH register contents

Bit position	Bit name	Function
31 to 9	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
8 to 0	SGnFH[8:0]	Higher part of the frequency value

For the calculation of the resulting tone frequency refer to Section 33.3.1.2, Tone frequency calculation.

NOTES

1. Legal values depend on the contents of register SGnFL.
   SGnFL defines the frequency of the PWM base frequency, which is the input clock to the tone generator.
2. The value read from this register does not necessarily reflect the current tone frequency, because this frequency is determined by the frequency compare buffer value. The buffer might not be updated yet.
   For details see Section 33.3.1.1, Updating the frequency and tone buffer values.

33.5.2.4 SGnFL – Frequency low register

This register is used to specify the target value for the PWM frequency. It holds the target value for the 9-bit counter SGnFL.

Frequency compare buffer

The value of this register becomes effective only when it is copied to the SGnFL frequency compare buffer. For details on register buffering refer to Section 33.3.3, Updating the buffer values.

Access: This register can be read/written in 32-bit units.

Address: <SGn_base> + 18_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	Counter SGnFL target value								
R	R	R	R	R	R	R	R/W								

Table 33.17 SGnFL register contents

Bit position	Bit name	Function
31 to 9	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
8 to 0	SGnFL[8:0]	Lower part of the frequency value

For the calculation of the resulting PWM frequency refer to Section 33.3.2.1, PWM calculations.

The value written to SGnFL defines also the reference value for the maximum sound amplitude (100 % PWM duty cycle).
A 100 % duty cycle (continually high) will be generated if the SGnPWM value is higher than the SGnFL value.
For details see Section 33.3.2.1, PWM calculations).

NOTES

1. The maximum value to be written is 510 ($01FE_H$). This yields a PWM frequency of 31.3 kHz (with PCLK = 16 MHz).
   The minimum value to be written depends on the capability of the external circuit. A value of 255 ($00FF_H$) would yield a PWM frequency of 62.5 kHz (with PCLK = 16 MHz).
2. The value read from this register does not necessarily reflect the current PWM frequency, because this frequency is determined by the frequency compare buffer value. The buffer might not be updated yet.
   For details see Section 33.3.1.1, Updating the frequency and tone buffer values.

33.5.2.5 SGnPWM – Volume register

This register is used to specify the tone volume. It holds the target value for the sound amplitude that is given by the duty cycle of the PWM signal.
In ALD or ADI mode, the SGnPWM register holds the start value.

Volume compare buffer

The value of this register becomes effective only when it is copied to the SGnPWM volume compare buffer. For details on register buffering refer to Section 33.3.3, Updating the buffer values.

Access: This register can be read/written in 32-bit units.
When SGnCTL.SGnEN = 0 not the SGnPWM register value will be read but the current value of the SGnPWM buffer.

Address: <SGn_base> + 20_H

Initial Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	Volume target value								
R	R	R	R	R	R	R	R/W								

Table 33.18 SGnPWM register contents

Bit position	Bit name	Function
31 to 9	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
8 to 0	SGnPWM[8:0]	Volume target value

The value written to this register must be considered in conjunction with the contents of register SGnFL. The register SGnFL specifies the maximum value of the counter SGnFL. For the calculation of the resulting duty cycle refer to Section 33.3.2.1, PWM calculations.

NOTES

1. The value read from this register does not necessarily reflect the current volume, because the value of counter SGnFL is compared with the contents of the volume buffer. The buffer might not be updated yet or changed by the ALD or ADI function.
2. The value of this register remains unchanged when the ALD or ADI is switched on.
3. The sound stops immediately when this register is cleared.

33.5.2.6 SGnDF – Duration factor register

This register is used to specify the duration of the sound.

- In ALD mode, it contains the value of the step size of volume recalculation.
- In ADC/ADI mode, it contains the value of the step size of counter decrement/increment.

Duration factor counter buffer

The value of this register becomes effective only when it is copied to the SGnDF duration factor counter buffer. For details on register buffering refer to Section 33.3.3, Updating the buffer values.

Access: This register can be read/written in 32-bit units.

Address: <SGn_base> + $0C_H$

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SGnDF[7:0]							
R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 33.19 SGnDF register contents

Bit position	Bit name	Function
31 to 8	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
7 to 0	SGnDF[7:0]	Duration factor for the ALD, ADI, and ADC mode • In ALD and ADI modes, a new volume value is calculated at every (SGnDF + 1) rising or falling edge of the tone frequency signal SGnOF. See Section 33.3.4, Automatic logarithmic fading (ALD) and Section 33.3.5, Automatic decrement/increment (ADI) of the volume for details. • In ADC mode the value of the tone duration buffer SGnADI is decreased at every (SGnDF + 1) rising or falling edge of the tone frequency signal SGnOF. See Section 33.3.6, Automatic duration control (ADC) for details.

NOTE

Change the contents of this register only when the sound is stopped (register SGnPWM cleared).

33.5.2.7 SGnCONF – Configuration register

This register is used to select output signals, buffer update method and the operation mode.

Configuration buffer

The value of this register becomes effective only when it is copied to the SGnCONF configuration buffer. For details on register buffering refer to Section 33.3.3, Updating the buffer values.

Access: This register can be read/written in 32-bit units.
Nevertheless the value of the register becomes effective only when the register value is buffered. For details on register buffering refer to Section 33.3.3, Updating the buffer values.

Address: <SGn_base> + 08_H

Initial Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	SGn OSI	SGn BE	SGn OS	SGn MODE[1:0]	
R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 33.20 SGnCONF register contents

Bit position	Bit name	Function
31 to 5	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
4	SGnOSI	SGTFAOL output selection 0: Low level output at SGTFAOL 1: Inversion of composite signal SGnO output at SGTFAOL
3	SGnBE	Registers buffers update method selection: 0: Buffer update by interrupt INTSGnTI disabled Buffer update by write to volume register SGnPWM enabled 1: Buffer update by interrupt INTSGnTI enabled Buffer update by write to volume register SGnPWM disabled For more details see Section 33.3.3, Updating the buffer values.
2	SGnOS	SGTFAO output selection 0: Tone frequency signal SGnOF output at SGTFAO 1: Composite signal SGnO output at SGTFAO
1 to 0	SGnMODE[1:0]	Mode selection: 00_B: Continuous sound output selected 01_B: ADC mode selected 10_B: ADI mode selected 11_B: ALD mode selected

NOTE

Change the contents of this register only when the sound is stopped (register SGnPWM cleared).

33.5.2.8 SGnADI – Automatic decrement/increment register

The function of this register depends on the mode:

- ADI mode: linear volume increment or decrement.
- ADC mode: initial value of the decrement/increment buffer.

Decrement/increment buffer

The value of this register becomes effective only when it is copied to the SGnADI decrement/increment buffer. For details on register buffering refer to Section 33.3.3, Updating the buffer values.

Access: This register can be read/written in 32-bit units.

Address: <SGn_base> + 10_H

Initial Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SGnADIL[8:0]								
R	R	R	R	R	R	R	R/W								

Table 33.21 SGnADI register contents

Bit position	Bit name	Function
31 to 9	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
8 to 0	SGnADIL[8:0]	In ADI mode: Volume increment/decrement value (signed integer) In ADC mode: Duration counter value

- In ADI mode the amplitude value stored in the SGnPWM compare buffer is decremented/incremented by the value stored in the SDnADI buffer. SGnADIL[8] is the sign bit and SGnADIL[7:0] the absolute value.
- In ADC mode the duration of the sound is controlled by the value stored in the SDnADI buffer.

See Section 33.3.6, Automatic duration control (ADC) and Section 33.3.5, Automatic decrement/increment (ADI) of the volume for details.

NOTE

Change the contents of this register only when the sound is stopped (register SGnPWM cleared).

33.5.2.9 SGnITH – Interrupt threshold register

In ALD and ADI modes, the volume is compared against the value of SGnITH. A *single* interrupt is generated at the next falling edge of the tone frequency signal SGnOF in the following cases:

- In ALD mode when the volume is less or equal to the SGnITH value.
- In ADI mode, depending on volume increment or decrement:
  - volume increment:
    When the volume is equal to or above the SGnITH value.
  - volume decrement:
    When the volume is equal to or below the SGnITH value.

Threshold buffer

The value of this register becomes effective only when it is copied to the SGnITH threshold buffer. For details on register buffering refer to Section 33.3.3, Updating the buffer values.

Access: This register can be read/written in 32-bit units.

Address: <SGn_base> + 14_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SGnITH[8:0]								
R	R	R	R	R	R	R	R/W								

Table 33.22 SGnITH register contents

Bit position	Bit name	Function
31 to 9	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
8 to 0	SGnITH[8:0]	Interrupt threshold value This is the amplitude threshold that causes an interrupt when it is compared against amplitude values in ALD and ADI modes.

NOTE

Change the contents of this register only when the sound is stopped (register SGnPWM cleared).

Refer details on INTSGnTI interrupt generation, refer to Section 33.3.7, INTSGnTI interrupt.

33.5.2.10 SGnEMU - Emulation register

This register controls whether the Sound Generator can be stopped during emulation, for instance upon a breakpoint hit.

Access: This register can be read/written in 32-bit units.

Address: <SGn_base> + 24_H

Initial Value: 0000_0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	SGnSVSDIS	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R

Table 33.23 SGnEMU register contents

Bit position	Bit name	Function
31 to 8	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
7	SGn SVSDIS	Emulation control 0: Sound Generator can be stopped during emulation 1: Sound Generator continuous operating during emulation
6 to 0	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.

Section 34 PCM-PWM Converter (PCMP)

This section contains a generic description of the PCM-PWM Converter.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

34.1 Overview of the RH850/D1L/D1M PCM-PWM Converters

34.1.1 Units

This microcontroller has the following number of units of the PCM-PWM Converter.

Table 34.1 Units

PCM-PWM Converter	
Units	1
Names	PCMP0

Unit index n

Throughout this section, the individual units of the PCM-PWM Converters are identified by the index “n” (n = 0), for example PCMPnCTL for the PCMPn control register.

Channel index x

The two output channels for stereo data are identified by index “x”:

- x = 0 indicates the right channel
- x = 1 indicates the left channel

34.1.2 Register addresses

All PCM-PWM Converter register addresses are given as address offsets from the individual base addresses <PCMPn_base>.
The <PCMPn_base> addresses of each PCMPn are listed in the following table:

Table 34.2 Register base addresses <PCMPn_base>

PCMPn unit	<PCMPn_base> address
PCMP0	FFF1 0000_H

34.1.3 Clock supply

All PCM-PWM Converters provide one clock input.

Table 34.3 Clock supply

PCMPn unit	PCMPn clock	Connected to
PCMP0	PCLK	Clock Controller CLKFIX

34.1.4 Interrupts and DMA

The PCM-PWM Converters can generate the following interrupt and DMA requests:

Table 34.4 PCMPn interrupt and DMA requests

PCMPn signals	Function	Connected to
PCMPTIFFIL	FIFO buffer fill interrupt	Interrupt Controller INTPCMP0FFIL DMA Controller trigger ID 117
PCMPTIFERR	Error interrupt	Interrupt Controller INTPCMP0FERR

34.1.5 Reset sources

The PCM-PWM Converters and their registers are initialized by the following reset signal:

Table 34.5 Reset sources

PCMPn unit	Reset signal
PCMP0	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

34.1.6 I/O signals

The following table shows the I/O signals of the PCM-PWM Converters.

Table 34.6 I/O signals connections

PCMPn signal	Function	Connected to
PCMP0:		
PCMPTPWMAP0	Pulse-width modulated output AP channel 0	Port PCMP0AP0
PCMPTPWMAN0	Pulse-width modulated output AN channel 0	Port PCMP0AN0
PCMPTPWMBP0	Pulse-width modulated output BP channel 0	Port PCMP0BP0
PCMPTPWMBN0	Pulse-width modulated output BN channel 0	Port PCMP0BN0
PCMPTPWMAP1	Pulse-width modulated output AP channel 1	Port PCMP0AP1
PCMPTPWMAN1	Pulse-width modulated output AN channel 1	Port PCMP0AN1
PCMPTPWMBP1	Pulse-width modulated output BP channel 1	Port PCMP0BP1
PCMPTPWMBN1	Pulse-width modulated output BN channel 1	Port PCMP0BN1

XOR Compare Unit check of PCMP output signals

All PCMP output signals can be checked by the XOR Compare Unit.

34.2 Functional Overview

Features summary

The PCM-PWM module has the following features:

- Converts pulse-code modulated (PCM) audio samples to pulse-width modulated (PWM) signals
- Carrier frequency up to 40 MHz
- Two separate output channels for stereo data
- Programmable output sample frequency in the range 10 kHz to 60 kHz
- Output sample resolution up to 16 bits
- Two output modes to drive both half H-bridges and full H-bridges
- Static signal output function
- Double output mode

The following block diagram shows the main components of the PCM-PWM module. Note that the module has two output channels identified by the index “x” (x = 0, 1).

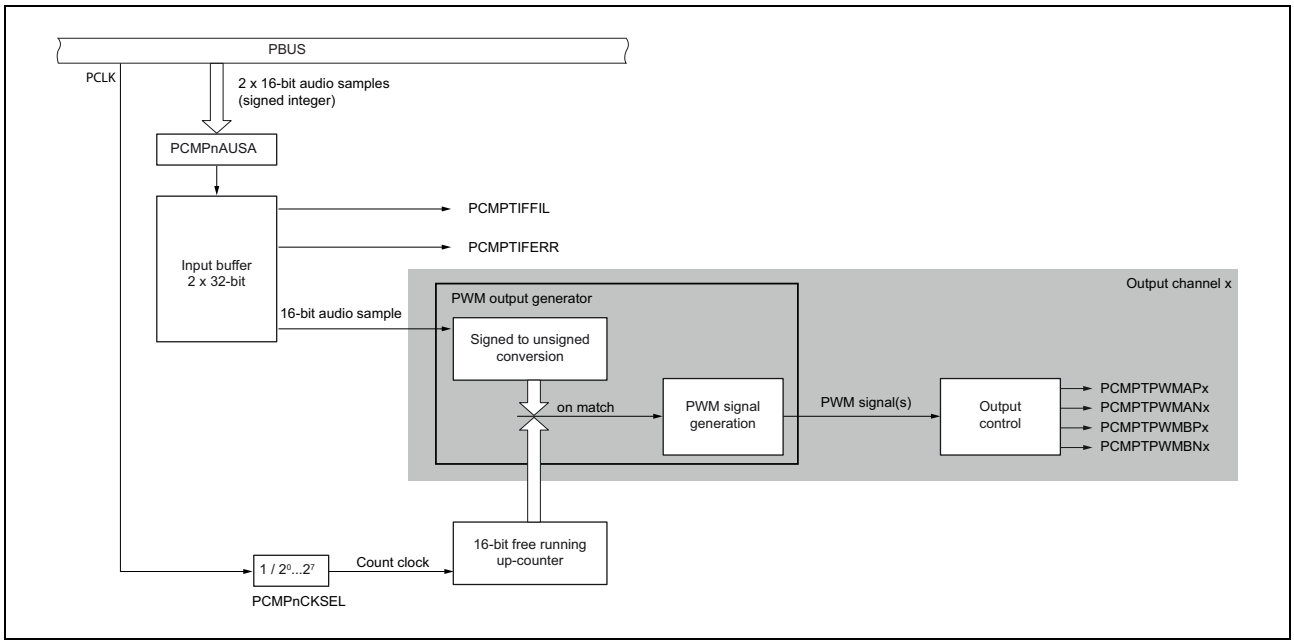


Figure 34.1 Functional blocks of the PCM-PWM module

34.3 Functional Description

The PCM-PWM module converts pulse-code modulated audio samples to pulse-width modulated signals. Two output modes are supported: either for driving a half H-bridge or for driving a full H-bridge. Refer to Section 34.3.4, Output modes.

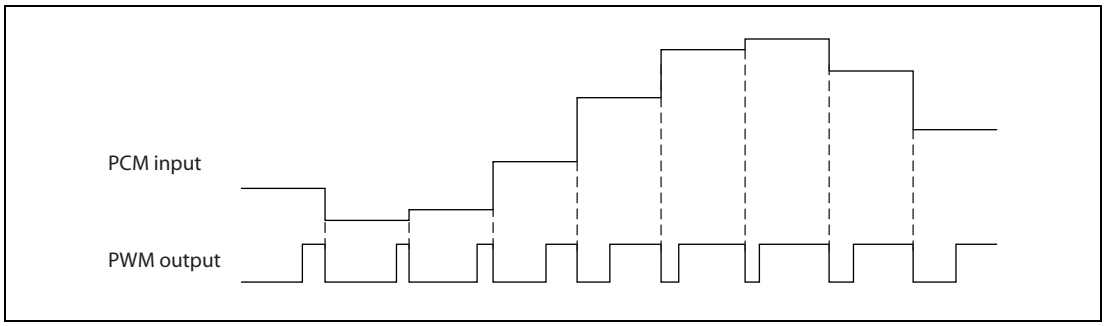


Figure 34.2 Converting PCM audio signals to PWM signals

Writing audio samples to the module

Audio samples are written to the audio sample input register PCMPnAUSA. Contents of the input register are transferred to the FIFO buffer.

For details on the FIFO buffer, refer to Section 34.3.1, PCM-coded data input.

Processing audio samples

The FIFO buffer sends the audio sample to be processed to the output channel. In mono mode, the same audio sample is sent to both output channels; in stereo mode, separate audio samples are sent. In double output mode, each audio sample is sent twice. Refer to Section 34.3.2, Mono mode / stereo mode and Section 34.3.3, Output sampling frequency for details.

Generating PWM output

The output channels first convert the PCM-coded data to an unsigned integer. A PWM signal with the corresponding duty cycle is then generated according to this value. For details refer to Section 34.3.5, PCM-PWM output generators and Section 34.3.3, Output sampling frequency.

The output control block generates the corresponding output signals. For details refer to Section 34.3.6, Output control.

34.3.1 PCM-coded data input

34.3.1.1 Audio sample format

The PCM-PWM module takes as input 16-bit signed integers. The PCM-coded data must be two's complement.

NOTES

1. If audio samples with less than 16 bits are used, they have to be sign-extended to 16 bits by the application to ensure correct interpretation of negative values.
2. Sample resolution and sample frequency is in a tradeoff. The application writing the audio signals to the PCM-PWM must scale values by sample resolution.

34.3.1.2 FIFO buffer

The PCM-PWM module has a FIFO buffer for incoming PCM-coded data. The FIFO buffer has two 32-bit blocks. Each 32-bit block stores two 16-bit audio samples.

In stereo mode (PCMPnCTL.PCMPnSTEREO = 1), the FIFO buffer stores one 32-bit stereo sample in each block: 16 bits for the left audio channel and 16 bits for the right audio channel.

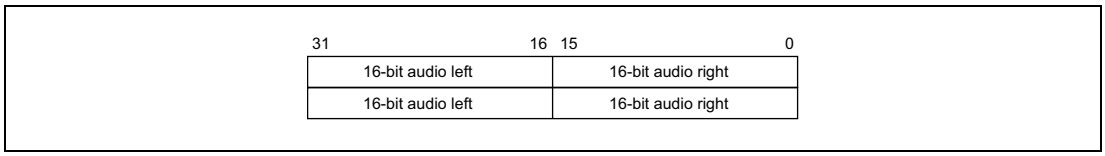


Figure 34.3 FIFO buffer in stereo mode

In mono mode (PCMPnCTL.PCMPnSTEREO = 0), the FIFO buffer stores two 16-bit audio samples in each block. The audio sample in the lower bit range is processed first.

See Section 34.3.2, Mono mode / stereo mode.

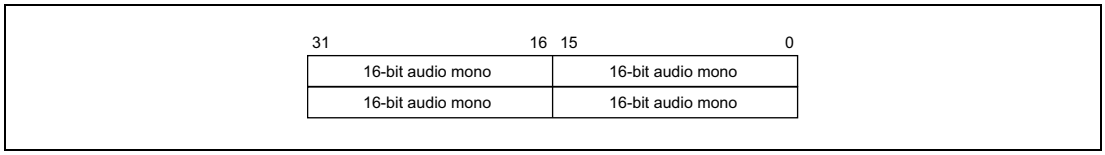


Figure 34.4 FIFO buffer in mono mode

FIFO fill request

The FIFO buffer fill state is indicated by bits PCMPnSTR.PCMPnFIFO[1:0].

When one of the blocks is empty (after processing of one stereo sample or two mono samples), the FIFO buffer fill state changes from full (PCMPnSTR.PCMPnFIFO[1:0] = 2) to not full (PCMPnSTR.PCMPnFIFO[1:0] = 1), which generates the FIFO fill request interrupt PCMPTIFFIL.

Data overflow and underrun

An overflow occurs if the FIFO buffer is full and a new audio sample is written to PCMPnAUSA. In this case, the FIFO buffer sets the bit PCMPnSTR.PCMPnOV = 1.

An underrun occurs if the FIFO buffer tries to get the next audio sample and PCMPnAUSA is empty. In this case, the FIFO buffer sets the bit PCMPnSTR.PCMPnUR = 1. The PCM-PWM will process the last audio sample again.

In both cases, the error interrupt PCMPTIFERR will be generated. Note that if the overflow or underrun condition is still pending (PCMPnSTR.PCMPnOV = 1 or PCMPnSTR.PCMPnUR = 1), no additional error interrupt request is generated.

34.3.2 Mono mode / stereo mode

The PCM-PWM module can generate either mono or stereo output. The bit PCMPnCTL.PCMPnSTEREO defines whether stereo mode is enabled.

34.3.2.1 Mono mode

In mono mode (PCMPnCTL.PCMPnSTEREO = 0), the PCM-PWM module processes the same audio sample simultaneously in both output channels:

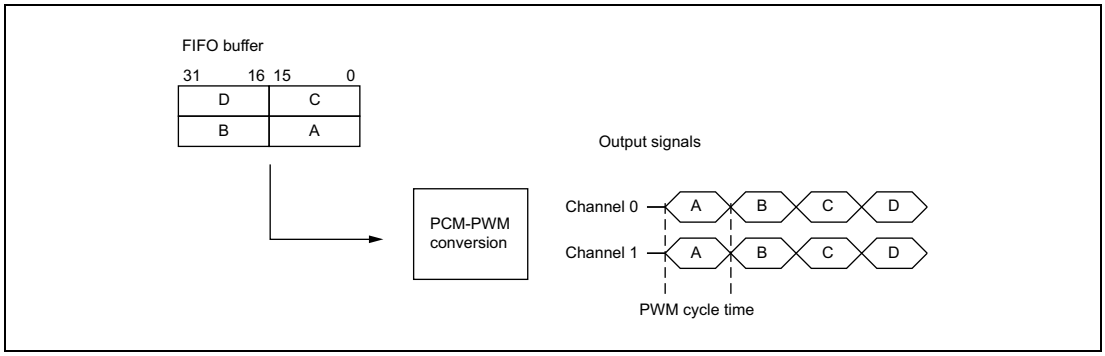


Figure 34.5 PCM-PWM in mono mode

34.3.2.2 Stereo mode

In stereo mode (PCMPnCTL.PCMPnSTEREO = 1), the PCM-PWM module processes a separate audio sample in each output channel.

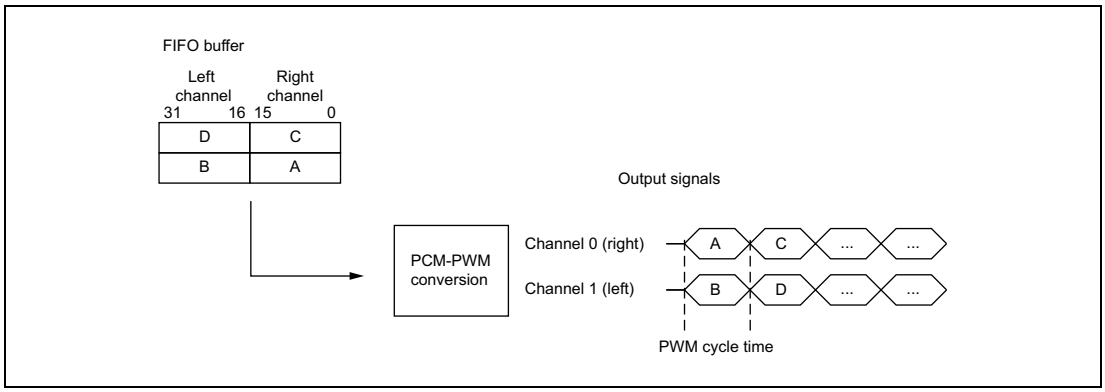


Figure 34.6 PCM-PWM in stereo mode

34.3.3 Output sampling frequency

The output sampling frequency depends on the settings in the control registers PCMPnTPWM (counter period) and PCMPnCKSEL (clock selection).

- The PCM-PWM module uses a 16-bit free running up-counter to determine the PWM duty cycle. The counter reset value is specified by PCMPnTPWM.
- The count clock frequency is determined by PCMPnCKSEL.

After conversion start (PCMPnCTL.PCMPnENAB = 1), the counter is started when the FIFO buffer is full (PCMPnSTR.PCMPnFIFO[1:0] = 2). This mechanism prevents the PCM-PWM module from trying to process audio samples until data is actually available.

The output sampling frequency and PWM cycle time are calculated as follows:

- Output sampling frequency = PCLK / (PCMPnTPWM x $2^{PCMPnCKSEL}$)
- PWM cycle time = (PCMPnTPWM x $2^{PCMPnCKSEL}$) / PCLK

34.3.3.1 Double output mode

The PCM-PWM module has a double output mode. When double output mode is enabled, every audio sample is processed twice. This doubles the output frequency with respect to the input frequency. The double output mode is enabled/disabled by the bit PCMPnCTL.PCMPnDOM.

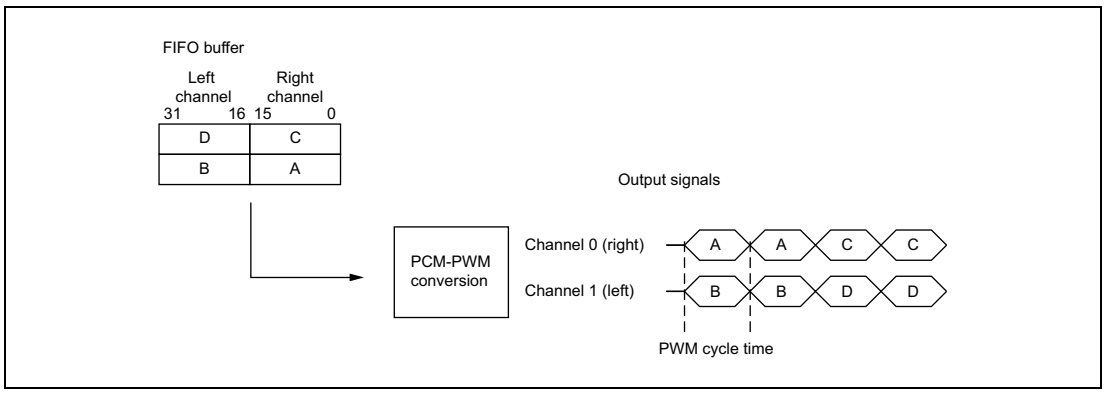


Figure 34.7 Double output mode (in stereo mode)

34.3.4 Output modes

34.3.4.1 Half H-bridge mode

In half H-bridge mode (PCMPnCTL.PCMPnMODE = 0), two pulse-width modulated signals (PCMPTPWMANx, PCMPTPWMAPx) are generated per output channel. The output signals can be used to drive a half H-bridge. Note that in this mode, the signals PCMPTPWMBPx and PCMPTPWMBNx are not used.

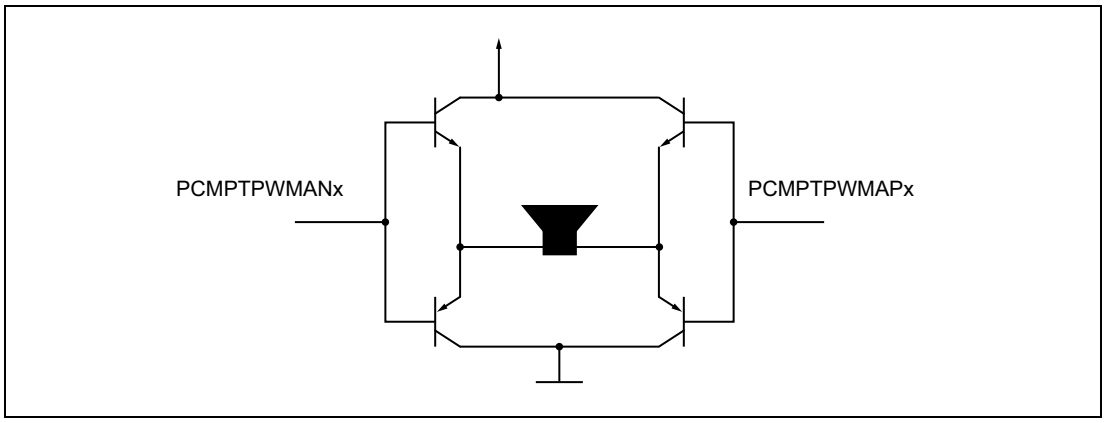


Figure 34.8 Driving a half H-bridge (here: emitter follower bridge)

The duty cycle of the PCMPTPWMAPx and PCMPTPWMANx signal reflects the sign of the PCM audio sample value. For example, if the duty cycle is above 50%, the audio sample has a positive value and if the duty cycle is below 50%, the audio sample has a negative value. Note that the active level can be set by control register PCMPnINV.

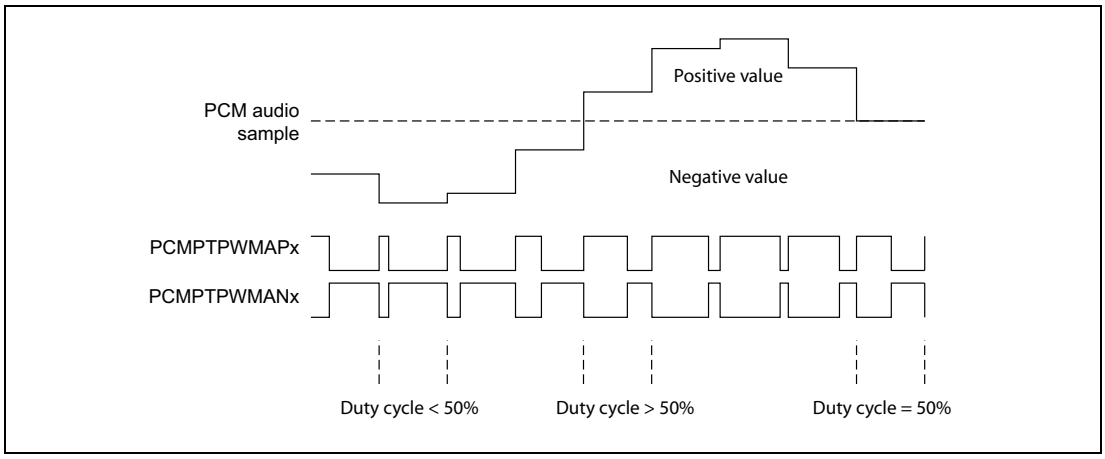


Figure 34.9 Example pulse-width modulated output in half H-bridge mode

NOTE

This mode can be used, for example, to drive a speaker that is driven by a single NPN transistor.

34.3.4.2 Full H-bridge mode

In full H-bridge mode, four pulse-width modulated signals are generated on each output channel (PCMPTPWMAPx, PCMPTPWMANx, PCMPTPWMBPx, PCMPTPWMBNx). These signals are used to drive a full H-bridge.

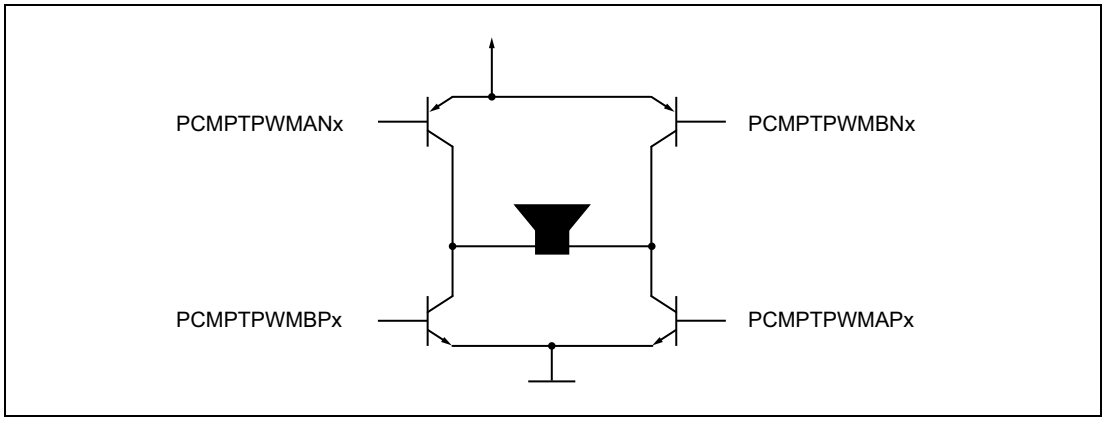


Figure 34.10 Driving a full H-bridge

In full H-bridge mode, the positive audio sample values are handled by the PCMPTPWMAPx / PCMPTPWMANx signals; the negative PCM audio sample values are handled by the PCMPTPWMBPx / PCMPTPWMBNx signals. Note that the active level can be set by control register PCMPnINV.

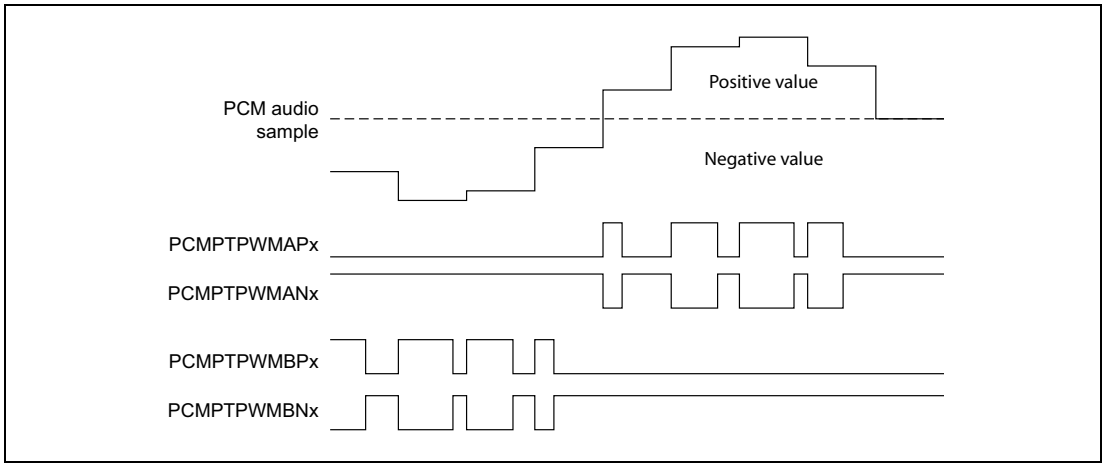


Figure 34.11 Example pulse-width modulated output in full H-bridge mode

NOTE

This output mode is used to drive a speaker with DC power-free operation during silence.

34.3.5 PCM-PWM output generators

The PCM-PWM output generators convert the PCM-coded audio samples to pulse-width modulated audio data as follows:

- Convert the signed integer value from the FIFO buffer to an unsigned integer value.
- Generate the corresponding PWM signals by comparing the resulting value with the current value of the internal counter.

The exact conversion algorithm depends on whether the module is in half H-bridge or full H-bridge mode.

34.3.5.1 PWM generation in half H-bridge mode

In half H-bridge mode, the PWM output generators add the value in the control register PCMPnOFFS to the signed integer value from the FIFO buffer. The resulting value is an unsigned integer.

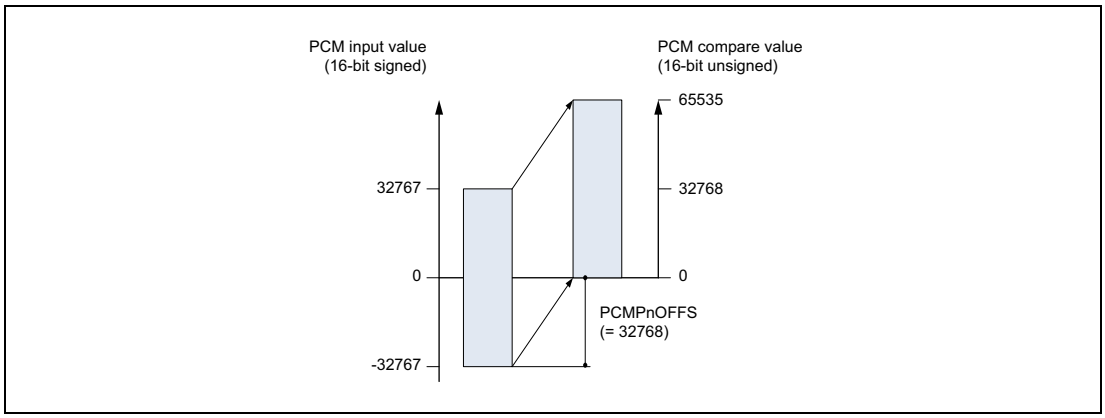


Figure 34.12 Converting the audio input in half H-bridge mode (PCMPnCTL.MODE = 0)

The following table shows the same mapping.

PCM input value	PCM compare value
-32768 (8000_H)	0
-1 ($FFFF_H$)	32767 ($7FFF_H$)
0	32768 (8000_H)
32767 ($7FFF_H$)	65535 ($FFFF_H$)

The resulting PCM compare values are then compared to the value of the counter. When the PCM compare value is greater than the value of the counter, the resulting PWM signal is high.

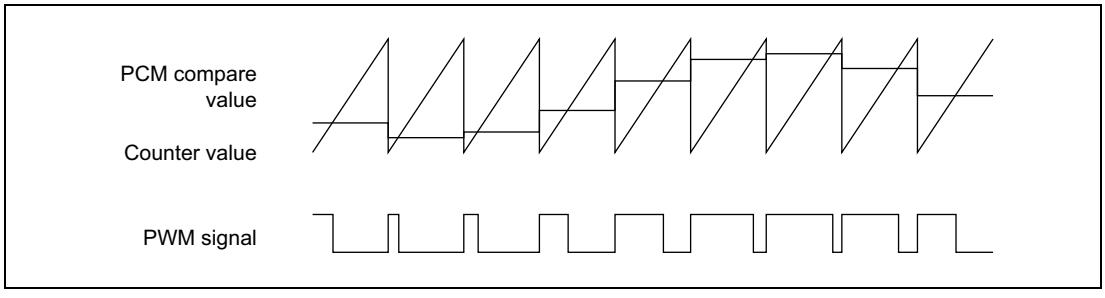


Figure 34.13 Converting the audio input in half H-bridge mode (PCMPnCTL.MODE = 0)

NOTES

1. The offset in PCMPnOFFS must be defined so that the lowest valid (negative) PCM value is converted to the unsigned integer value “0”.
2. The highest PCM input value must not exceed the maximum counter value specified in PCMPnTPWM.
3. The application writing the audio signals to the PCM-PWM must scale the values accordingly.

34.3.5.2 PWM generation in full H-bridge mode

In full H-bridge mode, two PWM signals are generated from the PCM input signals: one for the positive audio sample values (Ax) and one for the negative audio sample values (Bx). This is determined by the value of the most significant bit (MSB) in the PCM input value.

- If the MSB = 0, the PCM compare value equals the PCM input value. These values are used to generate the high level PWM signal.
- If the MSB = 1, the PCM compare value is calculated by subtracting 1 and inverting all bits. These values are used to generate the low level PWM signal.

PCM compare value = ~[(PCM input value) - 1]

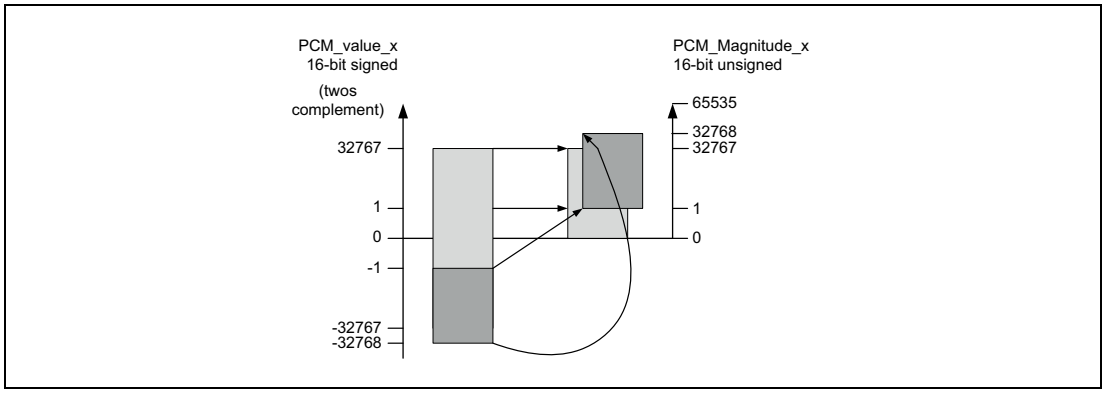


Figure 34.14 Converting the audio input in full H-bridge mode (PCMPnCTL.MODE = 1)

The following table shows the same mapping.

PCM input value	PCM compare value
-32768 (8000_H)	32768 (8000_H)
-1 ($FFFF_H$)	1 (0001_H)
0	0
1	1 (0001_H)
32767 ($7FFF_H$)	32767 ($7FFF_H$)

The resulting PCM compare values are then compared to the value of the counter. When the PCM compare value is greater than the value of the counter, the resulting PWM signal is high.

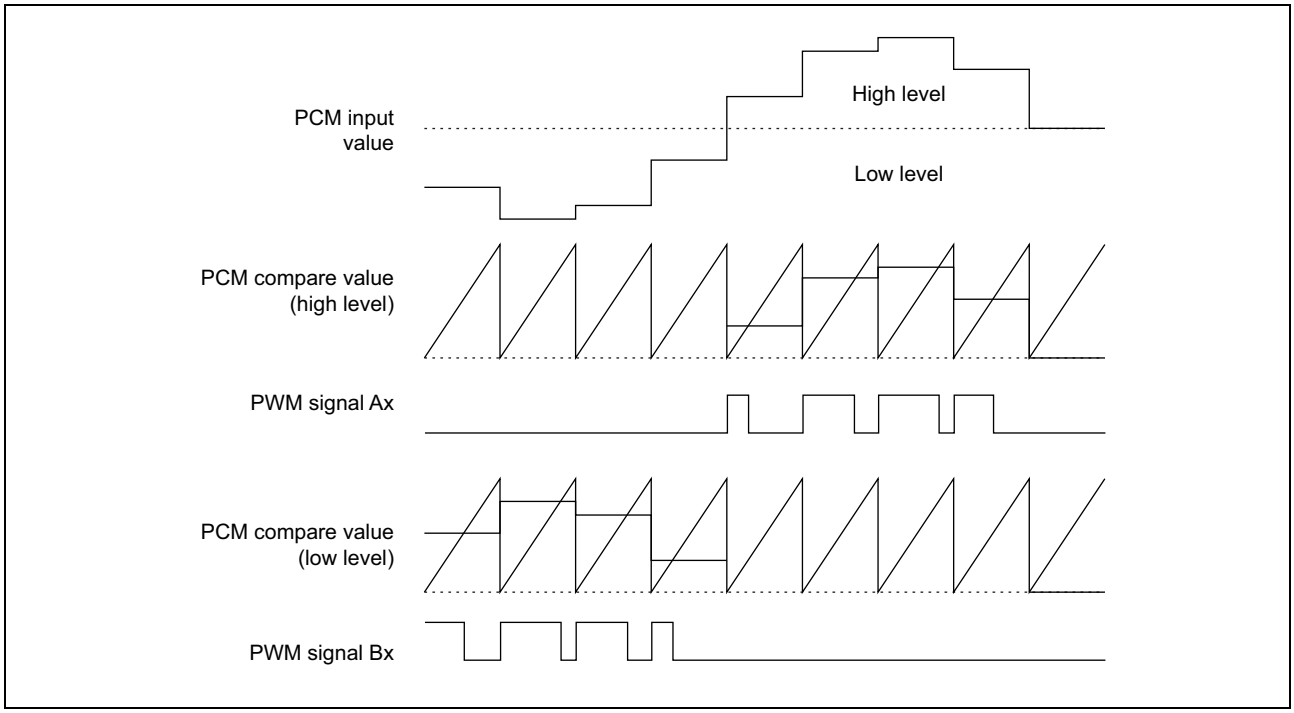


Figure 34.15 Calculating the PWM signals (full H-bridge mode)

34.3.6 Output control

The output control blocks receive the PWM signals from the PWM output generators and generate the PCMPTPWMAPx / PCMPTPWMANx and PCMPTPWMBPn / PCMPTPWMBNx output signals according to the following settings:

- Active level selection in the control register PCMPnINV
- Static signal output selection in the control registers PCMPnSTEN and PCMPnSTLV

Each of these registers allows individual settings for each of the output signals.

Active level selection

PCMPnINV defines whether the output signals are output non-inverted (active high) or inverted (active low). By default, the signals PCMPTPWMAPx and PCMPTPWMBPx are non-inverted, the signals PCMPTPWMANx and PCMPTPWMBNx are inverted.

Static signal output

PCMPnSTEN defines whether the PWM signal or a static signal is output. It is possible to change the selection during operation.

PCMPnSTLV and PCMPnINV define the level of the static signal. For details refer to **Table 34.7, Output control (registers interaction)**.

The following table provides an overview of how control registers PCMPnSTEN, PCMPnSTLV, and PCMPnINV interact.

Table 34.7 Output control (registers interaction)

PCMPnSTEN	PCMPnSTLV	PCMPnINV	Output	Description
0	X*1	0	PWM signal	PWM signal is output without changes.
		1	$\overline{\text{PWM signal}}$	PWM signal is inverted.
1	0	0	Low level	Static signal output Active: high, inactive: low Static level: inactive
		1	High level	Static signal output Active: low, inactive: high Static level: inactive
	1	0	High level	Static signal output Active: high, inactive: low Static level: active
		1	Low level	Static signal output Active: low, inactive: high Static level: active

Note 1. Control register PCMPnSTLV has no effect when control register PCMPnSTEN is 0.

NOTE

PCMPnSTEN and PCMPnSTLV can be used to perform a ramp up/down in half H-bridge mode, where one of the output pins is driven with the PWM signal and one is fixed. This avoids a crackling sound upon activation or deactivation of the PCM-PWM module.

34.3.7 Starting and stopping the PCM-PWM conversion

Start

To start PCM-PWM conversion, set PCMPnCTL.PCMPnENAB = 1. Audio samples can now be written to the control register PCMPnAUSA. The FIFO buffer picks up these samples.

When the FIFO buffer is full (PCMPnSTR.PCMPnFIFO[1:0] = 10_B), the PCM-PWM module starts converting the PCM data.

Stop

To stop PCM-PWM conversion, set PCMPnCTL.PCMPnENAB = 0. The conversion continues until the FIFO buffer is empty. After that, conversion is stopped. All output signals are set to active low level.

CAUTION

Since the PCM-PWM conversion continues until the FIFO buffer is empty, a final error interrupt PCMPTIFERR is generated to indicate a FIFO underrun, although PCMPnCTL.PCMPnENAB = 0.

34.3.8 Example setup

Example:

- Input data: 13 bit PCM audio samples
- Output mode: half H-bridge mode (emitter follower)
- Desired sample frequency: 11.025 kHz
- PCLK: 80 MHz
- Count clock: 80 MHz

Required settings:

- Count clock = PCLK / 2^0

  PCMPnCKSEL[2:0] = 000_B
- Maximum counter value = 80 MHz / 11.025 kHz = 7,256.24

  PCMPnTPWM[15:0] = 7,256 = $1C58_H$

This results in the following:

- Effective sample frequency f_{Seff}

  f_{Seff} = 80 MHz / 7,256 = 11.0254 kHz
- Effective output resolution R

  $7{,}256 = 2^R \Rightarrow R = \ln(7{,}256) / \ln(2) = 12.8$ bit
- Sample frequency error is < 0.05 %

34.4 PCM-PWM Registers

This section contains a description of all registers of the PCM-PWM module.

34.4.1 PCM-PWM registers overview

Table 34.8 PCM-PWM register overview

Register name	Shortcut	Address
Control register	PCMPnCTL	<PCMPn_base>
Counter period register	PCMPnTPWM	<PCMPn_base> + 04_H
Inverter control register	PCMPnINV	<PCMPn_base> + 08_H
Static level selection register	PCMPnSTLV	<PCMPn_base> + $0C_H$
Static level enable register	PCMPnSTEN	<PCMPn_base> + 10_H
Conversion offset register	PCMPnOFFS	<PCMPn_base> + 14_H
Audio sample input register	PCMPnAUSA	<PCMPn_base> + 18_H
Status register	PCMPnSTR	<PCMPn_base> + $1C_H$
Status clear register	PCMPnSTC	<PCMPn_base> + 20_H
Clock selection register	PCMPnCKSEL	<PCMPn_base> + 24_H
Emulation register	PCMPnEMU	<PCMPn_base> + 28_H

<PCMPn_base>

The base addresses <PCMPn_base> of the PCMPn is defined in the first section of this chapter under the key word “Register addresses”.

34.4.2 PCM-PWM registers details

34.4.2.1 PCMPnCTL - PCM-PWM control register

This register controls the operating mode and operating status of the PCM-PWM module.

Access: This register can be read/written in 32-bit units.

Address: <PCMPn_base>

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	PCMPnDOM	PCMPnSTEREO	PCMPnMODE	PCMPnENAB
R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.9 PCMPnCTL register contents

Bit position	Bit name	Function
31 to 4	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
3	PCMPn DOM	Enables/disables double output mode: 0: Disabled (every audio sample is output once) 1: Enabled (every audio sample is output twice)
2	PCMPn STEREO	Enables/disables stereo mode: 0: Disabled (mono mode) 1: Enabled (stereo mode)
1	PCMPn MODE	Specifies the output mode 0: Half H-bridge mode 1: Full H-bridge mode See Section 34.3.4, Output modes for additional information.
0	PCMPn ENAB	Enables/disables the PCM-PWM conversion: 0: Conversion disabled 1: Conversion enabled See Section 34.3.7, Starting and stopping the PCM-PWM conversion for details.

34.4.2.2 PCMPnTPWM - PCM-PWM counter period register

This register defines the period value of the counter. It influences the PWM cycle time and the output sample frequency as described in Section 34.3.3, Output sampling frequency.

Access: This register can be read/written in 32-bit units.
The PCM-PWM module must be disabled (PCMPnCTL.PCMPnENAB = 0) before writing to this register.

Address: <PCMPn_base> + 04_H

Initial Value: 0000 $FFFF_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCMPnTPWM[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.10 PCMPnTPWM register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
15 to 0	PCMPn TPWM[15:0]	Maximum counter value

34.4.2.3 PCMPnINV - PCM-PWM inverter control register

This register defines the active level of the eight PWM output signals.

Access: This register can be read/written in 32-bit units.
The PCM-PWM module must be disabled (PCMPnCTL.PCMPnENAB = 0) before writing to this register.

Address: <PCMPn_base> + 08_H

Initial Value: 0000 $0A0A_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	PCMPn INVBN1	PCMPn INVBP1	PCMPn INVAN1	PCMPn INVAP1	0	0	0	0	PCMPn INVBN0	PCMPn INVBP0	PCMPn INVAN0	PCMPn INVAP0
R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.11 PCMPnINV register contents

Bit position	Bit name	Function
31 to 12	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
11	PCMPn INVBN1	Sets the active level of the PCMPTPWMBN1 output signal: 0: Active high 1: Active low
10	PCMPn INVBP1	Sets the active level of the PCMPTPWMBP1 output signal: 0: Active high 1: Active low
9	PCMPn INVAN1	Sets the active level of the PCMPTPWMAN1 output signal: 0: Active high 1: Active low
8	PCMPn INVAP1	Sets the active level of the PCMPTPWMAP1 output signal: 0: Active high 1: Active low
7 to 4	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
3	PCMPn INVBN0	Sets the active level of the PCMPTPWMBN0 output signal: 0: Active high 1: Active low
2	PCMPn INVBP0	Sets the active level of the PCMPTPWMBP0 output signal: 0: Active high 1: Active low
1	PCMPn INVAN0	Sets the active level of the PCMPTPWMAN0 output signal: 0: Active high 1: Active low
0	PCMPn INVAP0	Sets the active level of the PCMPTPWMAP0 output signal: 0: Active high 1: Active low

34.4.2.4 PCMPnSTLV - PCM-PWM static level selection register

This register defines the static output level for every output pin if the static signal output function is enabled (corresponding bit in PCMPnSTEN is set). The static output level is also influenced by the setting of PCMPnINV. For details refer to Section 34.3.6, Output control.

Access: This register can be read/written in 32-bit units.
The PCM-PWM module must be disabled (PCMPnCTL.PCMPnENAB = 0) before writing to this register.

Address: <PCMPn_base> + $0C_H$

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	PCMPNSTLVBN1	PCMPnSTLVBP1	PCMPNSTLVAN1	PCMPnSTLVAP1	0	0	0	0	PCMPnSTLVBN0	PCMPnSTLVBP0	PCMPnSTLVAN0	PCMPnSTLVAP0
R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.12 PCMPnSTLV register contents

Bit position	Bit name	Function
31 to 12	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
11	PCMPn STLVBN1	Sets the static level of the PCMPTPWMBN1 output signal: 0: Static inactive 1: Static active
10	PCMPn STLVBP1	Sets the static level of the PCMPTPWMBP1 output signal: 0: Static inactive 1: Static active
9	PCMPn STLVAN1	Sets the static level of the PCMPTPWMAN1 output signal: 0: Static inactive 1: Static active
8	PCMPn STLVAP1	Sets the static level of the PCMPTPWMAP1 output signal: 0: Static inactive 1: Static active
7 to 4	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
3	PCMPn STLVBN0	Sets the static level of the PCMPTPWMBN0 output signal: 0: Static inactive 1: Static active
2	PCMPn STLVBP0	Sets the static level of the PCMPTPWMBP0 output signal: 0: Static inactive 1: Static active
1	PCMPn STLVAN0	Sets the static level of the PCMPTPWMAN0 output signal: 0: Static inactive 1: Static active
0	PCMPn STLVAP0	Sets the static level of the PCMPTPWMAP0 output signal: 0: Static inactive 1: Static active

34.4.2.5 PCMPnSTEN - PCM-PWM static level enable register

This register enables/disables the static signal output function for every output signal. Refer to Section 34.3.6, Output control for additional information.

Access: This register can be read/written in 32-bit units.

Address: <PCMPn_base> + 10_H

Initial Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	PCMPn STEN BN1	PCMPn STEN BP1	PCMPn STEN AN1	PCMPn STEN AP1	0	0	0	0	PCMPn STEN BN0	PCMPn STEN BP0	PCMPn STEN AN0	PCMPn STEN AP0
R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 34.13 PCMPnSTEN register contents

Bit position	Bit name	Function
31 to 12	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
11	PCMPn STENBN1	Enables/disables static output for the PCMPTPWMBN1 output signal: 0: PWM signal output enabled 1: Static output enabled
10	PCMPn STENBP1	Enables/disables static output for the PCMPTPWMBP1 output signal: 0: PWM signal output enabled 1: Static output enabled
9	PCMPn STENAN1	Enables/disables static output for the PCMPTPWMAN1 output signal: 0: PWM signal output enabled 1: Static output enabled
8	PCMPn STENAP1	Enables/disables static output for the PCMPTPWMAP1 output signal: 0: PWM signal output enabled 1: Static output enabled
7 to 4	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
3	PCMPn STENBN0	Enables/disables static output for the PCMPTPWMBN0 output signal: 0: PWM signal output enabled 1: Static output enabled
2	PCMPn STENBP0	Enables/disables static output for the PCMPTPWMBP0 output signal: 0: PWM signal output enabled 1: Static output enabled
1	PCMPn STENAN0	Enables/disables static output for the PCMPTPWMAN0 output signal: 0: PWM signal output enabled 1: Static output enabled
0	PCMPn STENAP0	Enables/disables static output for the PCMPTPWMAP0 output signal: 0: PWM signal output enabled 1: Static output enabled

34.4.2.6 PCMPnOFFS - PCM-PWM conversion offset register

This register stores the offset that is used for the signed-unsigned conversion in half H-bridge mode (see Section 34.3.5.1, PWM generation in half H-bridge mode).

Access: This register can be read/written in 32-bit units.
The PCM-PWM module must be disabled (PCMPnCTL.PCMPnENAB = 0) before writing to this register.

Address: <PCMPn_base> + 14_H

Initial Value: $0000\ 8000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCMPnOFFS[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34.14 PCMPnOFFS register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
15 to 0	PCMPn OFFS[15:0]	Offset for conversion in half H-bridge mode Refer to Section 34.3.5.1, PWM generation in half H-bridge mode for details.

34.4.2.7 PCMPnAUSA - PCM-PWM audio sample input register

This register is used to write a 32-bit pattern (as part of a sample) to the FIFO buffer. This 32-bit pattern has two 16-bit patterns. These patterns will either be used as left/right audio samples in stereo mode or as two consecutive mono samples in mono mode. The format of the sample is as a 16-bit signed integer in two's complement representation.

Access: This register can be written in 32-bit units.

Address: <PCMPn_base> + 18_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCMPnAUSL[15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCMPnAUSR[15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 34.15 PCMPnAUSA register contents

Bit position	Bit name	Function
31 to 16	PCMPn AUSL[15:0]	16-bit sample left
15 to 0	PCMPn AUSR[15:0]	16-bit sample right

NOTES

1. The PCM-encoded data must be scaled to this value by the software sending audio data. If, for example, PCMPnTPWM is set to 255, the maximum compare value derived from the PCM-encoded input data has to be 255. The PCM-PWM module does not scale the data. All compare values that exceed the maximum counter value result in a 100% duty cycle. For details on the generation of compare values, refer to **Section 34.3.5, PCM-PWM output generators**.
2. If audio samples with less than 16 bits are used, they have to be sign-extended to 16 bits by the application to ensure correct interpretation of negative values.

34.4.2.8 PCMPnSTR - PCM-PWM status register

This register is used to monitor the fill state of the FIFO buffer (if the FIFO buffer has data to be converted), occurrences of a FIFO buffer overflow and a FIFO buffer underrun.

Access: This register can be read in 32-bit units.

Address: <PCMPn_base> + $1C_H$

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCMPn FIFO[1:0]		0	0	0	0	0	0	0	0	0	0	0	0	PCMPn OV	PCMPn UR
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 34.16 PCMPnSTR register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
15 to 14	PCMPn FIFO[1:0]	Indicates the FIFO buffer fill state:

PCMPn FIFO[1:0]	Function
00_B	FIFO buffer empty.
01_B	FIFO buffer has one entry.
10_B	FIFO buffer has two entries.
11_B	Not used.

Bit position	Bit name	Function
13 to 2	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
1	PCMPn OV	FIFO buffer overflow flag: 0: No overflow occurred 1: Overflow occurred
0	PCMPn UR	FIFO buffer underrun flag: 0: No underrun occurred 1: Underrun occurred

34.4.2.9 PCMPnSTC - PCM-PWM status clear register

This register is used to reset the FIFO buffer status flags.

Access: This register can be written in 32-bit units.

Address: <PCMPn_base> + 20_H

Initial Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	PCMPn OVC	PCMPn URC
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 34.17 PCMPnSTC register contents

Bit position	Bit name	Function
31 to 2	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
1	PCMPn OVC	Clear FIFO buffer overflow flag: 0: No action 1: Clear PCMPnSTR.PCMPnOV
0	PCMPn URC	Clear FIFO buffer underrun flag: 0: No action 1: Clear PCMPnSTR.PCMPnUR

34.4.2.10 PCMPnCKSEL - PCM-PWM clock selection register

This register specifies the count clock.

Access: This register can be read/written in 32-bit units.
The PCM-PWM module must be disabled (PCMPnCTL.PCMPnENAB = 0) before writing to this register.

Address: <PCMPn_base> + 24_H

Initial Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	PCMPnCKSEL[2:0]		
R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 34.18 PCMPnCKSEL register contents

Bit position	Bit name	Function
31 to 3	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
2 to 0	PCMPn CKSEL[2:0]	Specifies the count clock:

PCMPn CKSEL[2:0]	Function
000_B	PCLK / 2^0
001_B	PCLK / 2^1
010_B	PCLK / 2^2
...	...
111_B	PCLK / 2^7

34.4.2.11 PCMPnEMU - PCM-PWM emulation register

This register controls whether the PCM-PWM can be stopped during emulation, for instance upon a breakpoint hit.

Access: This register can be read/written in 32-bit units.

Address: <PCMPn_base> + 28_H

Initial Value: 00_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	PCMPn SVSDIS	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R

Table 34.19 PCMPnEMU register contents

Bit position	Bit name	Function
31 to 8	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.
7	PCMPn SVSDIS	Emulation control: 0: PCM-PWM can be stopped during emulation 1: PCM-PWM operates continuously during emulation
6 to 0	Reserved	When read, the value after reset is returned. When written to these bits, write the value after reset.

Section 35 Serial Sound Interface (SSIF)

This section contains a generic description of the Serial Sound Interface.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

35.1 Overview of the RH850/D1L/D1M Serial Sound Interfaces

35.1.1 Units

This microcontroller has the following number of units of the Serial Sound Interface.

Table 35.1 Number of Units

Product Name	All products
Units	2
Names	SSIFn (n = 0, 1)

Table 35.2 Index

Index	Meaning
n	Throughout this section, the individual SSIF units are identified by the index “n” (n = 0, 1).

35.1.2 Register addresses

All Serial Sound Interface register addresses are given as address offsets from the individual base addresses <SSIFn_base>.
The <SSIFn_base> addresses of each SSIFn are listed in the following table:

Table 35.3 Register base addresses <SSIFn_base>

SSIFn unit	<SSIFn_base> address
SSIF0	FFF1 3000_H
SSIF1	FFF1 4000_H

35.1.3 Clock supply

All Serial Sound Interfaces provide following clock inputs.

Table 35.4 Clock supply

SSIFn unit	SSIFn clock	Connected to
SSIF0 SSIF1	PCLK	Clock Controller C_ISO_PCLK
	SSIACKINT	Clock Controller C_ISO_SSIF

35.1.4 Interrupt and DMA requests

The Serial Sound Interfaces can generate the following interrupt and DMA requests:

Table 35.5 SSIFn interrupt and DMA requests

SSIFn signals	Function	Connected to
SSIF0:		
TXI	Transmission data empty	Interrupt Controller INTSSIF0TX DMA trigger ID 113
RXI	Reception data full	Interrupt Controller INTSSIF0RX DMA trigger ID 114
IRQ	Multi-purpose interrupt	INTSSIF0
SSIF1:		
TXI	Transmission data empty	Interrupt Controller INTSSIF1TX DMA trigger ID 115
RXI	Reception data full	Interrupt Controller INTSSIF1RX DMA trigger ID 116
IRQ	Multi-purpose interrupt	INTSSIF1

35.1.5 Reset sources

The Serial Sound Interfaces and their registers are initialized by the following reset signal:

Table 35.6 Reset sources

SSIFn unit	Reset signal
SSIFn	• Reset Controller SYSRES • Reset Controller SG3RES for SSIF1 • Reset Controller SG2RES for SSIF0 • Reset upon wake-up from DEEPSTOP mode

CAUTION

By default the SGnRES resets are active.

Thus before accessing an SGn module this reset must be released via the MRSTC register of the Reset Controller. Refer to Section 9, Reset Controller for details.

35.1.6 I/O signals

The following table shows the I/O signals of the Serial Sound Interfaces.

Table 35.7 I/O signals connections

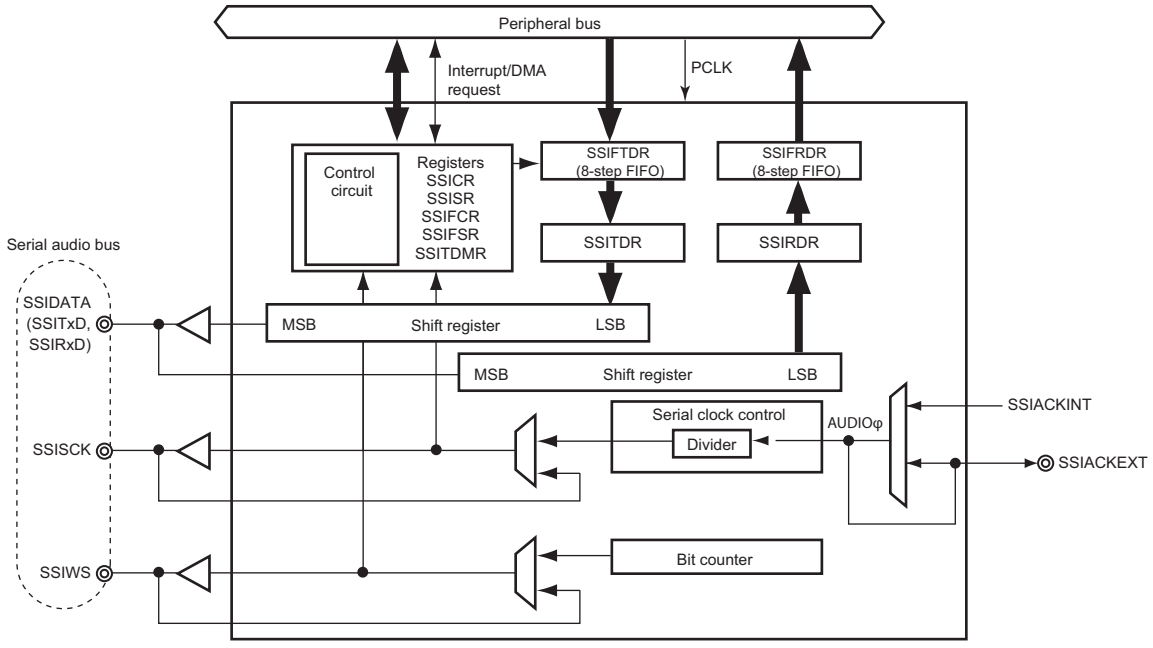
SSIFn signal	Function	Connected to
SSIACKEXT	Common audio clock input/output	Port SSIFACK
SSIF0:		
SSIDATA	Serial data input	Port SSIF0RXD
	Serial data output	Port SSIF0TXD
SSISCK	Serial bit clock input/output	Port SSIF0SCK
SSIWS	Word select input/output	Port SSIF0WS
SSIF1:		
SSIDATA	Serial data input	Port SSIF1RXD
	Serial data output	Port SSIF1TXD
SSISCK	Serial bit clock input/output	Port SSIF1SCK
SSIWS	Word select input/output	Port SSIF1WS

35.2 Features

The Serial Sound Interface is a module designed to exchange audio data with various devices offering I^2S bus compatibility. It also provides additional modes for other common formats, as well as support for multi-channel mode.

- Number of channels: Two channels
- Operating mode: Non-compressed mode
  The non-compressed mode supports serial audio streams divided by channels.
- Serves as both a transmitter and a receiver
- Capable of using serial bus format
- Asynchronous transfer takes place between the data buffer and the shift register.
- It is possible to select a dividing ratio for the clock used by the serial bus interface.
- It is possible to control data transmission or reception with DMA transfer and interrupt requests.
- Selects the oversampling clock input from among the following pins:
  - internal SSIACKINT
  - external SSIACKEXT
- Includes 8-stage FIFO buffers in transmitter and receiver
- Supports multi-channel mode (TDM mode) in which the SSIWS signal is high only for system word 1 period.
- Supports WS continue mode in which the SSIWS signal is not stopped.

The figure below shows a block diagram of this module.



[Legend]

SSICR:	Control register	SSIFCR:	FIFO control register
SSISR:	Status register	SSIFSR:	FIFO status register
SSITDR:	Transmit data register	SSIFTDR:	Transmit FIFO data register
SSIRDR:	Receive data register	SSIFRDR:	Receive FIFO data register
SSITDMR:	TDM mode register		

Note:
The selectable oversampling clocks SSIACKINT and SSIACKEXT are commonly used by both channels. All signals of the serial audio bus (SSIDATA (SSITxD, SSIRxD), SSISCK, SSIWS) are separated for each channel.

Figure 35.1 Block Diagram of the Serial Sound Interface

35.3 Register Description

The table below lists the register configuration. Note that explanations in the text omit the channel numbers.

Table 35.8 Register Configuration

Module Name	Channel	Register Name	Symbol	R/W	Initial Value	Address	Access Size
SSIF	0	Control register 0	SSICR_0	R/W	H'00000000	<SSIF0_base> + 00_H	32
SSIF		Status register 0	SSISR_0	R/W*1	H'02000013	<SSIF0_base> + 04_H	32
SSIF		FIFO control register 0	SSIFCR_0	R/W	H'00000000	<SSIF0_base> + 10_H	32
SSIF		FIFO status register 0	SSIFSR_0	R/(W)*2	H'00010000	<SSIF0_base> + 14_H	32
SSIF		Transmit FIFO data register 0	SSIFTDR_0	W	Undefined	<SSIF0_base> + 18_H	32
SSIF		Receive FIFO data register 0	SSIFRDR_0	R	Undefined	<SSIF0_base> + $1C_H$	32
SSIF		TDM mode register 0	SSITDMR_0	R/W	H'00000000	<SSIF0_base> + 20_H	32
SSIF	1	Control register 1	SSICR_1	R/W	H'00000000	<SSIF1_base> + 00_H	32
SSIF		Status register 1	SSISR_1	R/W*1	H'02000013	<SSIF1_base> + 04_H	32
SSIF		FIFO control register 1	SSIFCR_1	R/W	H'00000000	<SSIF1_base> + 10_H	32
SSIF		FIFO status register 1	SSIFSR_1	R/(W)*2	H'00010000	<SSIF1_base> + 14_H	32
SSIF		Transmit FIFO data register 1	SSIFTDR_1	W	Undefined	<SSIF1_base> + 18_H	32
SSIF		Receive FIFO data register 1	SSIFRDR_1	R	Undefined	<SSIF1_base> + $1C_H$	32
SSIF		TDM mode register 1	SSITDMR_1	R/W	H'00000000	<SSIF1_base> + 20_H	32
SELB	0 and 1	SSIF clock setting register	SSIFCLKCFG	R/W	H'00000000	FFC0 $602C_H$	32

Note 1. Although bits 29 to 26 in these registers can be read from or written to, bits other than these are read-only. For details, refer to Section 35.3.2, Status Register (SSISR).

Note 2. To bits 16 and 0 in these registers, only 0 can be written to clear the flags. Other bits are read-only. For details, refer to Section 35.3.6, FIFO Status Register (SSIFSR).

NOTES

1. In the header files the names of the above registers are defined in the following format:
   <ModuleName> + <Symbol>.
   <ModuleName> and <Symbol> are defined in the above table.
2. SSIF channel number is placed after "SSIF", and also needs "_0" "_1" at the end (i.e. SSIF0SSICR_0, SSIF1SSICR_1).
   Only SSIFCLKCFG has no channel number. (i.e. SELBSSIFCLKCFG).

35.3.1 Control Register (SSICR)

SSICR is a 32-bit readable/writable register that controls the IRQ, selects the polarity status, and sets operating mode.

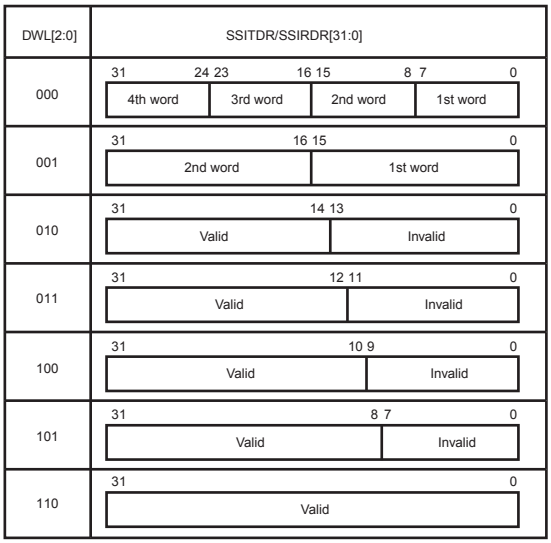
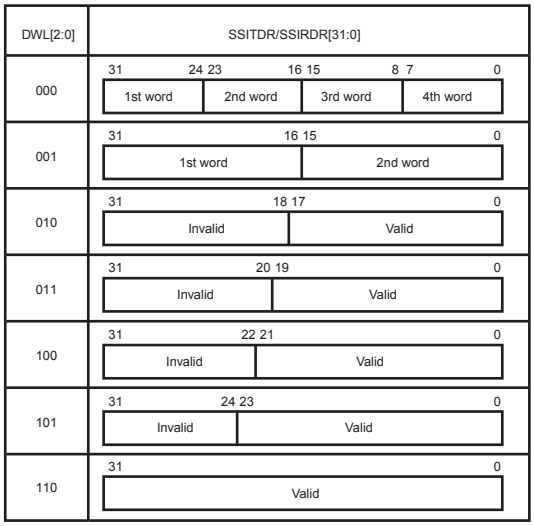
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	TUIEN	TOIEN	RUIEN	ROIEN	IIEN	-	CHNL[1:0]		DWL[2:0]			SWL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SCKD	SWSD	SCKP	SWSP	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	-	TEN	REN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	0	R	Reserved The read value is undefined. The write value should always be 0.
29	TUIEN	0	R/W	Transmit Underflow Interrupt Enable 0: Disables an underflow interrupt. 1: Enables an underflow interrupt.
28	TOIEN	0	R/W	Transmit Overflow Interrupt Enable 0: Disables an overflow interrupt. 1: Enables an overflow interrupt.
27	RUIEN	0	R/W	Receive Underflow Interrupt Enable 0: Disables an underflow interrupt. 1: Enables an underflow interrupt.
26	ROIEN	0	R/W	Receive Overflow Interrupt Enable 0: Disables an overflow interrupt. 1: Enables an overflow interrupt.
25	IIEN	0	R/W	Idle Mode Interrupt Enable 0: Disables an idle mode interrupt. 1: Enables an idle mode interrupt.
24	—	0	R	Reserved The read value is undefined. The write value should always be 0.
23, 22	CHNL[1:0]	00	R/W	Channels [When TDM = 0] These bits show the number of channels in each system word. 00: Having one channel per system word 01: Having two channels per system word 10: Having three channels per system word 11: Having four channels per system word [When TDM = 1] These bits show the number of system words in each TDM frame. 00: Setting prohibited 01: Having four system words per TDM frame 10: Having six system words per TDM frame 11: Having eight system words per TDM frame
21 to 19	DWL[2:0]	000	R/W	Data Word Length These bits indicate the number of bits in a data word. 000: 8 bits 001: 16 bits 010: 18 bits 011: 20 bits 100: 22 bits 101: 24 bits 110: 32 bits 111: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
18 to 16	SWL[2:0]	000	R/W	System Word Length These bits indicate the number of bits in a system word. 000: 8 bits 001: 16 bits 010: 24 bits 011: 32 bits 100: 48 bits 101: 64 bits 110: 128 bits 111: 256 bits
15	SCKD	0	R/W	Serial Bit Clock Direction 0: Serial bit clock is input, slave mode. 1: Serial bit clock is output, master mode. Note: Only the following settings are allowed: (SCKD, SWSD) = (0, 0) or (1, 1). Other settings are prohibited.
14	SWSD	0	R/W	Serial WS Direction 0: Serial word select is input, slave mode. 1: Serial word select is output, master mode. Note: Only the following settings are allowed: (SCKD, SWSD) = (0,0) or (1,1). Other settings are prohibited.
13	SCKP	0	R/W	Serial Bit Clock Polarity 0: SSIWS and SSIDATA change at the SSISCK falling edge (sampled at the SSISCK rising edge). 1: SSIWS and SSIDATA change at the SSISCK rising edge (sampled at the SSISCK falling edge).
12	SWSP	0	R/W	Serial WS Polarity [When TDM = 0] 0: SSIWS is low for the 1st channel, high for the 2nd channel. 1: SSIWS is high for the 1st channel, low for the 2nd channel. [When TDM = 1] 0: SSIWS is high only for system word 1 period, low for other periods. 1: Setting prohibited
11	SPDP	0	R/W	Serial Padding Polarity This bit is used to specify the active level of padding bits. 0: Padding bits are low. 1: Padding bits are high.
10	SDTA	0	R/W	Serial Data Alignment 0: Transmitting and receiving in the order of serial data and padding bits 1: Transmitting and receiving in the order of padding bits and serial data

	SCKP =0	SCKP = 1
SSIDATA input sampling timing at the time of reception	SSISCK rising edge	SSISCK falling edge
SSIDATA output change timing at the time of transmission	SSISCK falling edge	SSISCK rising edge
SSIWS input sampling timing at the time of slave mode (SWSD = 0)	SSISCK rising edge	SSISCK falling edge
SSIWS output change timing at the time of master mode (SWSD = 1)	SSISCK falling edge	SSISCK rising edge

Bit	Bit Name	Initial Value	R/W	Description
9	PDTA	0	R/W	Parallel Data Alignment When the data word length is 32 bits, this configuration field has no meaning. This bit applies to SSIRDR in receive mode and SSITDR in transmit mode. When data word length is 8 or 16 bits: 0: The lower bits of parallel data (SSITDR, SSIRDR) are transferred prior to the upper bits. 1: The upper bits of parallel data (SSITDR, SSIRDR) are transferred prior to the lower bits. When data word length is 18, 20, 22, or 24 bits: 0: Parallel data (SSITDR, SSIRDR) is left-aligned. 1: Parallel data (SSITDR, SSIRDR) is right-aligned. PDTA = 0  PDTA = 1 
8	DEL	0	R/W	Serial Data Delay 0: 1 clock cycle delay between SSIWS and SSIDATA 1: No delay between SSIWS and SSIDATA

Bit	Bit Name	Initial Value	R/W	Description
7 to 4	CKDV[3:0]	0000	R/W	Serial Oversampling Clock Division Ratio Sets the ratio between the oversampling clock (AUDIOφ) and the serial bit clock. When the SCKD bit is 0, the setting of these bits is ignored. The serial bit clock is used in the shift register and is supplied from the SSISCK pin. 0000: AUDIOφ*1 0001: AUDIOφ/2 0010: AUDIOφ/4 0011: AUDIOφ/8 0100: AUDIOφ/16 0101: AUDIOφ/32 0110: AUDIOφ/64 0111: AUDIOφ/128 1000: AUDIOφ/6 1001: AUDIOφ/12 1010: AUDIOφ/24 1011: AUDIOφ/48 1100: AUDIOφ/96 1101: Setting prohibited 1110: Setting prohibited 1111: Setting prohibited
3	MUEN	0	R/W	Mute Enable This bit is used to specify muting of the signals from this module during transmission. 0: This module is not muted. 1: This module is muted. Note: When this module is muted, serial data to be output is rewritten to 0 but data transmission in the module is not stopped. Write dummy data to the SSIFTDR not to generate a transmit underflow because the number of data in the transmit FIFO is decreasing.
2	–	0	R	Reserved The read value is undefined. The write value should always be 0.
1	TEN	0	R/W	Transmit Enable 0: Disables the transmit operation. 1: Enables the transmit operation. **Note:** Note: When TEN is set to 0 during transmission, SSIF must be reset for restart by SW reset (SG2RES or SG3RES).
0	REN	0	R/W	Receive Enable 0: Disables the receive operation. 1: Enables the receive operation.

Note 1. If CKDV[3:0] = 0000 (AUDIOφ) is selected, please make sure the I^2S clock duty requirement (< 5 %, i.e. duty cycle from 45 % to 55 %) is kept for external clocks . Also check the influence of activated digital noise filter.

35.3.2 Status Register (SSISR)

SSISR consists of status flags indicating the operational status of this module and bits indicating the current channel numbers and word numbers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	TUIRQ	TOIRQ	RUIRQ	ROIRQ	IIRQ	-	-	-	-	-	-	-	-	-
Initial value:	Undefined	Undefined	0	0	0	0	1	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W:	R	R	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	TCHNO[1:0]		TSWNO	RCHNO[1:0]		RSWNO	IDST
Initial value:	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	0	0	1	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note: * The bit can be read or written to. Writing 0 initializes the bit, but writing 1 is ignored.

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
29	TUIRQ	0	R/(W)*	Transmit Underflow Error Interrupt Status Flag This status flag indicates that transmit data was supplied at a lower rate than was required. This bit is set to 1 regardless of the value of the TUIEN bit and can be cleared by writing 0 to this bit. If TUIRQ = 1 and TUIEN = 1, an IRQ interrupt occurs. If TUIRQ = 1, SSITDR did not have data written to it before it was required for transmission. This will lead to the same data being transmitted once more and a potential corruption of multi-channel data. As a result, this module will output erroneous data. Note: When an underflow error occurs, the current data in the data buffer of this module is transmitted until the next data is written.
28	TOIRQ	0	R/(W)*	Transmit Overflow Error Interrupt Status Flag This status flag indicates that transmit data was supplied at a higher rate than was required. This bit is set to 1 regardless of the value of the TOIEN bit and can be cleared by writing 0 to this bit. If TOIRQ = 1 and TOIEN = 1, an SSIF interrupt occurs. If TOIRQ = 1, SSIFTDR had data written to it while the transmit FIFO is full (TDC[3:0] = H'8). This will lead to the loss of data and a potential corruption of multi-channel data.
27	RUIRQ	0	R/(W)*	Receive Underflow Error Interrupt Status Flag This status flag indicates that receive data was supplied at a lower rate than was required. This bit is set to 1 regardless of the value of the RUIEN bit and can be cleared by writing 0 to this bit. If RUIRQ = 1 and RUIEN = 1, an SSIF interrupt occurs. If RUIRQ = 1, SSIFRDR was read while the receive FIFO is empty (RDC[3:0] = H'0).This can cause invalid receive data to be stored, which may lead to corruption of multi-channel data.
26	ROIRQ	0	R/(W)*	Receive Overflow Error Interrupt Status Flag This status flag indicates that receive data was supplied at a higher rate than was required. This bit is set to 1 regardless of the value of the ROIEN bit and can be cleared by writing 0 to this bit. If ROIRQ = 1 and ROIEN = 1, an SSIF interrupt occurs. If ROIRQ = 1, SSIRDR was not read before there was new unread data written to it. This will lead to the loss of data and a potential corruption of multi-channel data. Note: When an overflow error occurs, the current data in the data buffer of this module is overwritten by the next incoming data from the SSIF interface.

Bit	Bit Name	Initial Value	R/W	Description
25	IIRQ	1	R	Idle Mode Interrupt Status Flag This status flag indicates whether this module is in the idle state. This bit is set to 1 regardless of the value of the IIEN bit to allow polling. The interrupt can be masked by clearing IIEN to 0, but cannot be cleared by writing 0 to this bit. If IIRQ = 1 and IIEN = 1, an SSIF interrupt occurs. 0: This module is not in idle state. 1: This module is in idle state.
24 to 7	—	Undefined	R	Reserved The read value is undefined. The write value should always be 0.
6, 5	TCHNO [1:0]	00	R	Transmit Channel Number These bits show the current channel number. These bits indicate the number of a channel whose data is required to be written to SSITDR. This value will change as the data is copied to the shift register, regardless of whether the data is written to SSITDR. When TDM or CONT is 1, these bits cannot be used.
4	TSWNO	1	R	Transmit Serial Word Number This status bit indicates the current word number. This bit indicates which system word is required to be written to SSITDR. This value will change as the data is copied to the shift register, regardless of whether the data is written to SSITDR. When TDM or CONT is 1, this bit cannot be used.
3, 2	RCHNO [1:0]	00	R	Receive Channel Number These bits show the current channel number. These bits indicate which channel the data in SSIRDR currently represents. This value will change as the data in SSIRDR is updated from the shift register. When TDM or CONT is 1, these bits cannot be used.
1	RSWNO	1	R	Receive Serial Word Number This status bit indicates the current word number. This bit indicates which system word the data in SSIRDR currently represents. This value will change as the data in SSIRDR is updated from the shift register, regardless of whether SSIRDR has been read. When TDM or CONT is 1, this bit cannot be used.
0	IDST	1	R	Idle Mode Status Flag This status flag indicates that the serial bus activity has stopped. This bit is cleared to 0 if the serial bus is currently active while TEN = 1 or REN = 1. This bit is automatically set to 1 if both TEN and REN are cleared to 0 and the current system word communication is completed. Note: If the external device stops the serial bus clock before the current system word is completed, this bit is not set.

Note: *The bit can be read or written to. Writing 0 initializes the bit, but writing 1 is ignored.

35.3.3 Transmit Data Register (SSITDR)

SSITDR is a 32-bit register that stores data to be transmitted. The data for transmission in the transmit FIFO data register (SSIFTDR) is automatically transferred to SSITDR.

Data written to this register is transferred to the shift register upon transmission request. If the data word length is less than 32 bits, the alignment is determined by the setting of the PDTA control bit in SSICR.

The CPU cannot read or write data from/to SSITDR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

35.3.4 Receive Data Register (SSIRDR)

SSIRDR is a 32-bit register that stores received data. The received data stored in SSIRDR is automatically transferred to the receive FIFO data register (SSIFRDR).

Data in this register is transferred from the shift register each time data word is received. If the data word length is less than 32 bits, the alignment is determined by the setting of the PDTA control bit in SSICR.

The CPU cannot read or write data from/to SSIRDR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

35.3.5 FIFO Control Register (SSIFCR)

SSIFCR is a 32-bit readable/writable register that specifies the data trigger numbers for the transmit and receive FIFO data registers, and enables or disables FIFO data resets and interrupt requests.

SSIFCR can always be read or written by the CPU.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	TTRG[1:0]		RTRG[1:0]		TIE	RIE	TFRST	RFRST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7, 6	TTRG[1:0]	00	R/W	Transmit Data Trigger Number When the FIFO is operating for transmission, these bits specify the number of bytes for transmission in the FIFO (trigger number for transmission) at which the TDE flag in the FIFO status register (SSIFSR) will be set. The TDE flag is set to 1 when the number of bytes for transmission in the transmit FIFO data register (SSIFTDR) has fallen to or below the trigger number corresponding to the setting as shown below. 00: 7 (1)* 01: 6 (2)* 10: 4 (4)* 11: 2 (6)* Note: * The values in parentheses are the number of empty stages in SSIFTDR at which the TDE flag is set.
5, 4	RTRG[1:0]	00	R/W	Receive Data Trigger Number When the FIFO is operating for reception, these bits specify the number of received bytes in the FIFO (trigger number for reception) at which the RDF flag in the FIFO status register (SSIFSR) will be set. The RDF flag is set to 1 when the number of received bytes in the receive FIFO data register (SSIFRDR) has risen to or above the trigger number corresponding to the setting as shown below. 00: 1 01: 2 10: 4 11: 6
3	TIE	0	R/W	Transmit Interrupt Enable This bit enables or disables generation of transmit data empty interrupt (TXI) requests in the following situation: when the FIFO is operating for transmission, the data for transmission in the transmit FIFO data register (SSIFTDR) are transferred to the transmit data register (SSITDR) and the number of data bytes in the transmit FIFO data register has become less than the set transmit trigger number, so that the TDE flag in the FIFO status register (SSIFSR) is set to 1. 0: Transmit data empty interrupt (TXI) request is disabled. 1: Transmit data empty interrupt (TXI) request is enabled.* Note: * TXI can be cleared by clearing either the TDE flag (see the description of the TDE bit for details) or TIE bit to 0.

Bit	Bit Name	Initial Value	R/W	Description
2	RIE	0	R/W	Receive Interrupt Enable Enables or disables generation of receive data full interrupt (RXI) requests when the RDF flag in the FIFO status register (SSIFSR) is set to 1 while the FIFO is operating for reception. 0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.* Note: * RXI can be cleared by clearing either the RDF flag (see the description of the RDF bit for details) or RIE bit to 0.
1	TFRST	0	R/W	Transmit FIFO Data Register Reset Invalidates the data in the transmit FIFO data register (SSIFTDR) to reset the FIFO to an empty state. 0: Reset is disabled.* 1: Reset is enabled. Note: * FIFO is reset at a power-on reset. **Note:** Note2: When TFRST is set to 0 during transmission, SSIF must be reset for restart by SW reset (SG2RES or SG3RES) .
0	RFRST	0	R/W	Receive FIFO Data Register Reset Invalidates the data in the receive FIFO data register (SSIFRDR) to reset the FIFO to an empty state. 0: Reset is disabled.* 1: Reset is enabled. Note: * FIFO is reset at a power-on reset.

35.3.6 FIFO Status Register (SSIFSR)

SSIFSR contains status flags that indicate the state of operation of the transmit and receive FIFO data registers.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	TDC[3:0]				-	-	-	-	-	-	-	TDE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	RDC[3:0]				-	-	-	-	-	-	-	RDF
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/(W)*

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
27 to 24	TDC[3:0]	0000	R	Number of Data Bytes Stored in SSIFTDR TDC[3:0] = H'0 indicates no data for transmission. TDC[3:0] = H'8 indicates that 32 bytes of data for transmission is stored in SSIFTDR.
23 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	TDE	1	R/(W)*	Transmit Data Empty Indicates that, when the FIFO is operating for transmission, the data for transmission in the transmit FIFO data register (SSIFTDR) is transferred to the transmit data register (SSITDR), the number of data bytes in SSIFTDR has become less than the transmit trigger number specified by TTRG[1:0] in the FIFO control register (SSIFCR), and thus writing of transmit data to SSIFTDR has been enabled. 0: Number of data bytes for transmission in SSIFTDR is greater than the set transmit trigger number. [Clearing conditions] 0 is written to TDE after data of the number of bytes larger than the set transmit trigger number is written to SSIFTDR. The direct memory access controller is activated by transmit data empty (TXI) interrupt, and data of the number of bytes larger than the set transmit trigger number is written to SSIFTDR. 1: Number of data bytes for transmission in SSIFTDR is equal to or less than the set transmit trigger number.*1 [Setting conditions] Power-on reset Number of transmission data bytes stored in SSIFTDR has become equal to or less than the set transmit trigger number. Note: *1Since SSIFTDR is an 8-stage FIFO register, the amount of data that can be written to it while TDE = 1 is "8 – transmit trigger number to be specified" bytes at maximum. Writing more data will be ignored. The number of data bytes in SSIFTDR is indicated in the TDC[3:0] bits in SSIFSR.
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
11 to 8	RDC[3:0]	0000	R	Number of Data Bytes Stored in SSIFRDR RDC[3:0] = H'0 indicates no received data. RDC[3:0] = H'8 indicates that 32 bytes of received data is stored in SSIFRDR.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	RDF	0	R/(W)*	Receive Data Full Indicates that, when the FIFO is operating for reception, the received data is transferred to the receive FIFO data register (SSIFRDR) and the number of data bytes in SSIFRDR has become greater than the receive trigger number specified by RTRG[1:0] in the FIFO control register (SSIFCR). 0: Number of received data bytes in SSIFRDR is less than the set receive trigger number. [Clearing conditions] Power-on reset 0 is written to RDF after the receive FIFO is emptied with writing 1 to RFRST. 0 is written to RDF after data is read from SSIFRDR until the number of data bytes in SSIFRDR becomes less than the set receive trigger number. The direct memory access controller is activated by receive data full (RXI) interrupt, and data is read from SSIFRDR until the number of data bytes in SSIFRDR becomes less than the set receive trigger number. 1: Number of received data bytes in SSIFRDR is equal to or greater than the set receive trigger number. [Setting condition] Data of the number of bytes that is equal to or greater than the set receive trigger number is stored in SSIFRDR.*1 Note: *1Since SSIFRDR is an 8-stage FIFO register, the amount of data that can be read from it while RDF = 1 is the set receive trigger number of bytes at maximum. Continuing to read data from SSIFRDR after reading all the data will result in undefined data to be read. The number of data bytes in SSIFRDR is indicated in the RDC[3:0] bits in SSIFSR.

Note: * The bit can be read or written to. Writing 0 initializes the bit, but writing 1 is ignored.

35.3.7 Transmit FIFO Data Register (SSIFTDR)

SSIFTDR is a FIFO register consisting of eight stages of 32-bit registers for storing data to be serially transmitted. On detecting that the transmit data register (SSITDR) is empty, this module transfers the data for transmission written to SSIFTDR to SSITDR to start serial transmission, which can continue until SSIFTDR becomes empty. SSIFTDR can be written to by the CPU at any time.

Note that when SSIFTDR is full of transmit data (32 bytes), the next data cannot be written to it. If writing is attempted, it will be ignored and an overflow occurs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Note: * Not writable while FIFO is receiving data.

35.3.8 Receive FIFO Data Register (SSIFRDR)

SSIFRDR is a FIFO register consisting of eight stages of 32-bit registers for storing serially received data. When four bytes of data have been received, this module transfers the received data in the receive data register (SSIRDR) to SSIFRDR to complete reception operation. Reception can continue until 32 bytes of data have been stored to SSIFRDR. SSIFRDR can be read but cannot be written to by the CPU. Note that when SSIFRDR is read while it does not hold received data, the value read is undefined and a reception underflow will occur.

After SSIFRDR becomes full of received data, the data received thereafter will be lost and a receive overflow occurs.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

35.3.9 TDM Mode Register (SSITDMR)

SSITDMR is a 32-bit readable/writable register that enables or disables muting of receive data in direct transfer, TDM mode, and WS continue mode.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	CONT	-	-	-	-	-	-	-	TDM
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	—	0	R/W	Reserved Always write 0 to this bit.
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	CONT	0	R/W	WS Continue Mode 0: Disables WS continue mode. 1: Enables WS continue mode. Note: This bit can be set only in master mode (SCKD = 1 and SWSD = 1)
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TDM	0	R/W	TDM Mode 0: Disables TDM mode. 1: Enables TDM mode.

35.3.10 SSIF Clock Setting Register (SSIFCLKCFG)

SSIFCLKCFG is a 32-bit readable/writable register that selects the AUDIOφ clock and controls the SSIF clocks behaviour during debug break.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PRMR0ISSA	PRMR0ISMA	PRMR0ISMS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 3	—	0	R	Reserved The read value is undefined. The write value should always be 0.
2	PRMR0ISSA	0	R/W	AUDIOφ clock selection 0: SSIACKINT input 1: SSIACKEXT input
1	PRMR0ISMA	0	R/W	Selects to continue or stop AUDIOφ clock during debugging. • When the EPC.SVSTOP bit is set to 0: Continues AUDIOφ clock regardless of the setting of this bit. • When the EPC.SVSTOP bit is set to 1: 0: Continues AUDIOφ clock. 1: Stops AUDIOφ clock.
0	PRMR0ISMS	0	R/W	Selects to continue or stop SSISCK clock during debugging. • When the EPC.SVSTOP bit is set to 0: Continues SSISCK clock regardless of the setting of this bit. • When the EPC.SVSTOP bit is set to 1: 0: Continues SSISCK clock. 1: Stops SSISCK clock.

35.4 Operation Description

35.4.1 Bus Format

This module can operate as a transmitter or a receiver and can be configured into many serial bus formats in either mode.

The bus format can be selected from one of the twelve major modes shown in the table below.

Table 35.9 Bus Format for SSIF Module

	TEN	REN	SCKD	SWSD	TDM	MUEN	IIEN	TOIEN	TUIEN	ROIEN	RUIEN	CONT	SWSP	DEL	PDTA	SDTA	SPDP	SCKP	SWL[2:0]	DWL[2:0]	CHNL[1:0]
Non-Compression Slave Receiver	0	1	0	0	0	Control Bits							Configuration Bits								
Non-Compression Slave Transmitter	1	0	0	0	0																
Non-Compression Slave Transceiver*1	1	1	0	0	0																
Non-Compression Master Receiver	0	1	1	1	0																
Non-Compression Master Transmitter	1	0	1	1	0																
Non-Compression Master Transceiver*1	1	1	1	1	0																
TDM Slave Receiver	0	1	0	0	1								0	Configuration Bits							
TDM Slave Transmitter	1	0	0	0	1								0								
TDM Slave Transceiver*1	1	1	0	0	1								0								
TDM Master Receiver	0	1	1	1	1								0								
TDM Master Transmitter	1	0	1	1	1								0								
TDM Master Transceiver*1	1	1	1	1	1								0								

Note 1. Set the TEN and REN bits to 1 at the same time when using transceiver mode.

35.4.2 Non-Compressed Modes

The non-compressed modes support all serial audio streams split into channels. It supports the I^2S compatible format as well as many more variants on these modes.

(1) Slave Receiver

This mode allows the module to receive serial data from another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

(2) Slave Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

(3) Slave Transceiver

This mode allows serial data transmission and reception between this module and another device. The clock and word select signal used for the serial data stream is also supplied from an external device. If these signals do not conform to the format specified in the configuration fields of this module, operation is not guaranteed.

(4) Master Receiver

This mode allows the module to receive serial data from another device. The clock and word select signals are internally derived from the oversampling clock. The format of these signals is defined in the configuration fields of this module. If the incoming data from another device does not follow the configured format, operation is not guaranteed.

(5) Master Transmitter

This mode allows the module to transmit serial data to another device. The clock and word select signals are internally derived from the oversampling clock. The format of these signals is defined in the configuration fields of this module.

(6) Master Transceiver

This mode allows serial data transmission and reception between this module and another device. The clock and word select signals are internally derived from the oversampling clock. The format of these signals is defined in the configuration fields of this module.

(7) Operation Setting Related to Word Length

All bits related to the SSICR's word length are valid in non-compressed modes. There are many configurations this module supports, but some of the combinations are shown below for the I^2S compatible format, MSB-first and left-aligned format, and MSB-first and right-aligned format.

- I^2S Compatible Format

Figure 35.2 and **Figure 35.3** show the I^2S compatible formats without and with padding, respectively. Padding occurs when the data word length is smaller than the system word length.

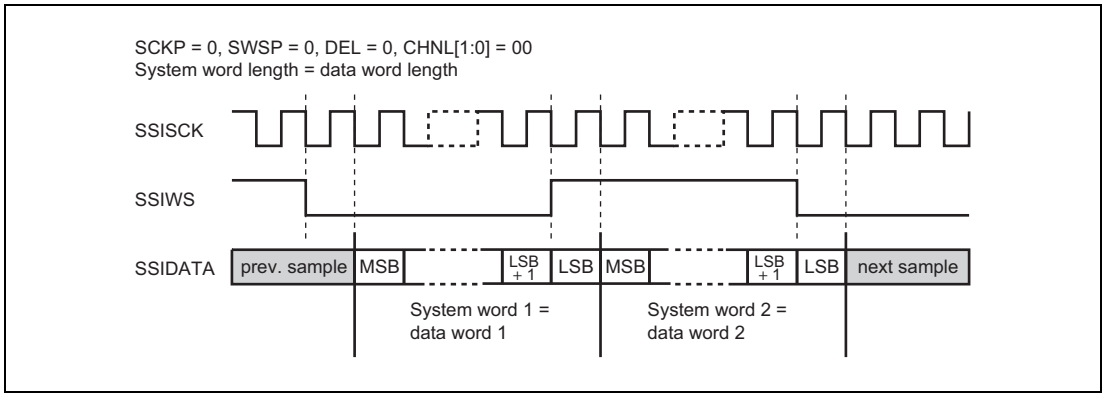


Figure 35.2 I^2S Compatible Format (without Padding)

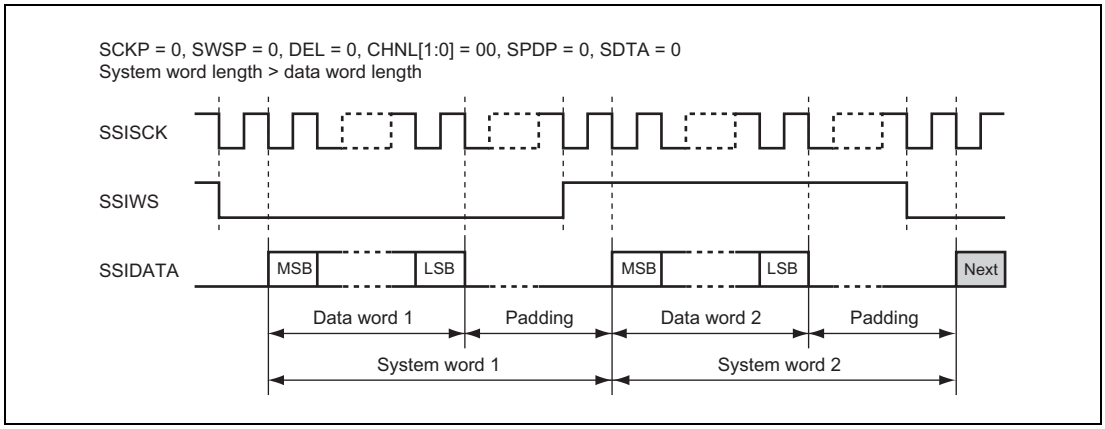


Figure 35.3 I2S Compatible Format (with Padding)

Figure 35.4 shows the MSB-first and left-aligned format and **Figure 35.5** shows the MSB-first and right-aligned format.

- MSB-first and Left-aligned Format

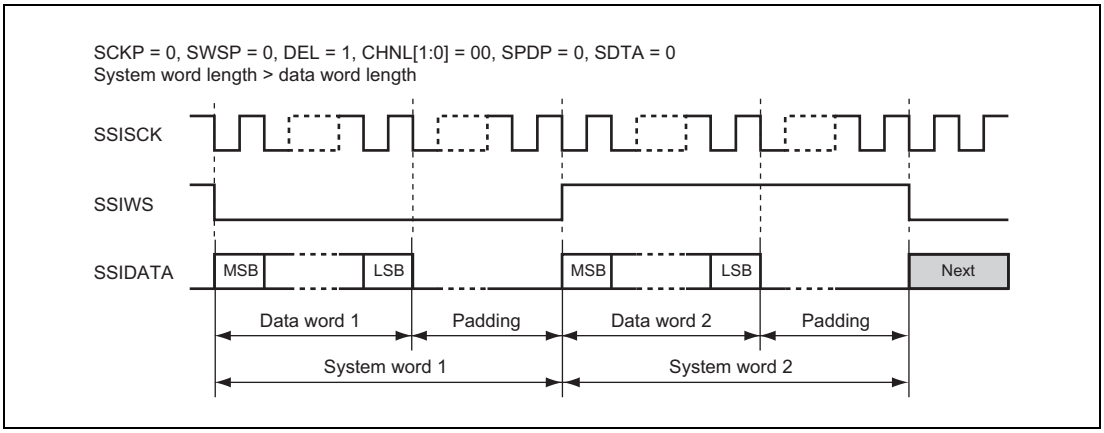


Figure 35.4 MSB-first and Left-aligned Format (Transmitted and Received in the Order of Serial Data and Padding Bits)

- MSB-first and Right-aligned Format

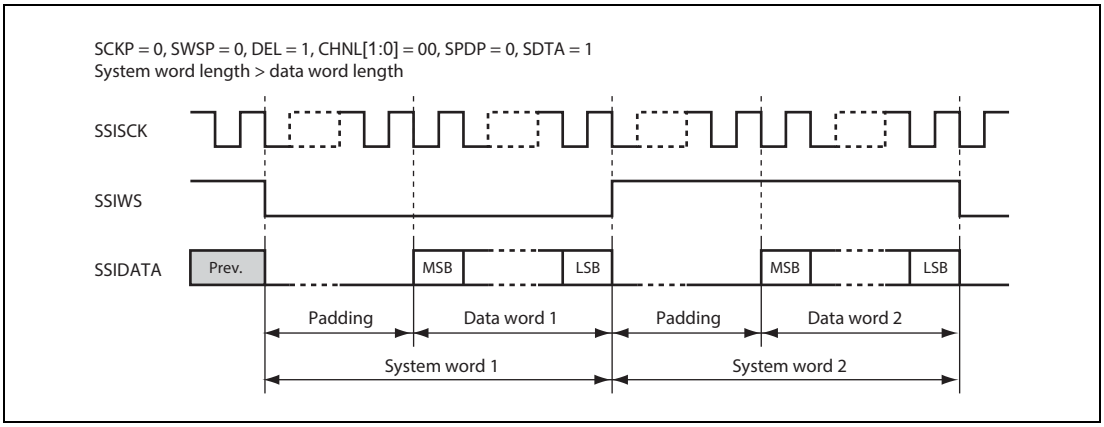


Figure 35.5 MSB-first and Right-aligned Format (Transmitted and Received in the Order of Padding Bits and Serial Data)

(8) Multi-channel Formats

Some devices extend the definition of the I^2S bus specification and allow more than 2 channels to be transferred within two system words.

This module supports the transfer of 4, 6, and 8 channels by using the CHNL[1:0], SWL[2:0] and DWL[2:0] bits only when the system word length (SWL[2:0]) is greater than or equal to the data word length (DWL[2:0]) multiplied by channels (CHNL[1:0]).

Table 35.10 shows the number of padding bits for each of the valid setting. If setting is not valid, "—" is indicated instead of a number.

Table 35.10 The Number of Padding Bits for Each Valid Setting

Padding Bits per System Word			DWL[2:0]	000	001	010	011	100	101	110
CHNL [1:0]	Decoded Channels per System Word	SWL [2:0]	Decoded Word Length	8	16	18	20	22	24	32
00	1	000	8	0	—	—	—	—	—	—
		001	16	8	0	—	—	—	—	—
		010	24	16	8	6	4	2	0	—
		011	32	24	16	14	12	10	8	0
		100	48	40	32	30	28	26	24	16
		101	64	56	48	46	44	42	40	32
		110	128	120	112	110	108	106	104	96
		111	256	248	240	238	236	234	232	224
01	2	000	8	—	—	—	—	—	—	—
		001	16	0	—	—	—	—	—	—
		010	24	8	—	—	—	—	—	—
		011	32	16	0	—	—	—	—	—
		100	48	32	16	12	8	4	0	—
		101	64	48	32	28	24	20	16	0
		110	128	112	96	92	88	84	80	64
		111	256	240	224	220	216	212	208	192
10	3	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	0	—	—	—	—	—	—
		011	32	8	—	—	—	—	—	—
		100	48	24	0	—	—	—	—	—
		101	64	40	16	10	4	—	—	—
		110	128	104	80	74	68	62	56	32
		111	256	232	208	202	196	190	184	160
11	4	000	8	—	—	—	—	—	—	—
		001	16	—	—	—	—	—	—	—
		010	24	—	—	—	—	—	—	—
		011	32	0	—	—	—	—	—	—
		100	48	16	—	—	—	—	—	—
		101	64	32	0	—	—	—	—	—
		110	128	96	64	56	48	40	32	0
		111	256	224	192	184	176	168	160	128

When this module acts as a transmitter, each word written to SSITDR is transmitted to the serial audio bus in the order they are written. When this module acts as a receiver, each word received by the serial audio bus is read from the SSIRDR register in the order they are received.

The figures below show how the data on 4, 6, and 8 channels are transferred to the serial audio bus. Note that there are no padding bits in the first example (**Figure 35.6**), the second example (**Figure 35.7**) is left-aligned and the third (**Figure 35.8**) is right-aligned. The other conditions in these examples have been selected arbitrarily.

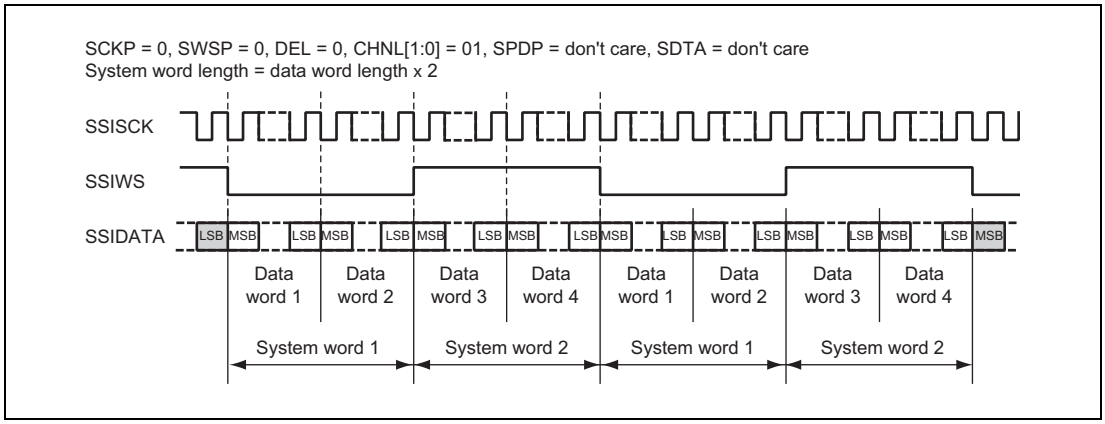


Figure 35.6 Multi-Channel Format (4 Channels Without Padding)

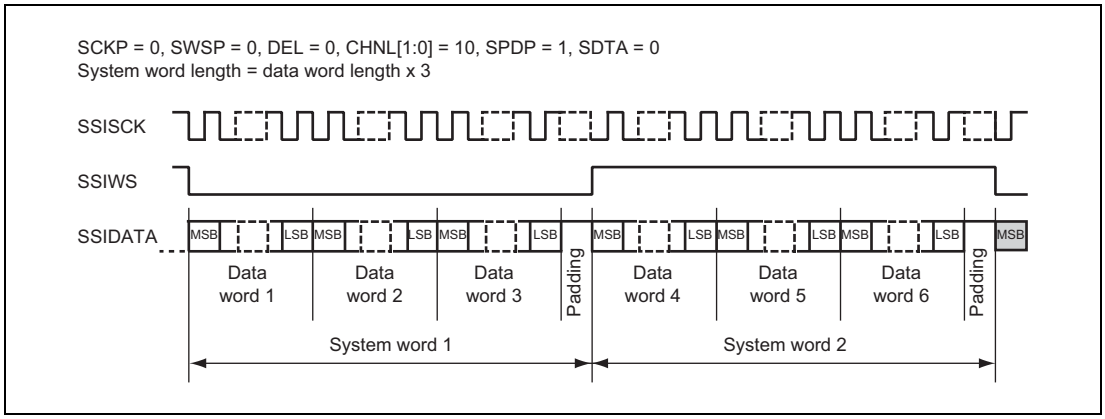


Figure 35.7 Multi-Channel Format (6 Channels with High Padding)

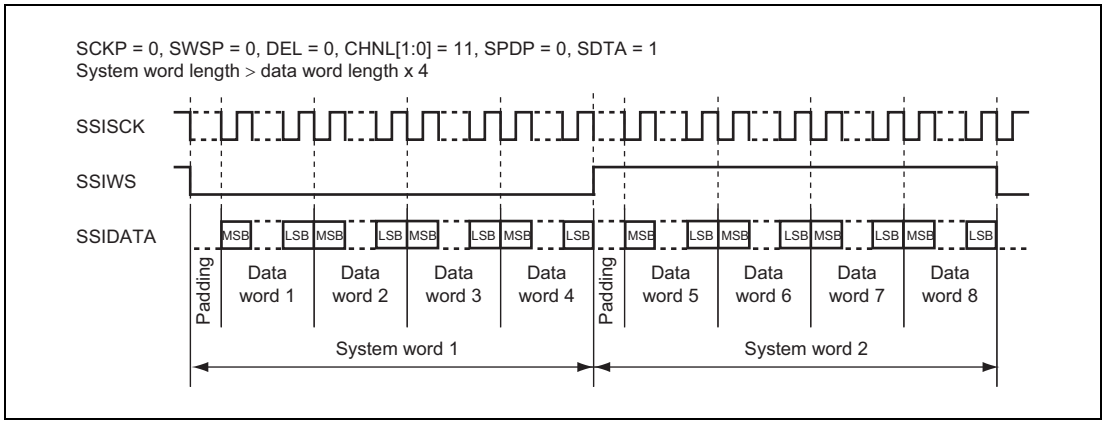


Figure 35.8 Multi-Channel Format (8 Channels; Transmitting and Receiving in the Order of Padding Bits and Serial Data ; with Padding)

(9) Operation Format Configuration Bits

Several more configuration bits in non-compressed mode are shown below. These bits are not mutually exclusive, but some combinations may not be useful.

These configuration bits are described below with reference to the basic sample format in the figure below.

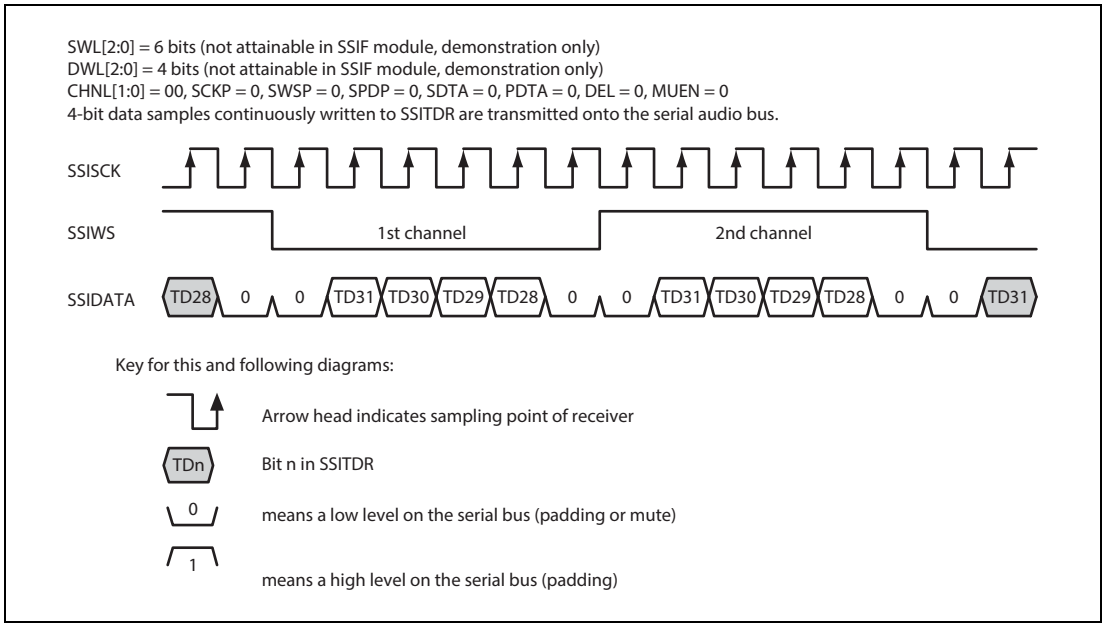


Figure 35.9 Basic Sample Format (Transmit Mode with Example System/Data Word Length)

The figure above uses a system word length of 6 bits and a data word length of 4 bits. These settings are not possible with this module but are used as an example only for clarification of the other configuration bits.

- Inverted Clock

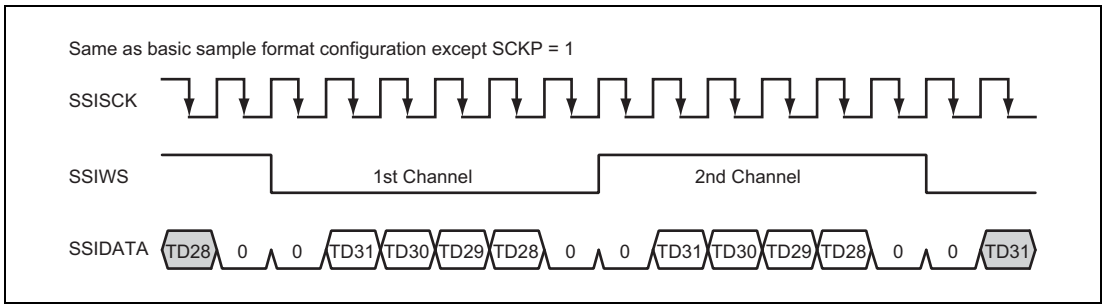


Figure 35.10 Inverted Clock

- Inverted Word Select

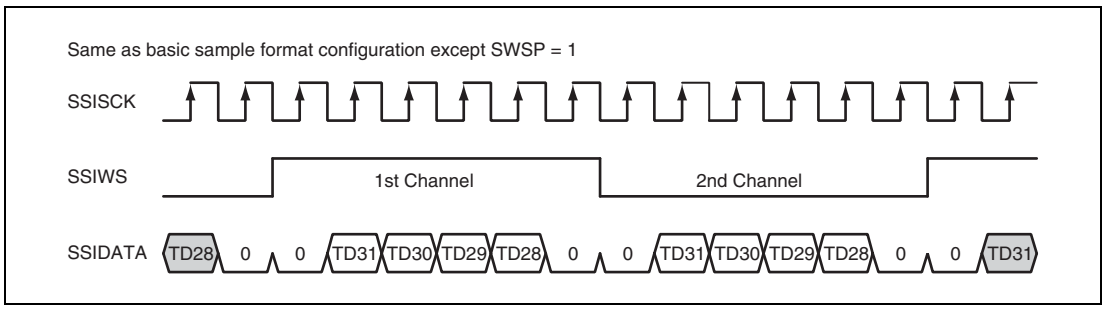


Figure 35.11 Inverted Word Select

- Inverted Padding Polarity

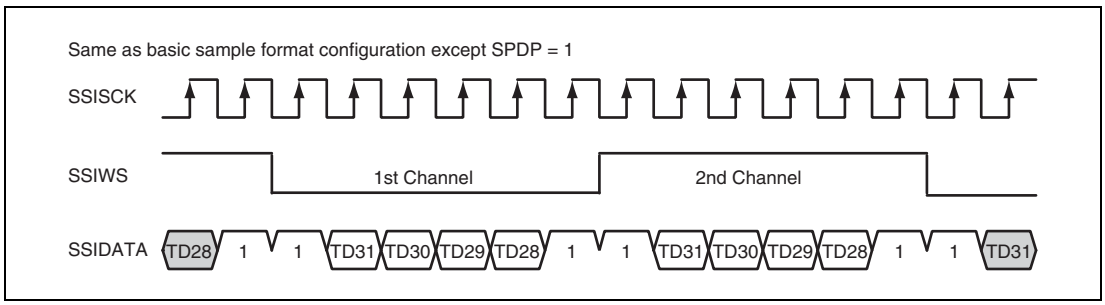


Figure 35.12 Inverted Padding Polarity

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

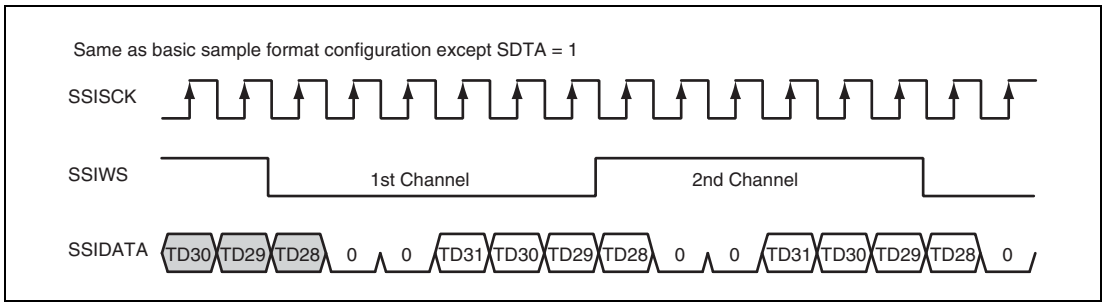


Figure 35.13 Transmitting and Receiving in the Order of Padding Bits and Serial Data; with Delay

- Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

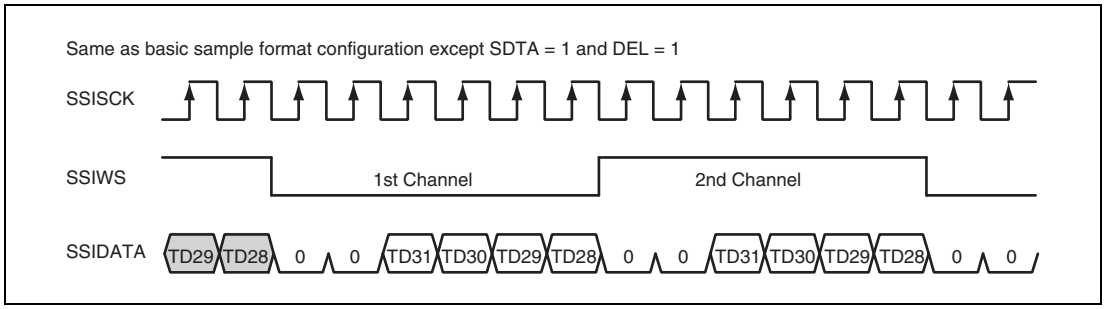


Figure 35.14 Transmitting and Receiving in the Order of Padding Bits and Serial Data; without Delay

- Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

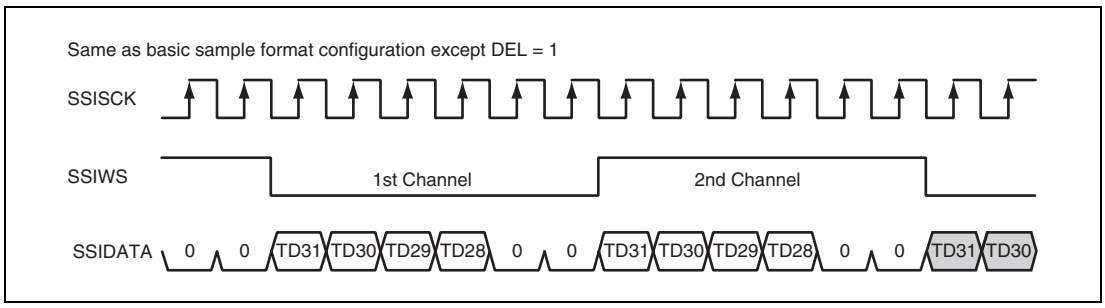


Figure 35.15 Transmitting and Receiving in the Order of Serial Data and Padding Bits; without Delay

- Parallel Right-Aligned with Delay

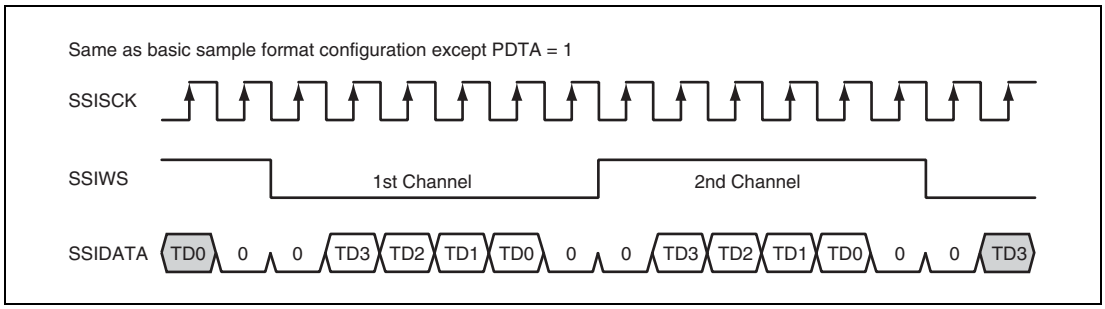


Figure 35.16 Parallel Right-Aligned with Delay

- Mute Enabled

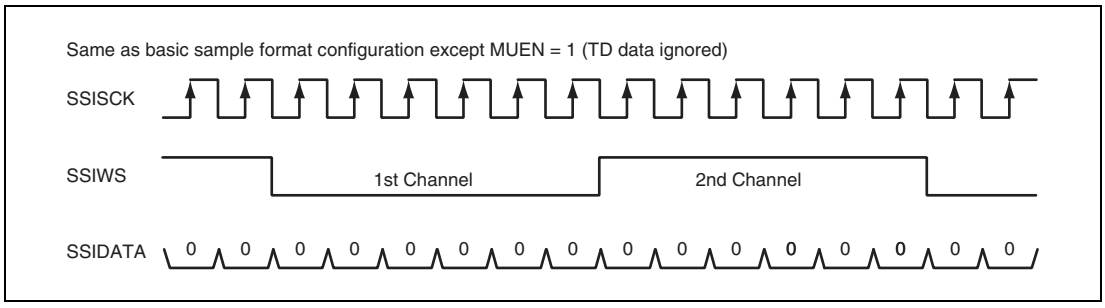


Figure 35.17 Mute Enabled

35.4.3 TDM Mode

TDM mode is provided to enable connection to multi-channel devices for TDM. This mode can be set using the TDM bit in the TDM mode register (SSITDMR). In this mode, the SSIWS signal is high only for system word 1 period and low for the other periods. The pulse produced on the SSIWS signal is defined as SYNC pulse. Note that the SYNC pulse always has the positive polarity (high only for system word 1 period).

The figures below show the TDM formats without and with padding, respectively.

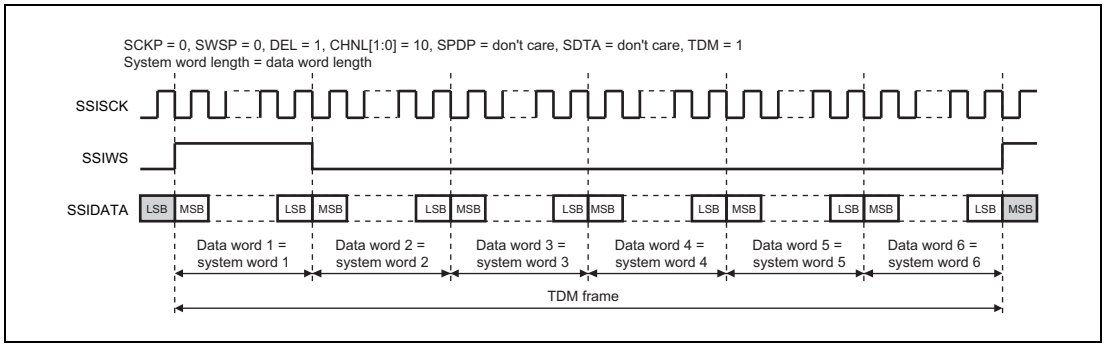


Figure 35.18 TDM Format (6 system words, no padding)

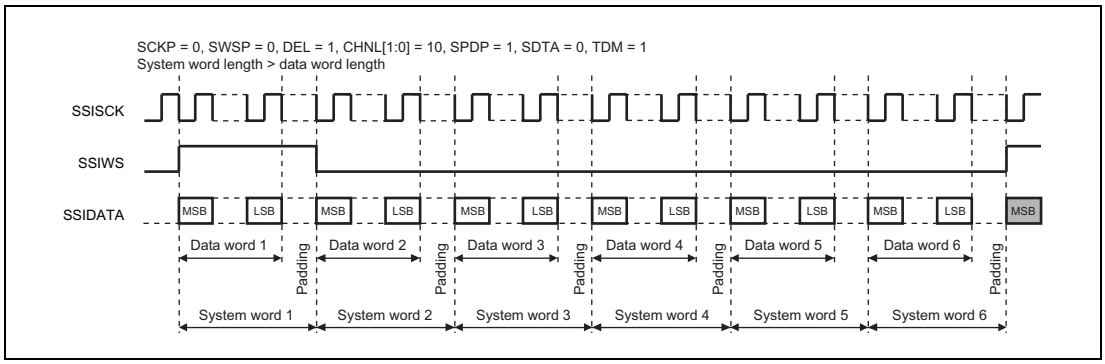


Figure 35.19 TDM Format (6 system words, with padding)

35.4.4 WS Continue Mode

In WS continue mode, the SSIWS signal continues to be output irrespective whether data transfer is enabled or disabled. This mode can be set using the CONT bit in the TDM mode register (SSITDMR). With this mode enabled, the SSIWS signal does not stop but continues operating even if TEN and REN bits in the control register (SSICR) are both set to 0 (transfer disabled). In case transfer is disabled while the MUEN bit in SSICR is set to 0, the SSITxD output keeps its last output data value until transfer is allowed again. With this mode disabled, the SSIWS signal stops if TEN and REN bits are both set to 0.

The figures below show the operations with WS continue mode enabled and disabled, respectively.

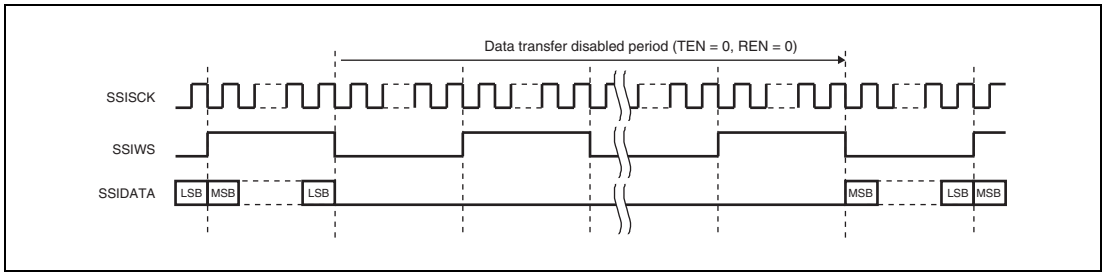


Figure 35.20 WS Continue Mode Enabled

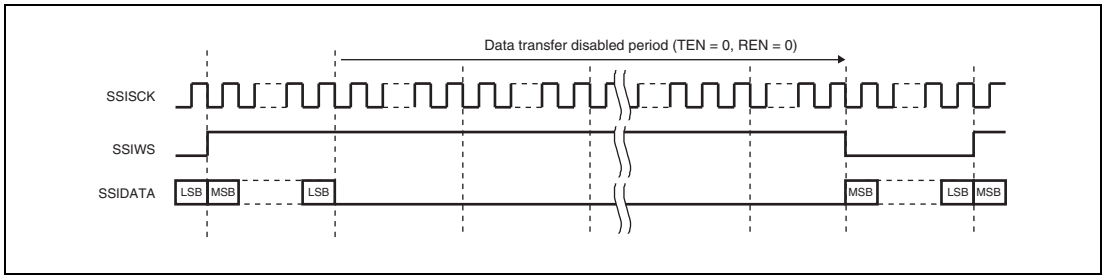


Figure 35.21 WS Continue Mode Disabled

35.4.5 Operation Modes

There are three modes of operation: configuration, enabled and disabled. The figure below shows how the module enters each of these modes.

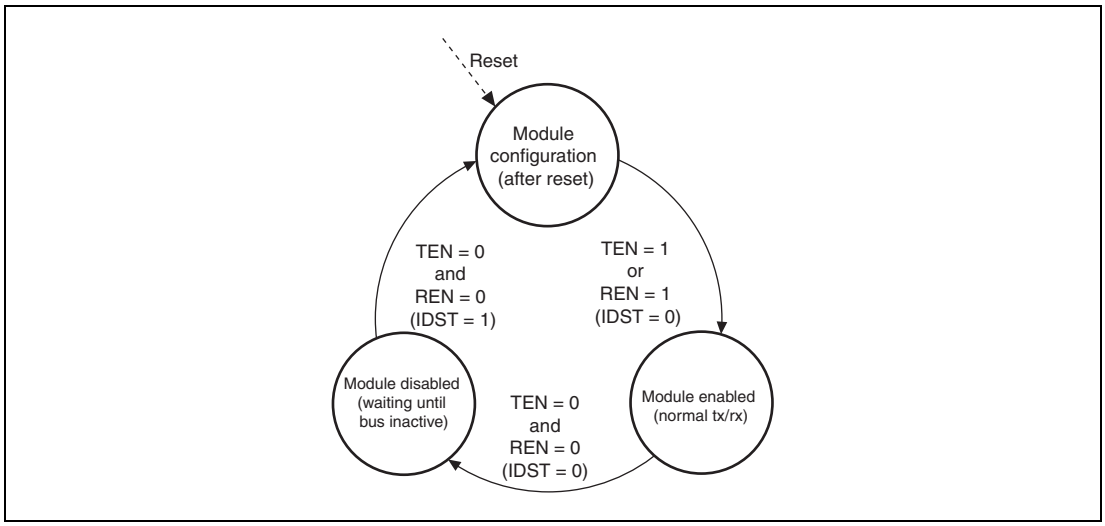


Figure 35.22 Operation Modes

(1) Configuration Mode

This mode is entered after the module is released from reset. All required configuration fields in the control register should be defined in this mode, before this module is enabled by setting the TEN and REN bits.

Setting the TEN and REN bits causes the module to enter the module enabled mode.

(2) Module Enabled Mode

Operation of the module in this mode is dependent on the operation mode selected. For details, refer to Section 35.4.6, Transmit Operation, and Section 35.4.7, Receive Operation, below.

35.4.6 Transmit Operation

Transmission can be controlled either by DMA transfer or interrupt.

DMA control is preferred to reduce the processor load. In DMA control mode, the processor will only receive interrupts if there is an underflow or overflow of data or if the DMA transfer has been completed.

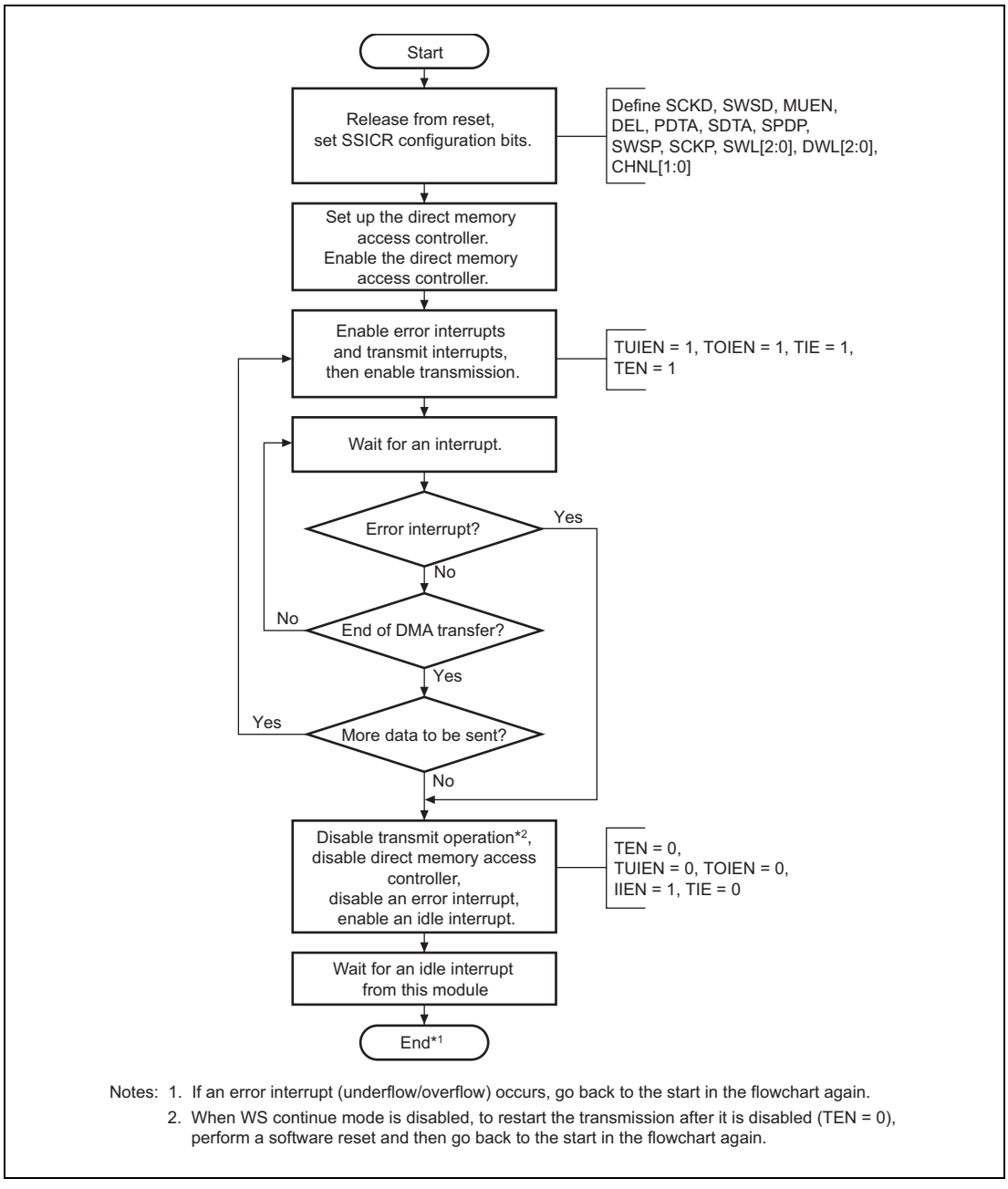
The alternative method is using the interrupts that this module generates to supply data as required.

When disabling this module, the clock* must be kept supplied to this module until the IIRQ bit indicates that the module is in the idle state.

Figure 35.23 shows the transmit operation in DMA control mode, and **Figure 35.24** shows the transmit operation in interrupt control mode.

Note: * Input clock from the SSISCK pin when SCKD = 0.
Oversampling clock when SCKD = 1.

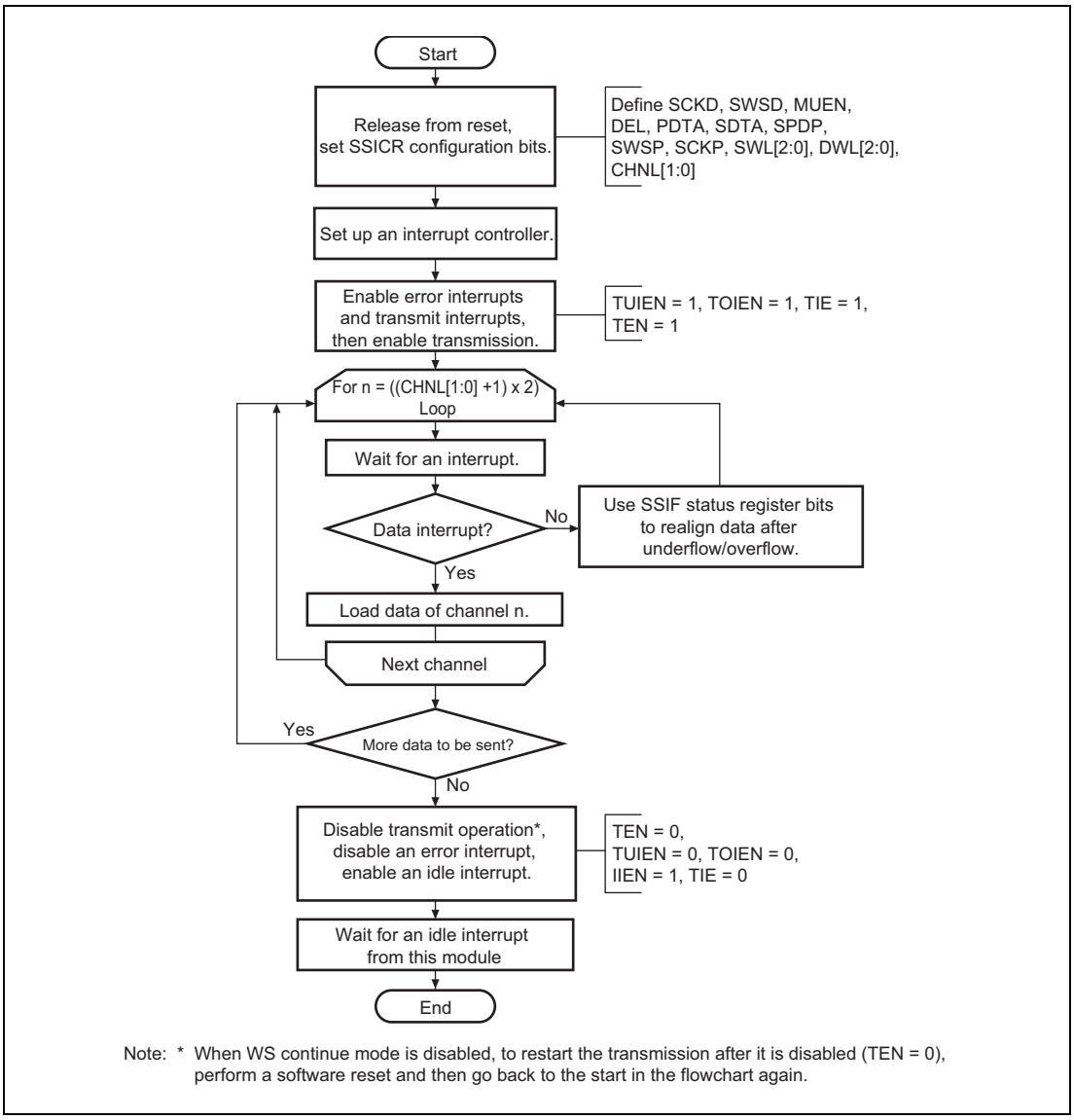
(1) Transmission Using Direct Memory Access Controller



Notes: 1. If an error interrupt (underflow/overflow) occurs, go back to the start in the flowchart again.
2. When WS continue mode is disabled, to restart the transmission after it is disabled (TEN = 0), perform a software reset and then go back to the start in the flowchart again.

Figure 35.23 Transmission Using Direct Memory Access Controller

(2) Transmission Using Interrupt-Driven Data Flow Control



Note: * When WS continue mode is disabled, to restart the transmission after it is disabled (TEN = 0), perform a software reset and then go back to the start in the flowchart again.

Figure 35.24 Transmission Using Interrupt-Driven Data Flow Control

35.4.7 Receive Operation

Like transmission, reception can be controlled either by DMA transfer or interrupt.

The figures below show the flow of operation.

When disabling this module, the clock* must be kept supplied to this module until the IIRQ bit indicates that the module is in the idle state.

Note: * Input clock from the SSISCK pin when SCKD = 0.
Oversampling clock when SCKD = 1.

(1) Reception Using Direct Memory Access Controller

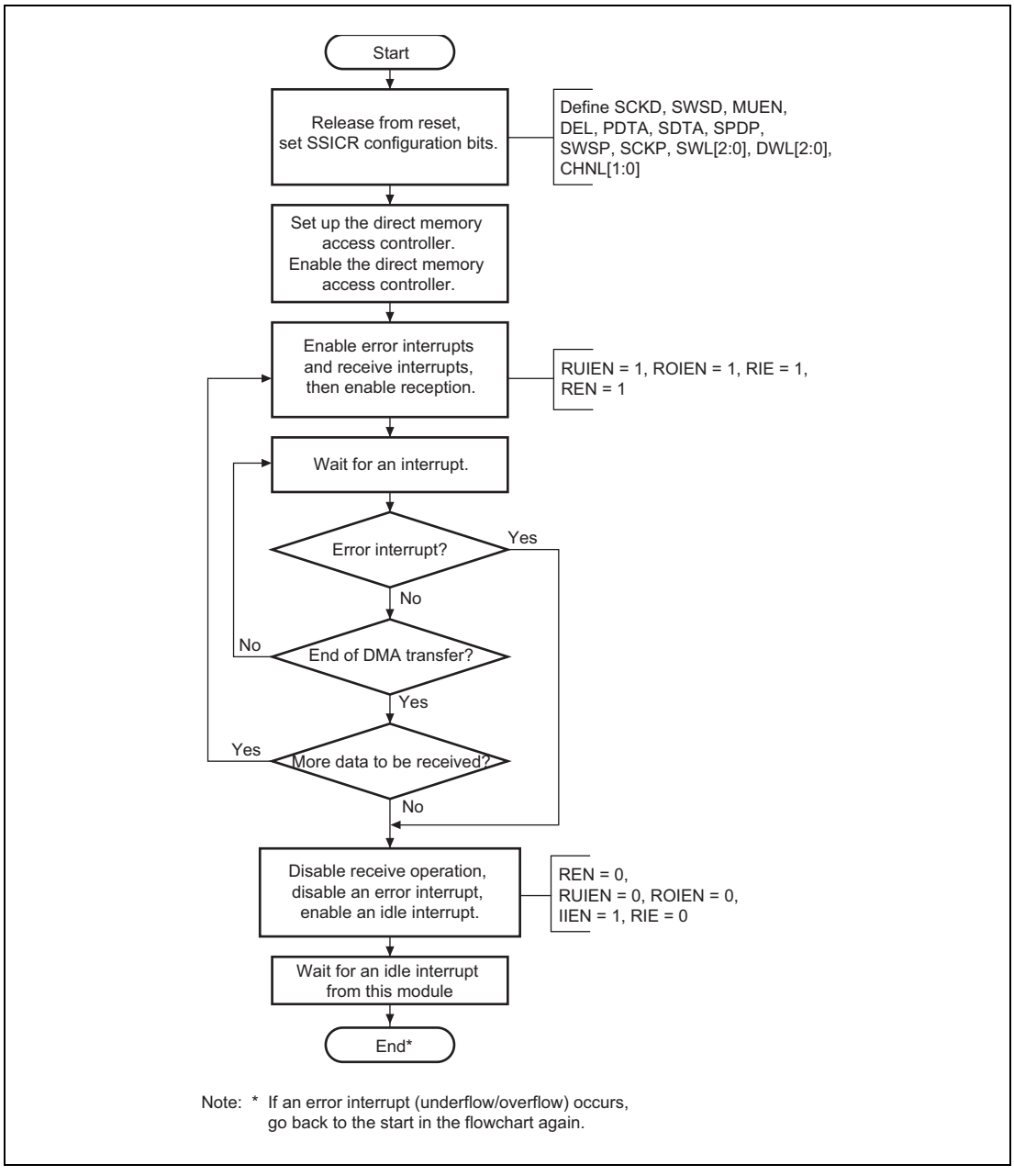


Figure 35.25 Reception Using Direct Memory Access Controller

(2) Reception Using Interrupt-Driven Data Flow Control

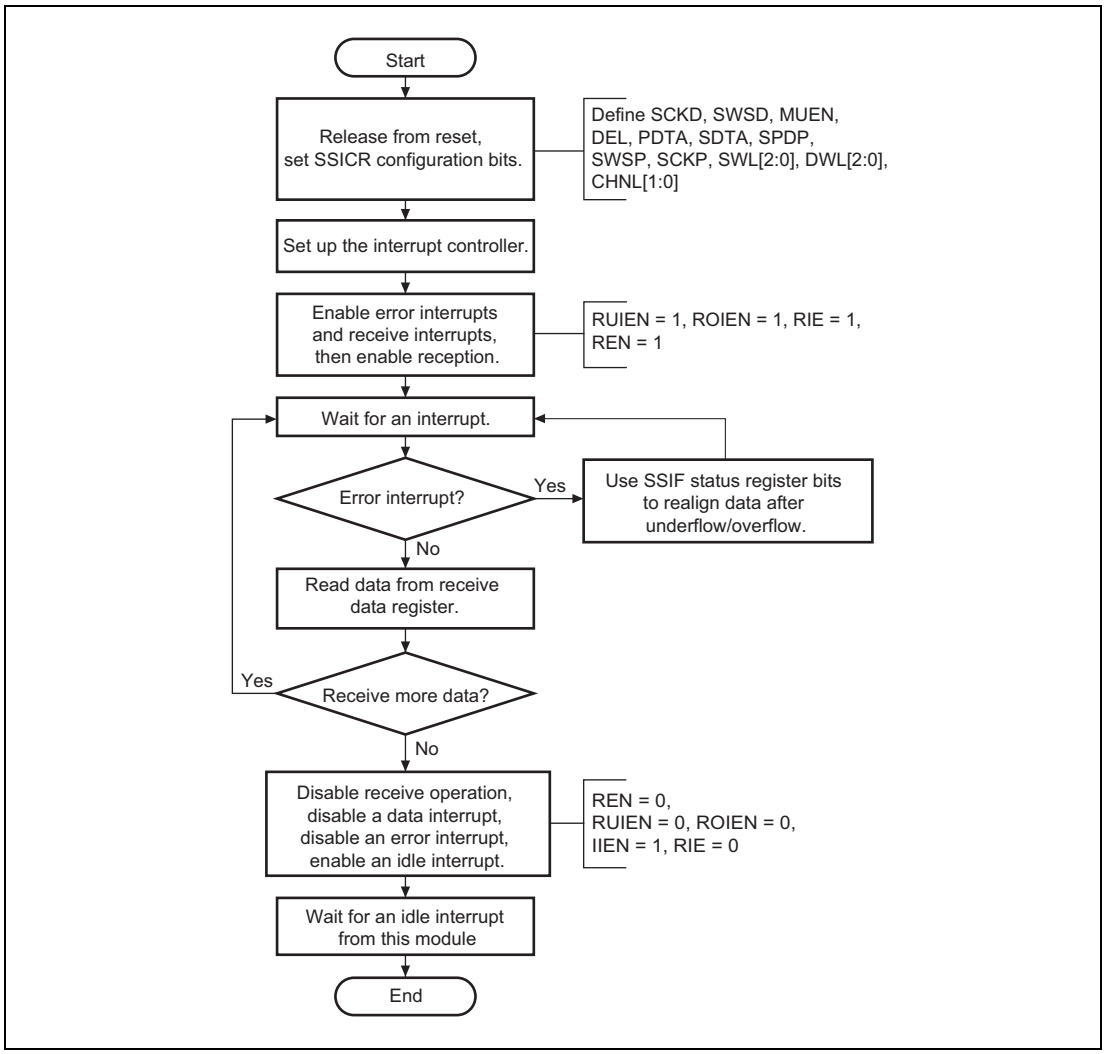


Figure 35.26 Reception Using Interrupt-Driven Data Flow Control

When an underflow or overflow error condition has matched, this module can be recovered to the status before underflow or overflow condition match by using the TCHNO[1:0] and TSWNO bits in transmission and the RCHNO[1:0] and RSWNO bits in reception. When an underflow or overflow occurs, the CPU can read the channel number and system word number to determine what point the serial audio stream has reached. In the transmitter case, the CPU can skip the data transmission until the SSIF module finds the data to transmit next, and so resynchronize with the audio data stream. In the receiver case the host CPU can store null data to make the number of receive data items consistent until it is ready to store the sample data that this module is indicating will be received next, and so resynchronize with the audio data stream.

35.4.8 Serial Bit Clock Control

This function is used to control and select the clock that is used for the serial bus interface.

If the serial bit clock direction is set to input (SCKD = 0), this module is in clock slave mode and the shift register uses the bit clock that was input to the SSISCK pin.

If the serial bit clock direction is set to output (SCKD = 1), this module is in clock master mode, and the shift register uses the oversampling clock or a divided oversampling clock as the bit clock. The oversampling clock is divided by the ratio specified by the serial oversampling clock division ratio bits (CKDV[3:0]) in SSICR for use as the bit clock by the shift register.

In either case above, the output of the SSISCK pin is the same as the bit clock.

35.5 Usage Notes

35.5.1 Limitations from Underflow or Overflow during DMA Operation

If an underflow or overflow occurs while the DMA is in operation, the SSIF module should be restarted. The transmit and receive buffers in the SSIF consists of 32-bit registers that share the L and R channels. Therefore, data to be transmitted and received at the L channel may sometimes be transmitted and received at the R channel if an underflow or overflow occurs, for example, under the following condition: the control register (SSICR) has a 32-bit setting for both data word length (DWL2 to DWL0) and system word length (SWL2 to SWL0).

If an error occurrence is confirmed with four types of error interrupts (transmit underflow, transmit overflow, receive underflow, and receive overflow) or the corresponding error status flags (the bits TUIRQ, TOIRQ, RUIRQ, and ROIRQ in SSISR), write 0 to the TEN or REN bit in SSICR to disable DMA transfer requests in this module, thus stopping the operation. (In this case, the direct memory access controller setting should also be stopped.) After this, write 0 to the error status flag bit to clear the error status, set the direct memory access controller again and restart the transfer.

35.5.2 Note on Changing Mode from Master Transceiver to Master Receiver

If a transmit underflow occurs in master transceiver mode while WS continue mode is disabled (SSITDMR.CONT = 0) and the TEN bit in SSICR is set to 0 in order to disable transmit operation, SSIWS output stops. In order to receive seamlessly after changing mode to master receiver mode, write dummy data to SSIFTDR to suppress transmit underflow.

35.5.3 Limits on TDM mode and WS Continue Mode

If TDM mode or WS continue mode setting is changed, the operation of the SSISCK and SSIWS signals immediately after switching are not guaranteed. If it affects the device to be connected, do not change the setting dynamically.

Section 36 LCD Bus Interface (LCBI)

This section contains a generic description of the LCD Bus Interface.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

36.1 Overview of the RH850/D1L/D1M LCD Bus Interfaces

36.1.1 Units

This microcontroller has the following number of units of the LCD Bus Interface.

Table 36.1 Units

LCD Bus I/F (LCBI)	D1L1	D1L2(H) D1M1 D1M1-V2	D1M1H D1M1A	D1M2(H)
Units	1	1	0	0
Names	LCBI0	LCBI0	–	–

Unit index n

Throughout this section, the individual units of the LCD Bus Interfaces are identified by the index “n” (n = 0), for example LCBInOPMODE for the LCD Bus Interface n operation mode control register.

36.1.2 Register addresses

All LCD Bus Interface register addresses are given as address offsets from the individual base addresses <LCBIn_base>.
The <LCBIn_base> addresses of each LCBIn are listed in the following table:

Table 36.2 Register base addresses <LCBIn_base>

LCBIn unit	<LCBIn_base> address
LCBI0	FFFB 0000_H

36.1.3 Clock supply

All LCD Bus Interfaces provide one clock input.

Table 36.3 Clock supply

LCBIn unit	LCBIn clock	Connected to
LCBI0	PCLK	Clock Controller C_ISO_LCBI

36.1.4 Interrupts and DMA

The LCD Bus Interfaces can generate the following interrupt and DMA requests:

Table 36.4 LCBIn interrupt and DMA requests

LCBIn signals	Function	Connected to
LCBI0:		
LCDIFTIRDY	Read data ready interrupt	Interrupt Controller INTLCBI0RDY DMA Controller trigger ID 12
LCDIFTIEMPT	Write buffer empty interrupt	Interrupt Controller INTLCBI0EMPTY DMA Controller trigger ID 13
LCDIFTIHALF	Write buffer half full interrupt	Interrupt Controller INTLCBI0HALF DMA Controller trigger ID 14
LCDIFTIFULL	Write buffer full interrupt	Interrupt Controller INTLCBI0FULL DMA Controller trigger ID 15
LCDIFTIQTR	Write buffer quarter full interrupt	Interrupt Controller INTLCBI0QTR DMA Controller trigger ID 16
LCDIFTI3QTR	Write buffer three quarters full interrupt	Interrupt Controller INTLCBI03QTR DMA Controller trigger ID 17

36.1.5 Reset sources

The LCD Bus Interfaces and their registers are initialized by the following reset signal:

Table 36.5 Reset sources

LCBIn unit	Reset signal
LCBI0	• Reset Controller SYSRES • Reset Controller LCBI0RES • reset upon wake-up from DEEPSTOP mode

CAUTION

By default the LCBI0RES reset is active.
Thus before accessing this module this reset must be released via the MRSTC register of the Reset Controller. Refer to Section 9, Reset Controller for details.

36.1.6 I/O signals

The following table shows the I/O signals of the LCD Bus Interfaces.

Table 36.6 I/O signals connections

LCBIn signal	Function	Connected to
LCBI0:		
LCDDATA[17:0]	Data I/O	Port LCBI0D[17:0]
$\overline{RD}$/E/PixelCLK	• Non-TFT operation mode: – RAM mode: Read strobe $\overline{RD}$ – E-Type mode: E strobe E • TFT operation mode: Pixel clock	Port $\overline{LCBI0RDE}$
$\overline{WR}$/RW	• RAM mode: Write strobe $\overline{WR}$ • E-Type mode: Read/Write strobe RW	Port $\overline{LCBI0WRRW}$
A0/DE	• Non-TFT operation mode: Data/control selection A0 • TFT operation mode: Data enable DE	Port LCBI0A0DE
$\overline{CS}$	Chip select	Port $\overline{LCBI0CS}$
$\overline{HSYNC}$	TFT horizontal synchronization	Port $\overline{LCBI0HSYNC}$
$\overline{VSYNC}$	TFT vertical synchronization	Port $\overline{LCBI0VSYNC}$

36.2 Functional Overview

Features summary

The LCD Bus Interface has the following features:

- Support of TFT and non-TFT LCDs
- Support of color LCDs using a color lookup table (CLUT)
  - CLUT RAM size: 256 x 18 bits
  - 16 color and 256 color palette
- Byte/halfword/word read and write operations
- DMA support for read and write operations
- 8 / 9 / 16 / 18 bit LCD bus width
  Data can be shifted so that it is accessed using different bits of the LCD bus (e.g. 8 bits can be written to bit position [7:0], [8:1], [9:2], or [17:10])
- Buffering of write data to avoid wait states on CPU/DMA access
  Interrupts indicate buffer fill state.
- Interrupt generation when LCD bus reading transfer is complete (read data ready interrupt)
- Flags that indicate the bus cycle state

The LCD Bus Interface has the following additional features for non-TFT LCDs:

- Support of two different control signal modes:
  - RAM mode with separate read and write strobe
  - E-Type mode with read/write signal and data strobe “E”
- Programmable transfer speed (100 kHz … 10 MHz) through
  - Selectable prescaler for operation clock
  - Programmable transfer time
  - Programmable wait states
  - Selectable fast bus cycle mode
- LCD bus timing is generated by hardware using A0 and $\overline{CS}$ control lines

The LCD Bus Interface has the following additional features for TFT LCDs:

- $\overline{HSYNC}$ and $\overline{VSYNC}$ signal generation
- Programmable pixel clock
- Programmable video synchronization timing

The following block diagram shows the main components of the LCD Bus Interface.

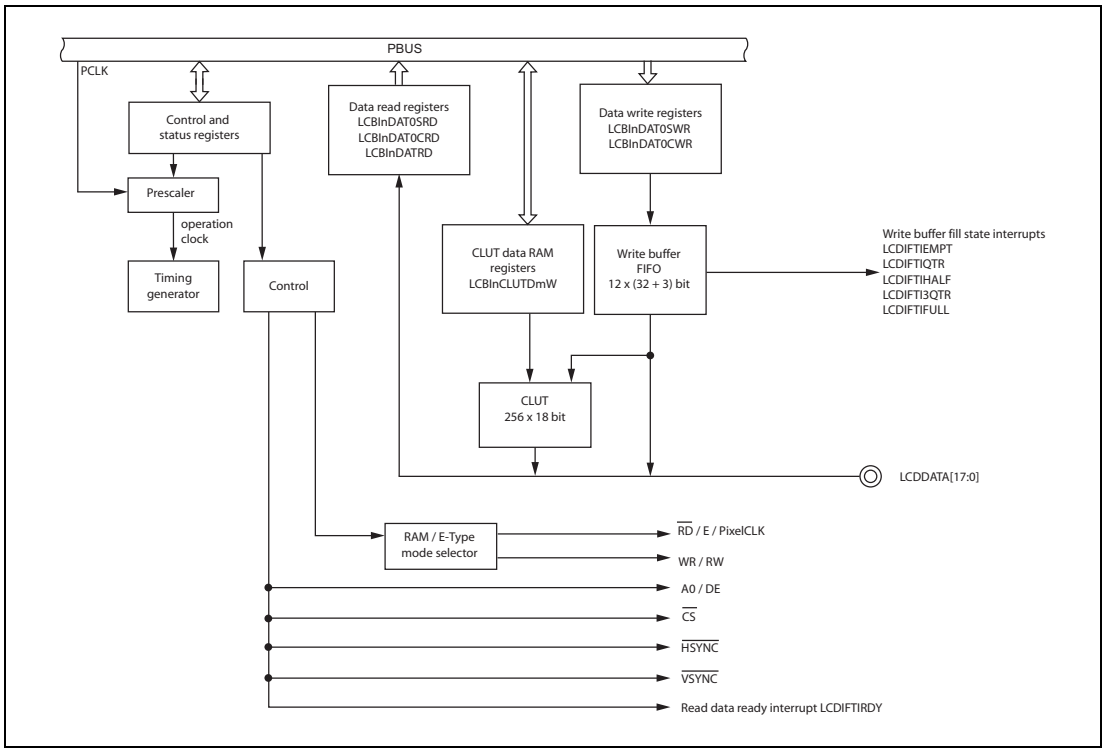


Figure 36.1 Block diagram of the LCD Bus Interface

36.3 Functional Description

The LCD Bus Interface connects the microcontroller to an external LCD controller. It provides an asynchronous 18-bit parallel data bus to send data to, and query data from, the LCD controller. Control lines specify the read/write timing and the transfer direction.

Non-TFT and TFT displays

The LCD Bus Interface supports both TFT and non-TFT LCDs using the corresponding operation mode, control registers, and control lines. The LCD bus has two operation modes that are selected by bits LCBInOPMODE.LCBInOPMODE[1:0].

For details on the operation modes, refer to

- Section 36.3.1, Non-TFT operation mode
- Section 36.3.2, TFT operation mode

Color display

To support color displays, the LCD Bus Interface supports the conversion of write data using a Color Lookup Table (CLUT). CLUT usage is possible in both operation modes.

For details on CLUT usage, refer to Section 36.3.3, Color usage.

CPU or DMA

Data can be read from, and written to, the LCD Bus Interface using the DMA Controller or by directly accessing the interface from the CPU.

Timing configuration

The timing of the LCD bus signals is determined by register settings.

Bus width

The LCD Bus Interface is 18 bits wide. It supports LCD controllers requiring 8-bit, 9-bit, 16-bit, or 18-bit data. In order to improve performance, the interface is equipped with 32-bit registers that allow the CPU or DMA to write and read data of 8-bit, 16-bit or 32-bit of payload size. The interface automatically generates an appropriate number of consecutive 8-bit, 9-bit, 16-bit, or 18-bit accesses on the LCD bus.

Write buffer

The LCD Bus Interface has an internal bit write buffer (FIFO) that allows the next data to be written to the data write registers while a transfer on the LCD bus is in progress.

The result of a read operation is directly available in the data read registers.

Operation clock

The operation clock of the LCD Bus Interface is derived from PCLK by a prescaler specified in the LCBInCKSEL register.

For TFT displays, the pixel clock is derived from the operation clock.

36.3.1 Non-TFT operation mode

Access modes and control signals

In non-TFT operation mode (LCBInOPMODE.LCBInOPMODE[1:0] = 01_B), the LCD Bus Interface can access an external LCD controller in two different access modes. The mode is selected by bit LCBInBCYCT.LCBInBCYTYPE.

- RAM mode (LCBInBCYCT.LCBInBCYTYPE = 0)

  The control signals $\overline{WR}$ and $\overline{RD}$ are used to control the external component.

- E-Type mode (LCBInBCYCT.LCBInBCYTYPE = 1)

  The control signals RW and E are used to control the external component.

36.3.1.1 Access types

The following access types are possible:

- Read or write access
- Display data or control data
- Byte, halfword, or word access

For every access type, a dedicated data register is provided. The name of the data register specifies the access type. For details on the data registers, refer to Section 36.4.5, LCD Bus Interface data registers details.

Write operation

Write access to the LCD bus is performed by writing data to one of the data write registers. The data is transferred to the write buffer (FIFO). If there is a transfer in progress, the new data is held in the write buffer. As soon as the current transfer has completed, the new data is fetched from the write buffer and transferred to the LCD controller.

Display data or control data can be written. Different data write registers and the A0 control line are used to distinguish the two data types.

- Data written to LCBInDAT0SWR registers is display data.

  It is converted by the CLUT, if enabled (LCBInTCONTROL.LCBInDTCC = 1), and written to the LCD bus.
  While writing display data, the A0 control line is active according to the settings in the output level register LCBInOUTLEV.

- Data written to LCBInDAT0CWR registers is control data.

  It is directly written to the LCD bus. The CLUT is not considered. The setting of control bit LCBInTCONTROL.LCBInDTCC is ignored.
  While writing control data, the A0 control line is inactive according to the settings in the output level register LCBInOUTLEV.

Depending on the LCD bus width, a single write access can result in one or more write accesses on the LCD bus.

For example, writing the data register LCBInDAT0SWRH writes 16 bits of display data to the LCD bus. If the LCD bus width is 8 bit, 2 consecutive write accesses are performed.

Read operation

When new data is available to read, the read data ready interrupt request LCDIFTIRDY is generated.

When the CPU or the DMA reads one of the data read registers, the read operation on the LCD Bus Interface is started.

The value read from the register is the data from the *previous* transfer. Therefore, an initial dummy read operation is required to update the register.

As soon as the data of the current transfer is available in the data read registers, the read data ready interrupt request LCDIFTIRDY is generated. The data can be retrieved with the next read operation:

- If control data shall be read, read the LCBInDAT0CRD register of the desired size.
- If display data shall be read, read the LCBInDAT0SRD register of the desired size.

Depending on the LCD bus width, a single read access can result in one or more read accesses on the LCD bus.

For example, reading the display data register LCBInDAT0SRDW reads 32 bits of new display data from the LCD bus. If the LCD bus width is 8 bit, 4 consecutive read accesses are performed.

When reading, the CLUT is not considered. The setting of control bit LCBInTCONTROL.LCBInDTCC is ignored.

During the read operation, similarly to the write operation, the A0 control line indicates whether the requested read data is display data or control data.

Read operation without initiating a bus transfer

Data can be read from the LCBInDATRDW data read register without initiating a new read transfer via the LCD Bus Interface.

The read access to the LCBInDATRDW data read register is useful when previous read accesses to any of the data read registers have been performed and only the most-recently transferred data should be read without starting a new LCD bus transfer.

CAUTION

When a read access is triggered (data read register is read), the old data of the data read register is overwritten. To avoid loss of data, the application must handle read accesses properly.

Priorities and pending data

Only one read access can be pending. There is no read buffer.

Up to 12 write accesses can be pending in the write buffer.

If a read transfer is initiated while another read transfer is already pending, the new read transfer is ignored. The status flag LCBInSTATUS.LCBInTCRPG indicates whether a read transfer is pending or not.

If both read and write transfers are pending, the write transfers have higher priority: write transfer is performed before read transfer.

36.3.1.2 Transfer speed

The LCD Bus Interface provides programmable transfer speeds from approximately 100 kHz to 10 MHz with a minimum of 20 steps. The value is specified using the following control registers:

- LCBInCKSEL
  This register sets the operation clock from PCLK, PCLK/2, PCLK/4, ..., PCLK/64.
- LCBInBCYC
  This register defines the number of operation clock cycles for one bus cycle. Wait states for different states within one bus cycle can be specified separately. For example, the number of operation clock cycles for activating the control signals.
  Note that the number of clock cycles equals the LCBInBCYC register bits value +1.
- LCBInOPMODE
  To improve performance, fast bus cycle mode can be activated by bit LCBInOPMODE.LCBInOPMFAST.
  In fast bus cycle mode the LCBInBCYC register timing settings LCBInTIAD[3:0], LCBInTIDW[3:0], LCBInTMAD[3:0] and LCBInTMDW[3:0] are taken as 0.

After reset, the slowest clock is selected as default.

The transfer speed depends on RAM or E-Type, slow or fast bus cycle mode and the timing settings in the LCBInBCYC register:

(1) RAM mode (LCBInBCYCT.LCBInBCYTYPE = 0)

- Slow bus cycle RAM *read* mode (LCBInOPMODE.LCBInOPMFAST = 0)

  Transfer speed = TIAD + TIDW + TIRD + TIDZ
  = (LCBInTIAD[3:0] + 1) + (LCBInTIDW[3:0] + 1) + (LCBInTIRD[7:0] + 1) + (LCBInTIDZ[3:0] + 1)

- Slow bus cycle RAM *write* mode (LCBInOPMODE.LCBInOPMFAST = 0)

  Transfer speed = TIAD + TIDW + TIWR + TIDZ
  = (LCBInTIAD[3:0] + 1) + (LCBInTIDW[3:0] + 1) + (LCBInTIWR[7:0] + 1) + (LCBInTIDZ[3:0] + 1)

- Fast bus cycle RAM *read* mode (LCBInOPMODE.LCBInOPMFAST = 1)

  Transfer speed = TIRD + TIDZ
  = (LCBInTIRD[7:0] + 1) + (LCBInTIDZ[3:0] + 1)

- Fast bus cycle RAM *write* mode (LCBInOPMODE.LCBInOPMFAST = 1)

  Transfer speed = TIWR + TIDZ
  = (LCBInTIWR[7:0] + 1) + (LCBInTIDZ[3:0] + 1)

(2) E-Type mode (LCBInBCYCT.LCBInBCYTYPE = 1)

- Slow bus cycle E-Type mode (LCBInOPMODE.LCBInOPMFAST = 0)

$$\begin{aligned}\text{Transfer speed} &= \text{TMAD} + \text{TMDW} + \text{TMED} + \text{TMDZ}\\ &= (\text{LCBInTMAD}[3:0] + 1) + (\text{LCBInTMDW}[3:0] + 1) + (\text{LCBInTMED}[7:0] + 1) + (\text{LCBInTMDZ}[3:0] + 1)\end{aligned}$$

- Fast bus cycle E-Type mode (LCBInOPMODE.LCBInOPMFAST = 1)

$$\begin{aligned}\text{Transfer speed} &= \text{TMED} + \text{TMDZ}\\ &= (\text{LCBInTMED}[7:0] + 1) + (\text{LCBInTMDZ}[3:0] + 1)\end{aligned}$$

36.3.1.3 General timing

Every bus cycle is split into bus cycle states. The bus cycle states that are valid for a bus cycle depend on the selected access mode (RAM or E-Type) and the transfer direction (read or write). The duration of every bus cycle state can be configured separately.

The following timing diagrams show the general timing of a bus cycle in RAM and E-Type mode for both read and write operation.

CAUTION

If the FIFO of LCBI runs empty, the $\overline{\text{CS}}$ signal returns to its inactive state after the current bus cycle. Especially after feeding the first data into the LCBI, this may happen already after the first LCD bus cycle.

(1) General timing in RAM mode

The data transmission in RAM mode is split into the following states:

Table 36.7 Data transmission states in RAM mode

State	Control bit	State number*1
No activity between two transmissions	–	0
Waiting for address	LCBInTIAD[3:0]	1
Activating read/write strobe	LCBInTIDW[3:0]	2
Width of the read strobe	LCBInTIRD[7:0]	3
End of the bus cycle	LCBInTIDZ[3:0]	4
Width of the write strobe	LCBInTIWR[7:0]	5

Note 1. The state number refers to the state line in the following RAM mode diagrams.

The following figure shows the general timing of a *read* operation in RAM mode.

Control signals level

The following figures assume following settings of the control signals levels:

- LCBInOUTLEV.LCBInTLCS = 0: $\overline{\text{CS}}$ is active-low
- LCBInOUTLEV.LCBInTLRDE = 0: $\overline{\text{RD}}$ is active-low
- LCBInOUTLEV.LCBInTLWR = 0: $\overline{\text{WR}}$ is active-low
- LCBInOUTLEV.LCBInTLA0 is set as required to indicate transmission of control or data information

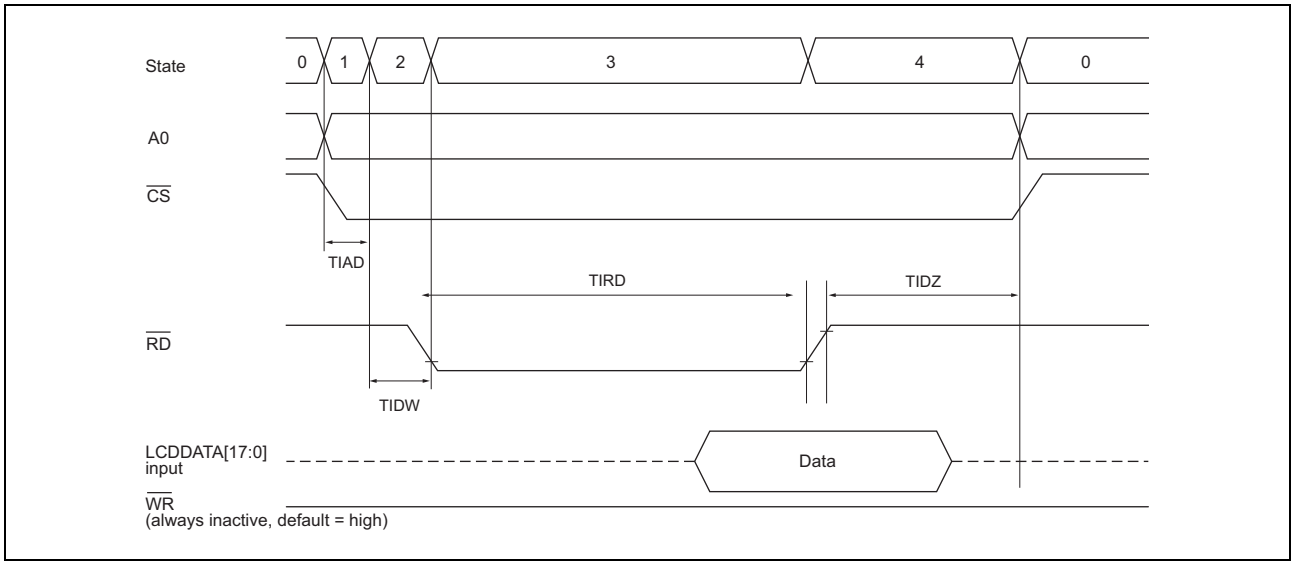


Figure 36.2 Timing diagram of a read operation in RAM mode

$\overline{CS}$ is active while the LCD Bus Interface is active. When the read strobe $\overline{RD}$ is active, data is read from the LCD bus. The write strobe $\overline{WR}$ is inactive.

The following figure shows the general timing of a *write* operation in RAM mode.

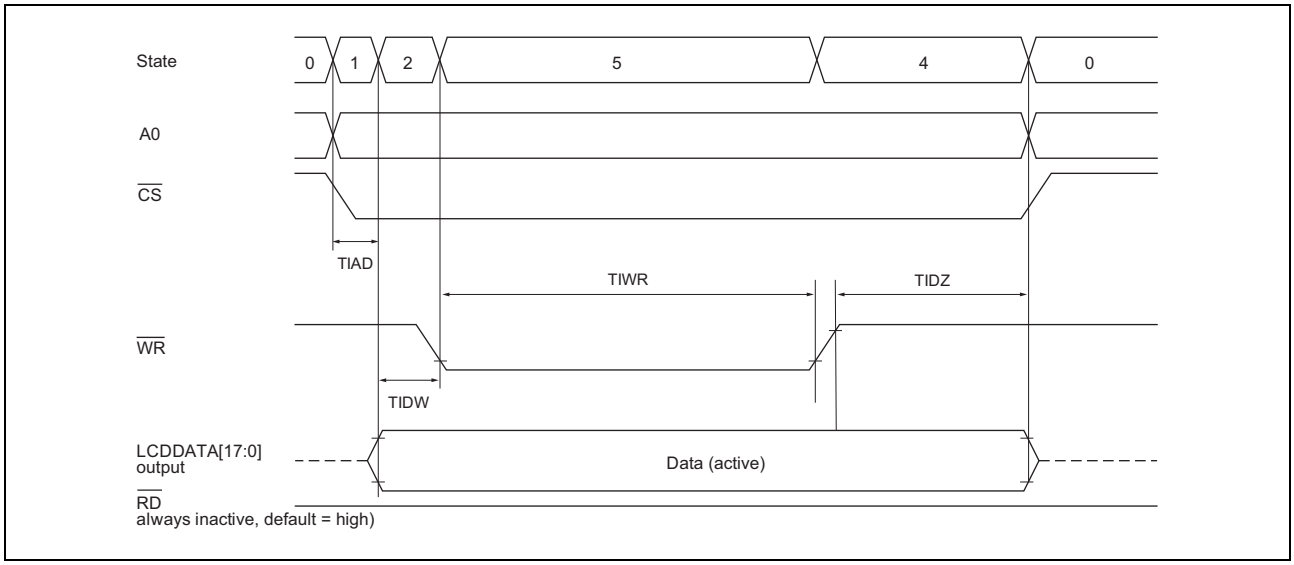


Figure 36.3 Timing diagram of a write operation in RAM access mode

$\overline{CS}$ is active while the LCD Bus Interface is active. When the write strobe $\overline{WR}$ is active, data is written to the LCD bus. The read strobe $\overline{RD}$ is inactive.

(2) General timing in E-Type mode

The data transmission in E-Type mode is split into the following states:

Table 36.8 Data transmission states in E-Type mode

State	Control bit	State number*1
No activity between two transmissions	–	0
Waiting for address	LCBInTMAD	1
Activating read/write direction	LCBInTMDW	2
Width of the read/write strobe	LCBInTMED	3
End of the bus cycle	LCBInTMDZ	4

Note 1. The state number refers to the state line in the following E-Type mode diagrams.

Control signals level

The following figures assume following settings of the control signals levels:

- LCBInOUTLEV.LCBInTLCS = 0: $\overline{CS}$ is active-low
- LCBInOUTLEV.LCBInTLRDE = 0: $\overline{E}$ is active-low
- LCBInOUTLEV.LCBInTLWR = 0: RW is
  - active-high for read
  - active-low for write
- LCBInOUTLEV.LCBInTLA0 is set as required to indicate transmission of control or data information

The following figure shows the general timing of a *read* operation in E-Type mode.

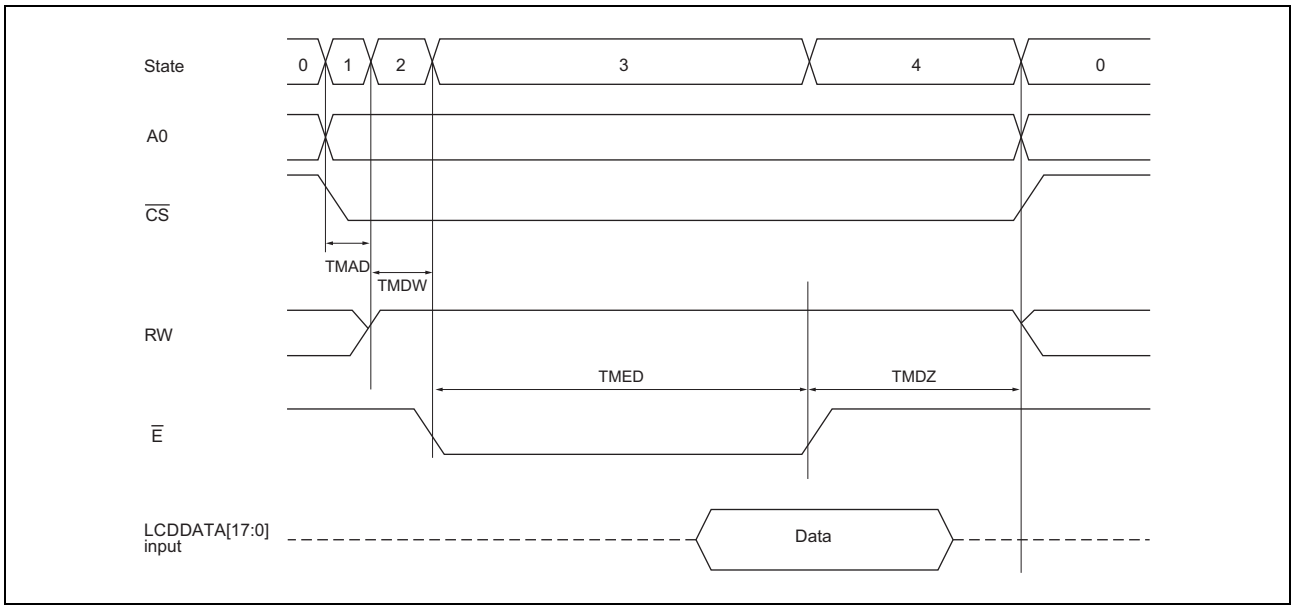


Figure 36.4 Timing diagram of a read operation in E-Type mode

$\overline{CS}$ is active while the LCD Bus Interface is active. When the read/write signal RW has high level and the $\overline{E}$ strobe is active, data is read from the LCD bus.

The following figure shows the general timing of a *write* operation in E-Type mode.

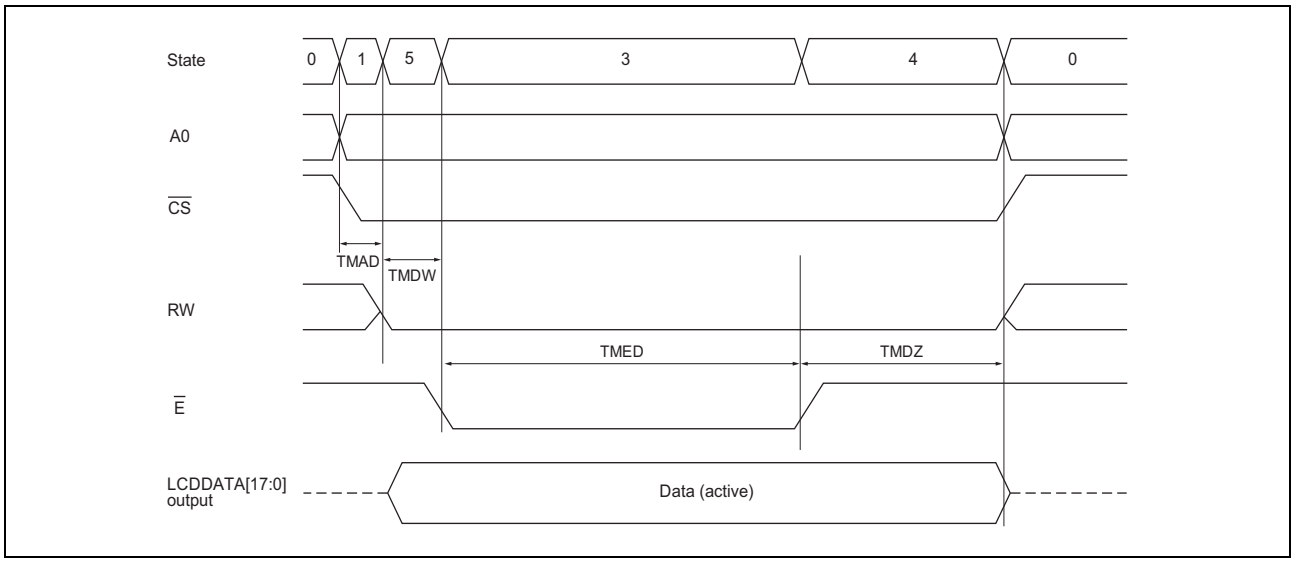


Figure 36.5 Timing diagram of a write operation in E-Type mode

$\overline{CS}$ is active while the LCD Bus Interface is active. When the read/write signal RW has low level and the $\overline{E}$ strobe is active, data is written to the LCD bus.

36.3.1.4 Status flags

The following status information is provided:

- Data transfer active (LCBInSTATUS.LCBInTCIDLE)
- Change of control registers possible (LCBInSTATUS.LCBInTCLOCK)

  Control registers can only be changed while no data transfer is active.
- Reading transfer pending (LCBInSTATUS.LCBInTCRPG)

36.3.2 TFT operation mode

In TFT operation mode, video data is sent to the LCD bus. One bus cycle contains the data of one pixel (e.g. RGB value of a pixel). One picture frame comprises pixels in a configurable number of lines and pixels per line.

Control signals are generated to indicate the start of a new pixel line ($\overline{HSYNC}$) or the start of a new picture frame ($\overline{VSYNC}$).

The pixel clock specifies the transfer speed of single pixels within a line.

Wait states can be specified, for example at the end of a line or a picture frame.

Write buffer

To sustain a constant frame rate on the TFT LCD, the CPU or DMA has to provide enough data for the write buffer. The write buffer generates interrupts to warn if specified states are reached. For details see Section 36.3.4, Write buffer details.

The TFT operation mode is automatically stopped (LCBInOPMODE.LCBInOPMODE[1:0] = 00_B), if there is a write buffer underrun.

36.3.2.1 Access types

The following access types are possible:

- Write access

  Read access is not possible in TFT operation mode.
- Display data (pixel data)
- Byte, halfword, or word access

Write access to the LCD bus is performed by writing data to one of the data write registers LCBInDAT0CWR or LCBInDAT0SWR. The data is transferred to the write buffer, which operates using a first-in first-out (FIFO) principle. It is converted by the CLUT, if enabled (LCBInTCONTROL.LCBInDTCC = 1), and written to the LCD bus.

Data is continuously fetched from the write buffer. If the write buffer runs empty (LCDIFTIEMPT), the LCD Bus Interface stops. When new data is written to the write buffer, the LCD Bus Interface can be restarted with a new picture frame.

Depending on the LCD bus width, a single write operation can contain one or more pixel data values on the LCD bus.

For example, writing the data register LCBInDAT0SWRH writes 16 bits of display data to the LCD bus. If the LCD bus width is 8 bits, data for two pixels is output.

The DE control line is active while display data is transmitted.

NOTE

Reading of data read registers LCBInDAT0CRD, LCBInDAT0SRD, or LCBInDATRDW is not allowed in TFT operation mode.

36.3.2.2 Transfer speed

The transfer speed depends on the pixel clock, the operation clock and the selected wait states. The following control registers are available:

- LCBInCKSEL

  This register sets the operation clock from PCLK, PCLK/2, PCLK/4, ..., PCLK/64.
- LCBInTFTPRS

  This register is used to specify a prescaler to derive the pixel clock from the operation clock.
- LCBInTFTCYC0, LCBInTFTCYC1, and LCBInTFTCYC2

  These registers define the number of pixel clock or operation clock cycles for one bus cycle. Wait states for different states within one bus cycle can be specified. For example, the back porch time.

After reset, the slowest clock is selected as default.

Pixel clock

The pixel clock can be calculated as follows:

Pixel clock = operation clock / (LCBInTFTPCPRS[15:0] + 1)

36.3.2.3 General timing

The timing depends on the number of lines per picture frame (LCBInTFTCYC1.LCBInTFNVC[9:0]) and the number of pixels per line (LCBInTFTCYC2.LCBInTFNHC[9:0]).

The transmission of a picture frame is split into the following states:

Table 36.9 States in transmission of a picture frame

State				Control bit*1	State number*2
Wait state between picture frames				LCBInTFDHV[7:0]	7
Vertical frame	VSYNC pulse width (new picture frame start)			LCBInTFNHVS[7:0]	1
	Vertical back porch time			LCBInTFNVB[7:0]	2
	Horizontal frame	Wait state between lines		LCBInTFDCH[7:0]	
		HSYNC pulse width (new line start)		LCBInTFNCHS[7:0]	3
		Horizontal back porch time		LCBInTFNHB[7:0]	4
		Pixel	Wait for pixel clock to become active	LCBInTFDCD[6:0]	5
			Pixel clock active time (transfer of one pixel)	LCBInTFDCW[7:0]	
		Horizontal front porch time		LCBInTFNHE[5:0]	6
	Vertical front porch time			LCBInTFNVE[5:0]	7

Note 1. Control bits starting with “LCBInTFD” give the wait states based on operation clock cycles. Control bits starting with “LCBInTFN” give the wait states based on pixel clock cycles.

Note 2. The state number refers to the state line in the following diagrams.

Control signals level

The following figures assume following settings of the control signals levels:

- LCBInOUTLEV.LCBInTLVS = 0: $\overline{\text{VSYNC}}$ is active-low
- LCBInOUTLEV.LCBInTLHS = 0: $\overline{\text{HSYNC}}$ is active-low
- LCBInOUTLEV.LCBInTLDE = 1: DE is active-high

The following figure shows the general timing in TFT operation mode.

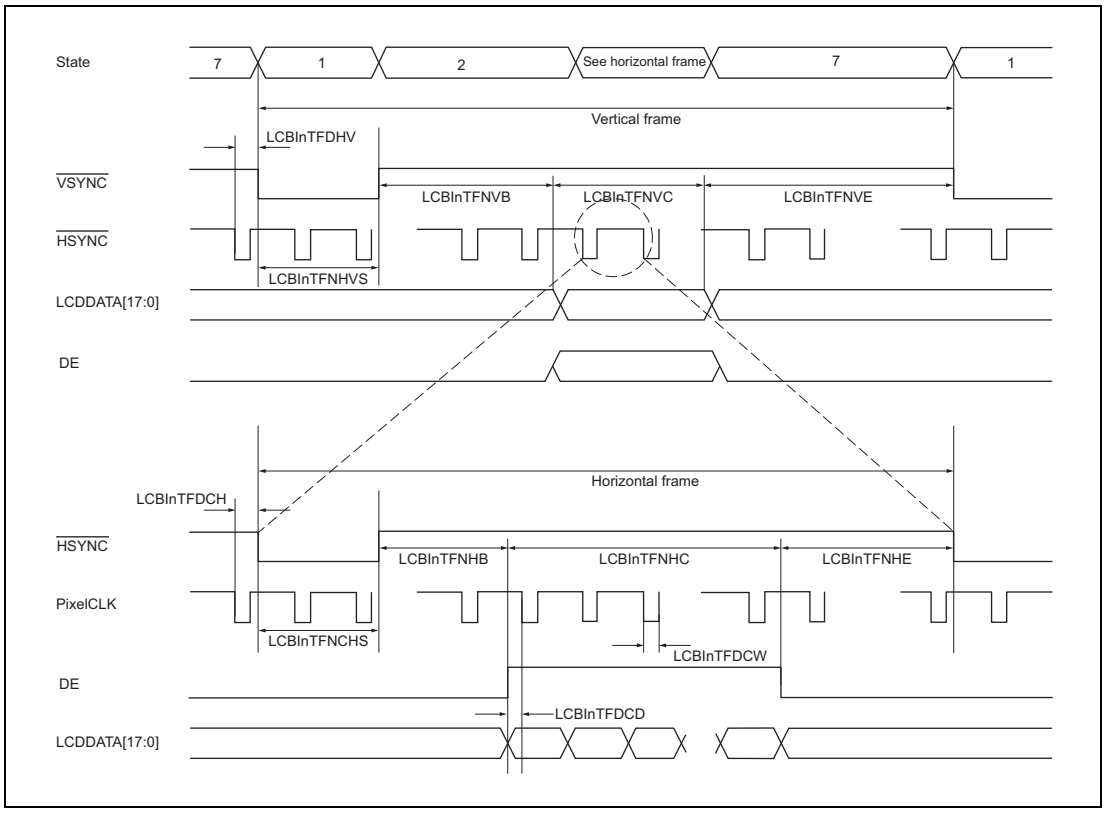


Figure 36.6 TFT operation mode general timing

The timing diagram above shows the following:

- One vertical frame (corresponding to a full image) consists of several horizontal frames (corresponding to a pixel line each)
- The signal DE is active when pixel data is transferred

36.3.2.4 Status flags

The following status information is provided:

- Change of control registers possible (LCBInSTATUS.LCBInTCLOCK)
- Control registers can only be changed during back porch time of vertical synchronization.

36.3.3 Color usage

To support color displays, the LCD Bus Interface can convert write data to RGB values using a Color Lookup Table (CLUT). The data from the write buffer is used as an address to index and select a RGB value. CLUT usage is enabled by control bit LCBInTCONTROL.LCBInDTCC.

The colors are defined in the CLUT data RAM. The CLUT data RAM has a size of 256 x 18 bits. This size is suitable for 18-bit and 9-bit LCD bus widths:

- LCD bus width = 18 bits: One 18-bit color definition per CLUT address
- LCD bus width = 9 bits: Two consecutive 9-bit bus accesses for the 18-bit color definition per CLUT address

A dedicated mode is provided for writing the CLUT data RAM.

CLUT palette modes

There are two CLUT palette modes:

- 8-bit palette mode

  1 color palette with 256 colors.
  For each color, an 8-bit data value is required.

- 4-bit palette mode

  16 color palettes each with 16 colors.
  The currently active palette is selected by the offset address in register LCBInCLUTOFFS.
  For each color, a 4-bit data value is required.

The CLUT palette mode is selected by bit LCBInTCONTROL.LCBInDTCP.

The following figure illustrates the difference between the two CLUT palette modes. For details refer to Section 36.3.5, Data transfer types.

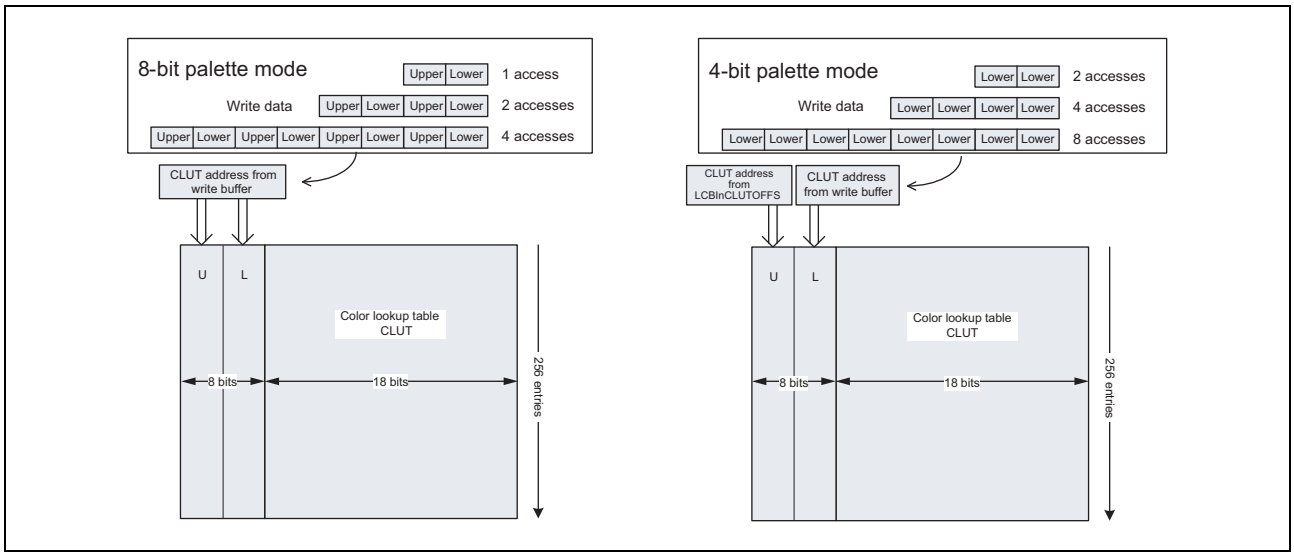


Figure 36.7 CLUT addressing

NOTE

When CLUT usage is enabled in non-TFT operation mode, the CLUT is nevertheless ignored in the following cases:

- When control data is written (writing to registers LCBInDAT0SWR)
- When data is read

CLUT definition mode

The CLUT color values are defined in CLUT definition mode. When this mode is selected (LCBInOPMODE.LCBInOPMODE[1:0] = 11_B), CLUT data values can be read and written via the CLUT data RAM registers LCBInCLUTDmW (m = 0 to 255).

During CLUT definition mode

- no data can be written to the write buffer
- no data is transferred via the LCD Bus Interface
- no TFT control signals ($\overline{\text{HSYNC}}$, $\overline{\text{VSYNC}}$, DE, PixelCLK) are generated.

36.3.4 Write buffer details

The write buffer is a FIFO buffer to store data that is to be written to the LCD bus. It optimizes the data transfer on the LCD bus. This means, that data is fetched from the write buffer while a transfer is still active on the LCD bus.

36.3.4.1 Size and content

The write buffer has a size of 12 x (32 + 3) bits. It can hold the following data for up to 12 write accesses:

- 32 data bits
- 2 bits for access size specification (8 bit, 16 bit, or 32 bit)
- 1 bit to hold the A0 setting

Example

When data is written to register LCBInDAT0SWRH, the write buffer holds the data plus a 2-bit code for 16-bit access and a 1-bit code for “A0 is set”.

36.3.4.2 Write buffer fill state interrupts

For every write buffer fill state, except overflow and underrun, an interrupt request is generated. All fill states are reflected in the LCBInIRQPEN register. This is important in TFT operation mode, because there have to be enough values in the FIFO to provide a constant frame rate.

Table 36.10 Write buffer fill states (1/2)

Fill state	Number of data in write buffer	Comments	Interrupt status flag
Empty	1→ 0	-	LCDIFTIEMPT LCBInIRQPEN.LCBInRWIRQE
Half full	5 → 6 7 → 6	-	LCDIFTIHALF LCBInIRQPEN.LCBInRWIRQH
Quarter full	4 → 3	-	LCDIFTIQTR LCBInIRQPEN.LCBInRWIRQQ
Three quarters full	8 → 9	-	LCDIFTI3QTR LCBInIRQPEN.LCBInRWIRQT

Table 36.10 Write buffer fill states (2/2)

Fill state	Number of data in write buffer	Comments	Interrupt status flag
Full	11→ 12	-	LCDIFTIFULL LCBInIRQPEN.LCBInRWIRQF
Overflow	12	A value is written to the write buffer while it is full. The new value is discarded.	LCBInIRQPEN.LCBInRWOVF
Underrun*1	0	• Non-TFT operation mode: No underrun can occur. LCD Bus Interface is stopped when write buffer is empty. • TFT operation mode: Data is fetched from the write buffer while it is empty. LCD Bus Interface is stopped. Video output is stopped.	LCBInIRQPEN.LCBInRWUNR

Note 1. Refer also to “Write buffer prefetch” below.

CAUTION

After the first data has been written to an empty buffer a write buffer empty interrupt LCDIFTIEMPT is generated, since this first data is immediately transferred to the interface.

Write buffer prefetch

While data is output via the LCD bus I/F the next data word is already fetched from the write buffer. This means in particular during output of the last data an underrun occurs because of the prefetch attempt of the next - but not existing - data.
Consequently the LCD bus I/F operation is stopped.

CAUTION

For continues data output via the LCD bus I/F make sure to have at least one data word in the write buffer to avoid an LCD bus I/F stop because of an underrun.

36.3.5 Data transfer types

Depending on the access type and the LCD bus width, bit shifts, and CLUT usage, one or more transfers are performed on the LCD Bus Interface:

- Access type

  Access to the data registers can be performed as:
  - Byte access (8 bit)
  - Halfword access (16 bit)
  - Word access (32 bit)

- CLUT usage

  When data is converted by the CLUT, the number of bits to address a color value in the CLUT is specified by the CLUT palette mode:

  – In 4 bit palette mode, 4 address bits are required.

  – In 8 bit palette mode, 8 address bits are required.

  18 or 2 x 9 data bits are transferred on the LCD bus.

- LCD bus width

  – Without CLUT usage, 8-bit, 9-bit, 16-bit, and 18-bit bus widths are supported.

  – With CLUT usage, 9-bit and 18-bit bus widths are supported.

  The desired LCD bus width is selected by control bits LCBInTCONTROL.LCBInDTCM[1:0].

- Bit shifts

  On the LCD bus, data can be shifted.
  For example, 8 bits can be written to bit positions [7:0], [8:1], [9:2], or [17:10]. The bit shift is specified by control bits LCBInTCONTROL.LCBInDTCS[1:0].

NOTES

1. Every access must address the base address of the data registers. Access to the individual bytes within the registers is prohibited.
2. Data transfers with CLUT usage are only allowed for 9 bit and 18 bit LCD busses.

The following sections show the different data transfer types, depending on the above mentioned settings.

36.3.5.1 Transfers without CLUT usage on any LCD bus width

The following table summarizes the number of transfers on the LCD bus, depending on the LCD bus width and the access type for data transfers without CLUT usage (LCBInTCONTROL.LCBInDTCC = 0).

Table 36.11 Number of data transfers without CLUT usage

LCD bus width (LCBInDTCM[1:0])	Number of data transfers on LCD bus		
	Byte access	Halfword access	Word access
8 bit	1	2	4
9 bit	-	1	2
16 bit	-	1	2
18 bit	-	-	1

The following figure illustrates the data transfer types.

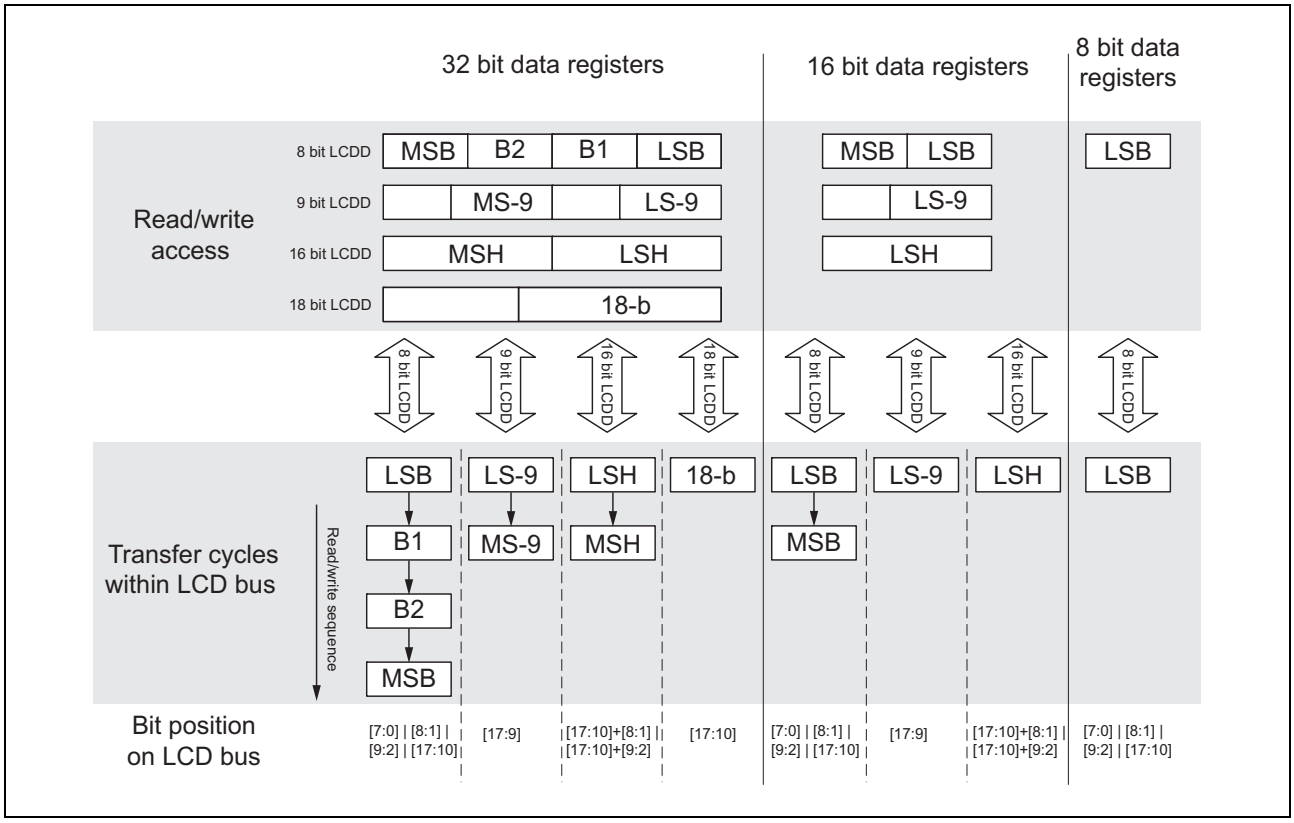


Figure 36.8 Data transfer types without CLUT usage

36.3.5.2 Data transfers with CLUT usage on 18-bit LCD bus

When CLUT usage is enabled (LCBInTCONTROL.LCBInDTCC = 1), every data access is converted to one or more 18-bit color values.

The following table summarizes the number of transfers with CLUT usage on an 18-bit LCD bus, depending on the CLUT palette mode and the access type for data transfers.

NOTE

When CLUT usage is enabled in non-TFT operation mode, control data write operations and read operations are performed without CLUT conversion.

Table 36.12 Number of data transfers with CLUT usage on 18-bit LCD bus

CLUT palette mode (LCBInDTCP)	Number of data transfers on LCD bus		
	Byte access	Halfword access	Word access
8 bit	1	2	4
4 bit	2	4	8

Example

In 4 bit palette mode, 4 color values can be addressed with a halfword access. Every color value requires one 18 bit data transfer on the LCD bus, so that a total of 4 data transfers is performed.

The following figure illustrates the data transfer types.

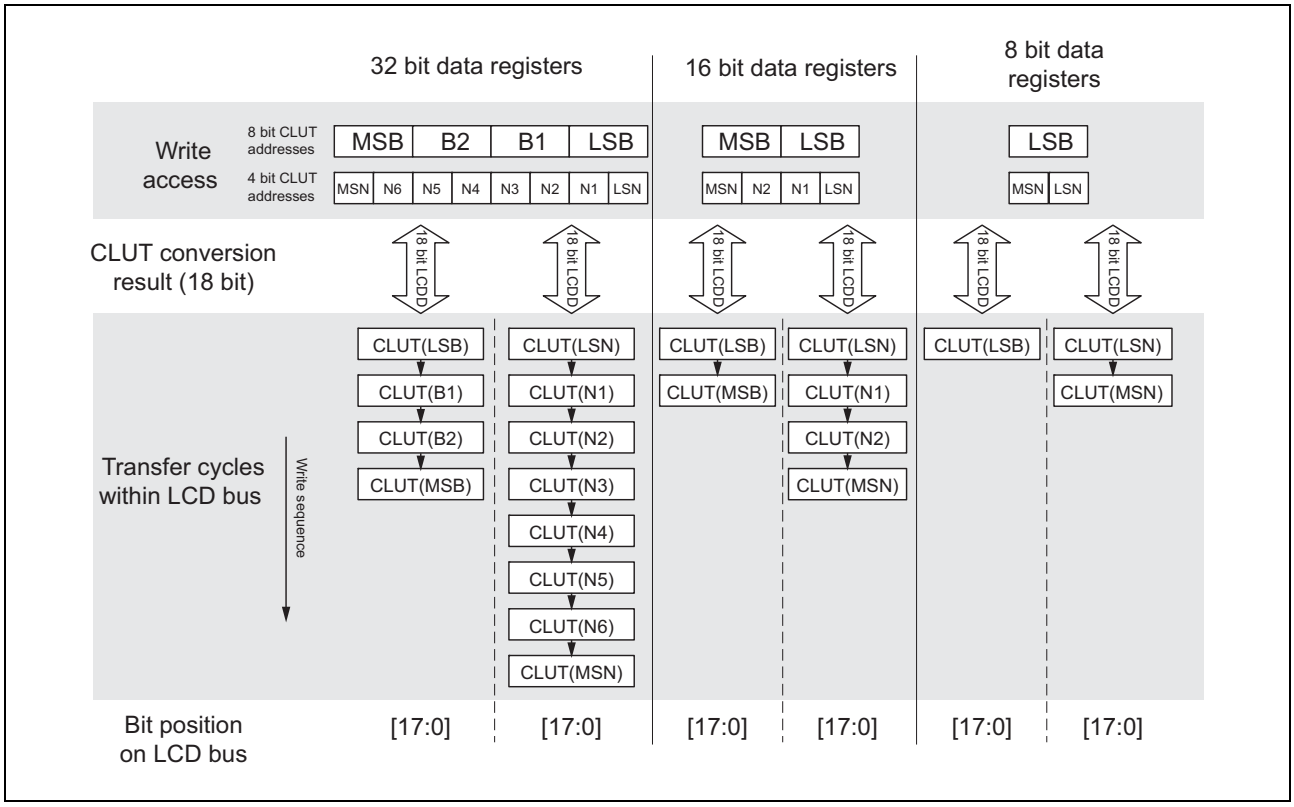


Figure 36.9 Data transfer types with CLUT usage on 18-bit LCD bus

CLUT conversion starts with the LSB and ends with the MSB.

LSB is output first.

36.3.5.3 Data transfers with CLUT usage on 9-bit LCD bus

When CLUT usage is enabled (LCBInTCONTROL.LCBInDTCC = 1), every data access is converted to one or more 18-bit color values.

The following table summarizes the number of transfers with CLUT usage on a 9-bit LCD bus, depending on the CLUT palette mode and the access type for data transfers.

NOTE

When CLUT usage is enabled in non-TFT operation mode, control data write operations and read operations are performed without CLUT conversion.

Table 36.13 Number of data transfers with CLUT usage on 9-bit LCD bus

CLUT palette mode (LCBInDTCP)	Number of data transfers on LCD bus		
	Byte access	Halfword access	Word access
8 bit	2	4	8
4 bit	4	8	16

Example

In 4 bit palette mode, 4 color values can be addressed with a halfword access. Every color value requires two 9 bit data transfer on the LCD bus, so that a total of 8 data transfers is required.

The following figure illustrates the data transfer types.

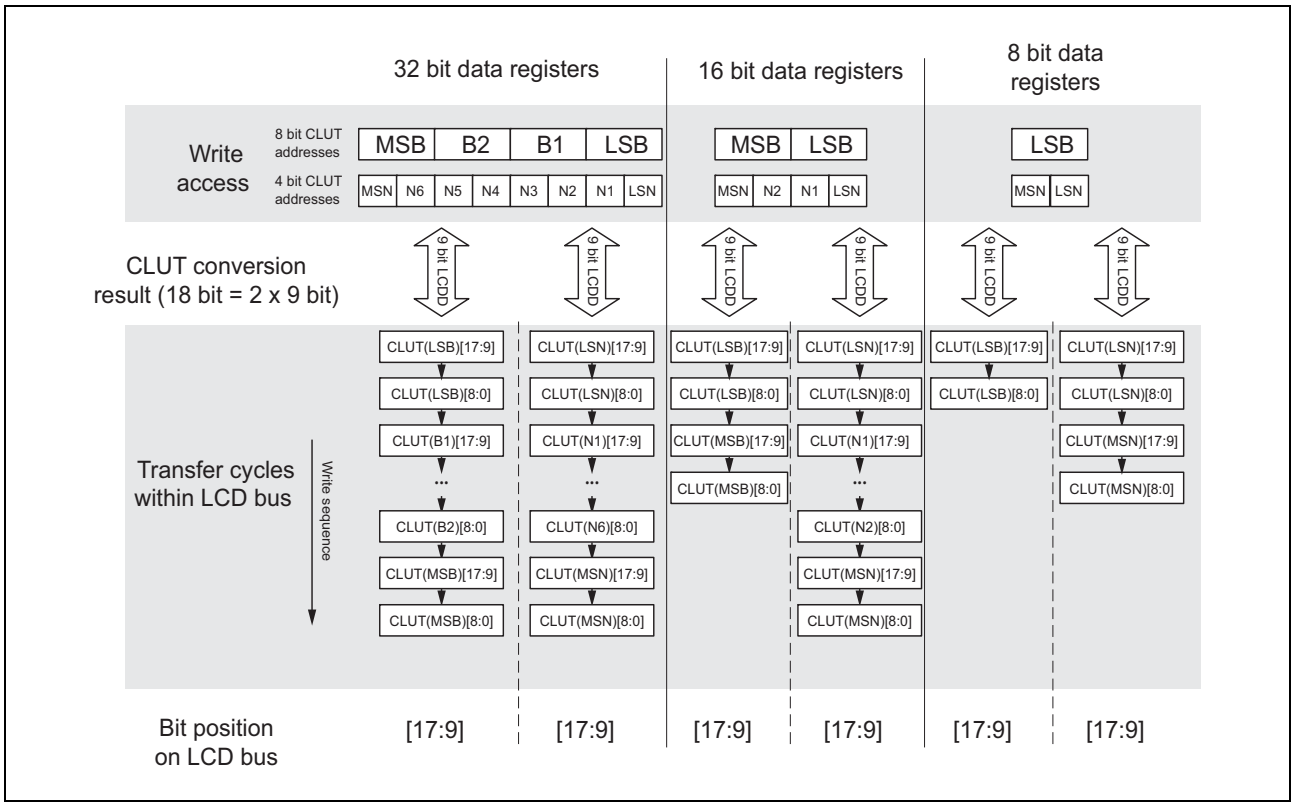


Figure 36.10 Data transfer types with CLUT usage on 9-bit LCD bus

CLUT conversion starts with the LSB and ends with the MSB.

LSB of the data is output first, but the upper 9 bits of the CLUT output are transmitted first on the bus.

36.3.6 Soft reset

The LCD Bus Interface can be reset by a soft reset (LCBInSRESET.LCBInSRST).

When the LCD Bus Interface is reset by software, all active transfers are stopped, pending transfers are deleted and the write buffer is cleared.

Note that not all control registers are reset by a soft reset.

36.3.7 Change control registers settings

To change the control register settings, the following procedure must be considered:

1. Stop the LCD Bus Interface (LCBInOPMODE.LCBInOPMODE[1:0] = 00_B).
   The write buffer is cleared.
2. Read the LCBInOPMODE register to ensure that the LCD Bus Interface is stopped (LCBInOPMODE.LCBInOPMODE[1:0] = 00_B). If unsuccessful, repeat writing to the LCBInOPMODE register.
3. Change any of the following control register settings.
   – LCBInCKSEL (operation clock setting)
   – LCBInTCONTROL (specification of CLUT usage, CLUT palette mode, LCD bus width, and bit shifting on LCD bus)
   – LCBInBCYCT and LCBInBCYC (Bus cycle timing settings for non-TFT operation

mode)
- LCBInTFTPRS (pixel clock setting for TFT operation mode)
- LCBInTFTCYC0, LCBInTFTCYC1, LCBInTFTCYC2 (Bus cycle timing settings for TFT operation mode)
- LCBInOUTLEV (output level specification of control signals)

4. Set the desired operation mode to restart the LCD Bus Interface (LCBInOPMODE.LCBInOPMODE[1:0]).

36.4 Registers

This section contains a description of all registers of the LCD Bus Interface.

36.4.1 LCD Bus Interface registers overview

The LCD Bus Interface is controlled and operated by the following registers:

Table 36.14 Control registers overview

Register name	Shortcut	Address
General control registers		
Clock selection control register	LCBInCKSEL	<LCBIn_base> + 440_H
Operation mode control register	LCBInOPMODE	<LCBIn_base> + $44C_H$
Data transfer control register	LCBInTCONTROL	<LCBIn_base> + 454_H
CLUT offset control register	LCBInCLUTOFFS	<LCBIn_base> + 450_H
Output level control register	LCBInOUTLEV	<LCBIn_base> + 470_H
Status indication register	LCBInSTATUS	<LCBIn_base> + 474_H
Pending interrupt and FIFO error register	LCBInIRQPEN	<LCBIn_base> + $47C_H$
Pending interrupt clear register	LCBInIRQCLR	<LCBIn_base> + 480_H
Soft reset control register	LCBInSRESET	<LCBIn_base> + 448_H
Emulation register	LCBInEMU	<LCBIn_base> + 444_H
Non-TFT mode control registers		
Bus cycle type non-TFT control register	LCBInBCYCT	<LCBIn_base> + 458_H
Non-TFT modes bus cycle specification control register	LCBInBCYC	<LCBIn_base> + $45C_H$
TFT mode control registers		
TFT cycle prescaler control register	LCBInTFTPRS	<LCBIn_base> + 460_H
TFT cycle timing specification control register 0	LCBInTFTCYC0	<LCBIn_base> + 464_H
TFT cycle timing specification control register 1	LCBInTFTCYC1	<LCBIn_base> + 468_H
TFT cycle timing specification control register 2	LCBInTFTCYC2	<LCBIn_base> + $46C_H$

Table 36.15 Data registers overview

Register name	Shortcut	Address
Data write registers	LCBInDAT0SWR	<LCBIn_base> + 400_H
	LCBInDAT0SWRH	<LCBIn_base> + 404_H
	LCBInDAT0SWRW	<LCBIn_base> + 408_H
	LCBInDAT0CWR	<LCBIn_base> + $40C_H$
	LCBInDAT0CWRH	<LCBIn_base> + 410_H
	LCBInDAT0CWRW	<LCBIn_base> + 414_H
Data read registers	LCBInDAT0SRD	<LCBIn_base> + 418_H
	LCBInDAT0SRDH	<LCBIn_base> + $41C_H$
	LCBInDAT0SRDW	<LCBIn_base> + 420_H
	LCBInDAT0CRD	<LCBIn_base> + 424_H
	LCBInDAT0CRDH	<LCBIn_base> + 428_H
	LCBInDAT0CRDW	<LCBIn_base> + $42C_H$
	LCBInDATRDW	<LCBIn_base> + 430_H

Table 36.16 CLUT data RAM registers overview

Register name	CLUT data area	Address
CLUT data RAM registers	LCBInCLUTD0W	<LCBIn_base>
	LCBInCLUTD1W	<LCBIn_base> + 4_H
	LCBInCLUTD2W	<LCBIn_base> + 8_H
	...	...
	LCBInCLUTD254W	<LCBIn_base> + $3F8_H$
	LCBInCLUTD255W	<LCBIn_base> + $3FC_H$

<LCBIn_base>

The base addresses <LCBIn_base> of the LCBIn is defined in the first section of this chapter under the key word “Register addresses”.

36.4.2 LCD Bus Interface general control registers details

36.4.2.1 LCBInCKSEL - Clock selection control register

This register selects the prescaler for the LCD Bus Interface operation clock.

Access: This register can be accessed in 32-bit units.
The LCD Bus Interface must be stopped (LCBInOPMODE.LCBInOPMODE[1:0] = 00_B) before writing to this register. For details refer to Section 36.3.7, Change control registers settings.

Address: <LCBIn_base> + 440_H

Initial Value: 0000 0006_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	LCBInCKS[2:0]		
R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 36.17 LCBInCKSEL register contents

Bit position	Bit name	Function
31 to 3	Reserved	When read, the value after reset is returned. When written, write the value after reset.
2 to 0	LCBIn CKS[2:0]	Selects the prescaler for the operation clock.

CKS2	CKS1	CKS0	Operation clock
0	0	0	PCLK/1
0	0	1	PCLK/2
0	1	0	PCLK/4
0	1	1	PCLK/8
1	0	0	PCLK/16
1	0	1	PCLK/32
1	1	0	PCLK/64
1	1	1	Not allowed

36.4.2.2 LCBInOPMODE - Operation mode control register

This register selects the operation mode for the LCD Bus Interface.

Access: This register can be accessed in 32-bit units.
If a new value is written to this register while control registers are locked (LCBInSTATUS.LCBInTCLOCK = 1) the new value does not take effect until the control registers are unlocked (LCBInSTATUS.LCBInTCLOCK = 0).

Address: <LCBIn_base> + $44C_H$

Initial Value: $0000\ 0000_H$. This register is initialized by a LCBI H/W reset and by soft reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	LCBInOPM FAST	0	0	LCBIn OPMODE [1:0]	
R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W

Table 36.18 LCBInOPMODE register contents

Bit position	Bit name	Function
31 to 5	Reserved	When read, the value after reset is returned. When written, write the value after reset.
4	LCBIn OPMFAST	Enables fast bus cycle mode: 0: Fast bus cycle mode disabled (slow bus cycle mode enabled) 1: Fast bus cycle mode enabled (slow bus cycle mode disabled) This bit only applies in non-TFT operation mode.
3 to 2	Reserved	When read, the value after reset is returned. When written, write the value after reset.
1 to 0	LCBIn OPMODE[1:0]	Specifies the operation mode.

LCBIn OPMODE1	LCBIn OPMODE0	Description
0	0	LCD Bus Interface is stopped The write buffer is cleared
0	1	Non-TFT operation mode
1	0	TFT operation mode
1	1	CLUT definition mode

36.4.2.3 LCBInTCONTROL - Data transfer control register

This register specifies the LCD bus width, the bit shifts on the LCD bus, CLUT usage, and CLUT palette mode.

Access: This register can be accessed in 32-bit units.
The LCD Bus Interface must be stopped (LCBInOPMODE.LCBInOPMODE[1:0] = 00_B) before writing to this register. For details refer to Section 36.3.7, Change control registers settings.

Address: <LCBIn_base> + 454_H

Initial Value: $0000\ 0000_H$. This register is initialized by a LCBI H/W reset and by soft reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	LCBIn DTCS[1:0]		LCBIn DTCM[1:0]		0	0	LCBIn DTCP	LCBIn DTCC
R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Table 36.19 LCBInTCONTROL register contents

Bit position	Bit name	Function
31 to 8	Reserved	When read, the value after reset is returned. When written, write the value after reset.
7 to 6	LCBIn DTCS[1:0]	Specifies the bit shifts of data on the LCD bus. These bits only apply for 8-bit and 16-bit bus width. Refer to Table 36.20, Bit shifts on the LCD bus.
5 to 4	LCBIn DTCM[1:0]	Selects the LCD bus width. **LCBIn DTCM1 / LCBIn DTCM0 / LCD bus width** 0 / 0 / 8 bit 0 / 1 / 16 bit 1 / 0 / 9 bit 1 / 1 / 18 bit This setting is only supported with 32-bit accesses. When CLUT usage is enabled (LCBInTCONTROL.LCBInDTCC = 1), only 9-bit or 18-bit bus widths are allowed. Refer to Section 36.3.5, Data transfer types.
3 to 2	Reserved	When read, the value after reset is returned. When written, write the value after reset.
1	LCBInDTCP	Specifies the CLUT palette mode: 0: 8-bit palette mode 1: 4-bit palette mode This bit only applies when LCBInTCONTROL.LCBInDTCC = 1.
0	LCBInDTCC	Enables/disables CLUT usage: 0: CLUT usage disabled 1: CLUT usage enabled For details refer to Section 36.3.3, Color usage.

Table 36.20 Bit shifts on the LCD bus

LCBInDTCM[1:0]	LCBInDTCS[1:0]	LCD bus width	Bit position of data on the LCD bus
00	00	8 bit	[7:0]
	01		[8:1]
	10		[9:2]
	11		[17:10]
01	00	16 bit	Setting prohibited
	01		[17:10] and [8:1]
	10		[17:10] and [9:2]
	11		Setting prohibited
10	X	9 bit	[17:9]
11		18 bit	[17:0]

36.4.2.4 LCBInCLUTOFFS - CLUT offset control register

This register selects one of the 16 alternative color palettes each with 16 colors. It specifies the 4 bit offset address of the color palette.

This register only applies when CLUT usage is selected (LCBInTCONTROL.LCBInDTCC = 1) and 4 bit palette mode is selected (LCBInTCONTROL.LCBInDTCP = 1).

Access: This register can be accessed in 32-bit units.

Address: <LCBIn_base> + 450_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	LCBInCLTOFFS[7:4]				0	0	0	0
R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R

Table 36.21 LCBInCLUTOFFS register contents

Bit position	Bit name	Function
31 to 8	Reserved	When read, the value after reset is returned. When written, write the value after reset.
7 to 4	LCBIn CLTOFFS[7:4]	Specifies the offset address of color palette in 4-bit palette mode
3 to 0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

36.4.2.5 LCBInOUTLEV - Output level control register

This register selects the output level of the LCD bus control signals. The default level is active-low.

Access: This register can be accessed in 32-bit units.
The LCD Bus Interface must be stopped (LCBInOPMODE.LCBInOPMODE[1:0] = 00_B) before writing to this register. For details refer to Section 36.3.7, Change control registers settings.

Address: <LCBIn_base> + 470_H

Initial Value: 0000 0000_H. This register is initialized by a LCBI H/W reset and by soft reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	LCBIn TLDE	LCBIn TLVS	LCBIn TLHS	0	0	0	0	LCBIn TLRDE	LCBIn TLWR	LCBIn TLCS	LCBIn TLA0
R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 36.22 LCBInOUTLEV register contents (1/2)

Bit position	Bit name	Function
31 to 11	Reserved	When read, the value after reset is returned. When written, write the value after reset.
10	LCBInTLDE	Specifies the active level of the data enable signal DE in TFT mode, that indicates valid data: 0: DE is active-low 1: DE is active-high **Note:** In non-TFT mode, LCBInTLDE has no meaning.
9	LCBInTLVS	Specifies the active level of the vertical synchronization signal $\overline{\text{VSYNC}}$ in TFT mode, that indicates new frame: 0: VSYNC is active-low 1: VSYNC is active-high **Note:** In non-TFT mode, LCBInTLVS has no meaning.
8	LCBInTLHS	Specifies the active level of the horizontal synchronization signal $\overline{\text{HSYNC}}$ in TFT mode, that indicates new line: 0: $\overline{\text{HSYNC}}$ is active-low 1: HSYNC is active-high **Note:** In non-TFT mode, LCBInTLHS has no meaning.
7 to 4	Reserved	When read, the value after reset is returned. When written, write the value after reset.
3	LCBInTLRDE	Specifies the active level of the read strobe $\overline{\text{RD}}$ or the E strobe in non-TFT mode: • $\overline{\text{RD}}$ strobe in RAM mode: 0: $\overline{\text{RD}}$ strobe active-low 1: RD strobe active-high • E strobe in E-Type mode: 0: $\overline{\text{E}}$ strobe active-low 1: E strobe active-high **Note:** In TFT mode, LCBInTLRDE has no meaning.

Table 36.22 LCBInOUTLEV register contents (2/2)

Bit position	Bit name	Function
2	LCBInTLWR	Specifies the active level of the write strobe $\overline{WR}$ or the RW signal in non-TFT mode: • $\overline{WR}$ strobe in RAM mode: 0: $\overline{WR}$ strobe active-low 1: WR strobe active-high • RW signal in E-Type mode: 0: RW is – active-high for read – active-low for write 1: RW is – active-low for read – active-high for write **Note:** In TFT mode, LCBInTLWR has no meaning.
1	LCBInTLCS	Specifies the active level of the chip select signal $\overline{CS}$ in non-TFT mode, that indicates transfer enabled: 0: $\overline{CS}$ is active-low 1: CS is active-high **Note:** In TFT mode, LCBInTLCS has no meaning.
0	LCBInTLA0	Specifies the active level of the A0 control signal in non-TFT mode: 0: A0 – = 0 indicates control data – = 1 indicates display data 1: A0 – = 0 indicates display data – = 1 indicates control data **Note:** In TFT mode, LCBInTLA0 has no meaning.

36.4.2.6 LCBInSTATUS - Status indication register

This register indicates the status of pending read transfers, of the control register lock state, and of the idle state.

In TFT operation mode, only the register lock state is applied.

Access: This register can be accessed in 32-bit units.

Address: <LCBIn_base> + 474_H

Initial Value: $0000\ 0001_H$. This register is initialized by a LCBI H/W reset and by soft reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	LCBInTCRPG	LCBInTCLOCK	LCBInTCIDLE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.23 LCBInSTATUS register contents

Bit position	Bit name	Function
31 to 3	Reserved	When read, the value after reset is returned. When written, write the value after reset.
2	LCBIn TCRPG	Indicates whether a read transfer is pending. 0: No read transfer is pending 1: Read transfer is pending This bit is only valid in non-TFT operation mode.
1	LCBIn TCLOCK	Indicates the status of the control register locking: 0: Control registers are *not* locked and *can* be changed 1: Control registers are locked and *cannot* be changed
0	LCBIn TCIDLE	Indicates the idle state in non-TFT operation mode: 0: Not idle Data transfer is currently active and/or pending. 1: Idle No data transfer is currently active and the write buffer empty. This bit is only applied in non-TFT operation mode. In TFT operation mode, this bit is always cleared.

36.4.2.7 LCBInIRQPEN - Pending interrupt and FIFO error register

This register indicates the pending interrupt requests due to certain write buffer fill states and when read data is available on the LCD bus.

The flags can be cleared by writing to register LCBInIRQCLR.

NOTE

The status of the bits of this register has no impact on the generation of an interrupt request. Thus even if a bit is not cleared, a new interrupt request will be generated, provided its condition is fulfilled.
If the interrupt requests are checked by polling of this register, the concerned bit must be cleared by writing to LCBInIRQCLR.

Access: This register can be accessed in 32-bit units.

Address: <LCBIn_base> + $47C_H$

Initial Value: $0000\ 0000_H$. This register is initialized by a LCBI H/W reset and by soft reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	LCBIn RW UNR	LCBIn RW OVF	0	0	LCBIn RW IRQT	LCBIn RW IRQQ	LCBIn RW IRQF	LCBIn RW IRQH	LCBIn RW IRQE	LCBIn TC IRQ
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.24 LCBInIRQPEN register contents (1/2)

Bit position	Bit name	Function
31 to 10	Reserved	When read, the value after reset is returned. When written, write the value after reset.
9	LCBIn RWUNR	Indicates the write buffer underrun state: 0: No underrun 1: Underrun occurred The LCD Bus Interface is stopped. This bit is only set in TFT operation mode.
8	LCBIn RWOVF	Indicates the write buffer overflow state: 0: No overflow 1: Overflow occurred The last value written to the write buffer is lost.
7 to 6	Reserved	When read, the value after reset is returned. When written, write the value after reset.
5	LCBIn RWIRQT	Indicates a three quarters filled write buffer (LCDIFTI3QTR interrupt request): 0: No interrupt request pending 1: Interrupt request pending
4	LCBIn RWIRQQ	Indicates a quarter filled write buffer (LCDIFTIQTR interrupt request): 0: No interrupt request pending 1: Interrupt request pending
3	LCBIn RWIRQF	Indicates a completely filled write buffer (LCDIFTIFULL interrupt request): 0: No interrupt request pending 1: Interrupt request pending

Table 36.24 LCBInIRQPEN register contents (2/2)

Bit position	Bit name	Function
2	LCBIn RWIRQH	Indicates a half filled write buffer (LCDIFTIHALF interrupt request): 0: No interrupt request pending 1: Interrupt request pending
1	LCBIn RWIRQE	Indicates an empty write buffer (LCDIFTIEMPT interrupt request): 0: No interrupt request pending 1: Interrupt request pending
0	LCBIn TCIRQ	Indicates that read data is available (LCDIFTIRDY interrupt request): 0: No interrupt request pending (no read data is available) 1: Interrupt request pending (read data is available)

36.4.2.8 LCBInIRQCLR - Pending interrupt and FIFO error clear register

This register clears the interrupt pending and FIFO error flags of register LCBInIRQPEN.

Access: This register can be accessed in 32-bit units.

Address: <LCBIn_base> + 480_H

Initial Value: $0000\ 0000_H$. This register is initialized by a LCBI H/W reset and by soft reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	LCBIn RW UNRC	LCBIn RW OVFC	0	0	LCBIn RW IRQTC	LCBIn RW IRQQC	LCBIn RW IRQFC	LCBIn RW IRQHC	LCBIn RW IRQEC	LCBIn TC IRQC
R	R	R	R	R	R	W	W	R	R	W	W	W	W	W	W

Table 36.25 IRQCLR register contents

Bit position	Bit name	Function
31 to 10	Reserved	When read, the value after reset is returned. When written, write the value after reset.
9	LCBIn RWUNRC	Clears the corresponding status flag in register LCBInIRQPEN: 0: No function 1: Clears the corresponding flag
8	LCBIn RWOVFC	
7 to 6	Reserved	When read, the value after reset is returned. When written, write the value after reset.
5	LCBIn RWIRQTC	Clears the corresponding status flag in register LCBInIRQPEN: 0: No function 1: Clears the corresponding flag
4	LCBIn RWIRQQC	
3	LCBIn RWIRQFC	
2	LCBIn RWIRQHC	
1	LCBIn RWIRQEC	
0	LCBIn TCIRQC	

36.4.2.9 LCBInSRESET - Soft reset control register

This register forces the stop of all active transfers, pending transfers are deleted, the write buffer is cleared, and the idle state is set (LCBInSTATUS.LCBInTCIDLE = 1).

Access: This register can be accessed in 32-bit units.
This register always reads as 0000 0000_H.

Address: <LCBIn_base> + 448_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LCBInSRST
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 36.26 LCBInSRESET register contents

Bit position	Bit name	Function
31 to 1	Reserved	When read, the value after reset is returned. When written, write the value after reset.
0	LCBInSRST	Triggers soft reset: 0: Soft reset is not triggered 1: Soft reset is triggered

36.4.2.10 LCBInEMU - Emulation register

This register controls whether the LCD Bus I/F can be stopped during emulation, for instance upon a breakpoint hit.

Access: This register can be accessed in 32-bit units.

Address: <LCBIn_base> + 444_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LCBIn SVSDIS
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 36.27 LCBInEMU register contents

Bit position	Bit name	Function
31 to 1	Reserved	When read, the value after reset is returned. When written, write the value after reset.
0	LCBIn SVSDIS	Emulation control 0: LCBI can be stopped during emulation 1: LCBI continuous operating during emulation

36.4.3 LCD Bus Interface non-TFT mode control registers details

36.4.3.1 LCBInBCYCT - Bus cycle type non-TFT control register

This register selects the access mode in non-TFT operation mode.

Access: This register can be accessed in 32-bit units.
The LCD Bus Interface must be stopped (LCBInOPMODE.LCBInOPMODE[1:0] = 00_B) before writing to this register. For details refer to Section 36.3.7, Change control registers settings.

Address: <LCBIn_base> + 458_H

Initial Value: 0000 0000_H. This register is initialized by a LCBI H/W reset and by soft reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LCBInBCY TYPE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 36.28 LCBInBCYCT register contents

Bit position	Bit name	Function
31 to 1	Reserved	When read, the value after reset is returned. When written, write the value after reset.
0	LCBIn BCYTYPE	Selects the access mode: 0: RAM 1: E-Type

36.4.3.2 LCBInBCYC - Non-TFT modes bus cycle specification control register

This register specifies the duration of bus cycle states in non-TFT operation mode. The duration is based on operation clock cycles.

The names and functions of the bits differ in RAM access mode (LCBInBCYCT.LCBInBCYTYPE = 0) and E-Type access mode (LCBInBCYCT.LCBInBCYTYPE = 1).

Access: This register can be accessed in 32-bit units.
The LCD Bus Interface must be stopped (LCBInOPMODE.LCBInOPMODE[1:0] = 00_B) before writing to this register. For details refer to Section 36.3.7, Change control registers settings.

Address: <LCBIn_base> + $45C_H$

Initial Value: FFFF $0FFF_H$. This register is initialized by a LCBI H/W reset and by soft reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCBInTIRD[7:0]								LCBInTIWR[7:0] LCBInTMED[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	LCBInTIDW[3:0] LCBInTMDW[3:0]				LCBInTIDZ[3:0] LCBInTMDZ[3:0]				LCBInTIAD[3:0] LCBInTMAD[3:0]			
R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 36.29 LCBInBCYC register contents in RAM access mode

Bit position	Bit name	Function
31 to 24	LCBIn TIRD[7:0]	Width of the read strobe TIRD = LCBInTIRD[7:0] + 1
23 to 16	LCBIn TIWR[7:0]	Width of the write strobe TIWR = LCBInTIWR[7:0] + 1
15 to 12	Reserved	When read, the value after reset is returned. When written, write the value after reset.
11 to 8	LCBIn TIDW[3:0]	Activating read/write strobe TIDW = LCBInTIDW[3:0] + 1 **Note:** not effective in fast bus cycle mode (LCBInOPMODE.LCBInOPMFAST = 1)
7 to 4	LCBIn TIDZ[3:0]	End of the bus cycle TIDZ = LCBInTIDZ[3:0] + 1
3 to 0	LCBIn TIAD[3:0]	Waiting for address TIAD = LCBInTIAD[3:0] + 1 **Note:** not effective in fast bus cycle mode (LCBInOPMODE.LCBInOPMFAST = 1)

Table 36.30 LCBInBCYC register contents in E-Type access mode (1/2)

Bit position	Bit name	Function
31 to 24	Reserved	When read, the value after reset is returned. When written, write the value after reset.
23 to 16	LCBIn TMED[7:0]	Width of the read/write strobe TMED = LCBInTMED[7:0] + 1
15 to 12	Reserved	When read, the value after reset is returned. When written, write the value after reset.
11 to 8	LCBIn TMDW[3:0]	Activating read/write direction TMDW = LCBInTMDW[3:0] + 1 **Note:** not effective in fast bus cycle mode (LCBInOPMODE.LCBInOPMFAST = 1)
7 to 4	LCBIn TMDZ[3:0]	End of the bus cycle TMDZ = LCBInTMDZ[3:0] + 1

Table 36.30 LCBInBCYC register contents in E-Type access mode (2/2)

Bit position	Bit name	Function
3 to 0	LCBIn TMAD[3:0]	Waiting for address TMAD = LCBInTMAD[3:0] + 1 **Note:** not effective in fast bus cycle mode (LCBInOPMODE.LCBInOPMFAST = 1)

36.4.4 LCD Bus Interface TFT mode control registers details

36.4.4.1 LCBInTFTPRS - TFT cycle prescaler control register

This register specifies the prescaler for the pixel clock in TFT operation mode.

The duration of most bus cycle states in TFT operation mode is based on pixel clock cycles. The exceptions are LCBInTFDCH, LCBInTFDCW and LCBInTFDCD; they are based on operation clock cycles.

Access: This register can be accessed in 32-bit units.
The LCD Bus Interface must be stopped (LCBInOPMODE.LCBInOPMODE[1:0] = 00_B) before writing to this register. For details refer to Section 36.3.7, Change control registers settings.

Address: <LCBIn_base> + 460_H

Initial Value: 0000 $FFFF_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCBInTFTPCPRS[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 36.31 LCBInTFTPRS register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, write the value after reset.
15 to 0	LCBInTFTPCPRS[15:0]	Specifies the pixel clock prescaler.

LCBInTFTPCPRS[15:0]	Pixel clock
0000_H	Setting prohibited
0001_H	Operation clock / 2
0002_H	Operation clock / 3
...	...
$FFFE_H$	Operation clock / 65535
$FFFF_H$	Operation clock / 65536

CAUTION

When the prescaler for the operation is 1 (LCBInCKSEL.LCBInCKS[2:0] = 000_B), the prescaler for the operation clock must not be set to one (LCBInTFTPRS.LCBInTFTPCPRS[15:0] = 0000_H).

36.4.4.2 LCBInTFTCYC0 - TFT cycle timing specification control register 0

This register specifies the duration of bus cycle states in TFT operation mode. The duration is based on pixel clock cycles or on operation clock cycles.

Access: This register can be accessed in 32-bit units.
The LCD Bus Interface must be stopped (LCBInOPMODE.LCBInOPMODE[1:0] = 00_B) before writing to this register. For details refer to Section 36.3.7, Change control registers settings.

Address: <LCBIn_base> + 464_H

Initial Value: $FF7F\ FFFF_H$. This register is initialized by a LCBI H/W reset and by soft reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCBInTFDCW[7:0]								0	LCBInTFDCD[6:0]						
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCBInTFDCH[7:0]								LCBInTFDHV[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 36.32 LCBInTFTCYC0 register contents

Bit position	Bit name	Function
31 to 24	LCBIn TFDCW[7:0]	Pixel clock active time 00_H: 1 operation clock cycle ... FF_H: 256 operation clock cycles
23	Reserved	When read, the value after reset is returned. When written, write the value after reset.
22 to 16	LCBIn TFDCD[6:0]	Wait state between data becoming active and pixel clock becoming active 00_H: 1 operation clock cycle ... $7F_H$: 128 operation clock cycles
15 to 8	LCBIn TFDCH[7:0]	Wait state between lines 00_H: 1 operation clock cycle ... FF_H: 256 operation clock cycles
7 to 0	LCBIn TFDHV[7:0]	Wait state between picture frames 00_H: 1 pixel clock cycle ... FF_H: 256 pixel clock cycles

CAUTION

Every pixel clock cycle comprises (pixel clock prescaler value) / (operation clock prescaler) operation clock cycles.

- **The values of LCBInTFDCW[7:0] and LCBInTFDCD[6:0] must be less than or equal to LCBInTFTPCPRS[15:0] / LCBInCKS[2:0].**

Example

- Pixel clock prescaler value = 80 (LCBInTFTPRS.LCBInTFTPCPRS[15:0] = $4F_H$)
- Operation clock prescaler value = 4 (LCBInCKSEL.LCBInCKS[2:0] = 010_B)

The values of LCBInTFDCW[7:0] and LCBInTFDCD[6:0] must be less than or equal to $20 = 14_H$. This means, the wait states must be set to 13_H or less.

36.4.4.3 LCBInTFTCYC1 - TFT cycle timing specification control register 1

This register specifies the duration of bus cycle states in TFT operation mode and the number of lines per picture frame. The duration of bus cycle states is based on the number of lines per picture frame.

Access: This register can be accessed in 32-bit units.
The LCD Bus Interface must be stopped (LCBInOPMODE.LCBInOPMODE[1:0] = 00_B) before writing to this register. For details refer to Section 36.3.7, Change control registers settings.

Address: <LCBIn_base> + 468_H

Initial Value: FFFF $FFFF_H$. This register is initialized by a LCBI H/W reset and by soft reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCBInTFNVE[5:0]						LCBInTFNVC[9:0]									
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCBInTFNVB[7:0]								LCBInTFNHVS[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 36.33 LCBInTFTCYC1 register contents

Bit position	Bit name	Function
31 to 26	LCBIn TFNVE[5:0]	Vertical front porch time*1 00_H: 1 x $\overline{\text{HSYNC}}$ pulse ... $3F_H$: 64 x $\overline{\text{HSYNC}}$ pulse
25 to 16	LCBIn TFNVC[9:0]	Number of lines per picture frame 000_H: 1 x $\overline{\text{HSYNC}}$ pulse ... $3FF_H$: 1024 x $\overline{\text{HSYNC}}$ pulse
15 to 8	LCBIn TFNVB[7:0]	Vertical back porch time*1 00_H: 1 x $\overline{\text{HSYNC}}$ pulse ... FF_H: 256 x $\overline{\text{HSYNC}}$ pulse
7 to 0	LCBIn TFNHVS[7:0]	$\overline{\text{VSYNC}}$ pulse width*1 00_H: 1 x $\overline{\text{HSYNC}}$ pulse ... FF_H: 256 x $\overline{\text{HSYNC}}$ pulse

Note 1. The front porch time, back porch time and $\overline{\text{VSYNC}}$ pulse width is determined by counting the active edges of $\overline{\text{HSYNC}}$ pulses up to the respective number given in this register.
Since the $\overline{\text{HSYNC}}$ edges are only counted, while the data enable signal DE is inactive, the real time length is also dependent on LCBInTFTCYC0.LCBInTFDHV[7:0].

36.4.4.4 LCBInTFTCYC2 - TFT cycle timing specification control register 2

This register specifies the duration of bus cycle states in TFT operation mode. The duration is based on pixel clock cycles.

Access: This register can be accessed in 32-bit units.
The LCD Bus Interface must be stopped (LCBInOPMODE.LCBInOPMODE[1:0] = 00_B) before writing to this register. For details refer to Section 36.3.7, Change control registers settings.

Address: <LCBIn_base> + $46C_H$

Initial Value: FFFF $FFFF_H$. This register is initialized by a LCBI H/W reset and by soft reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCBInTFNHE[5:0]						LCBInTFNHC[9:0]									
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCBInTFNHB[7:0]								LCBInTFNCHS[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 36.34 LCBInTFTCYC2 register contents

Bit position	Bit name	Function
31 to 26	LCBIn TFNHE[5:0]	Horizontal front porch time*1 00_H: 1 pixel clock cycle ... $3F_H$: 64 pixel clock cycles
25 to 16	LCBIn TFNHC[9:0]	Number of pixels per line 000_H: 1 pixel clock cycle ... $3FF_H$: 1024 pixel clock cycles
15 to 8	LCBIn TFNHB[7:0]	Horizontal back porch time*1 00_H: 1 pixel clock cycle ... FF_H: 256 pixel clock cycles
7 to 0	LCBIn TFNCHS[7:0]	HSYNC pulse width*1 00_H: 1 pixel clock cycle ... FF_H: 256 pixel clock cycles

Note 1. The front porch time, back porch time and $\overline{\text{HSYNC}}$ pulse width is determined by counting the active edges of pixel clock pulses up to the respective number given in this register.
Since the pixel clock edges are only counted, while the data enable signal DE is inactive, the real time length is also dependent on LCBInTFTCYC0.LCBInTFDCD[6:0].

36.4.5 LCD Bus Interface data registers details

36.4.5.1 LCBInDAT0SWR, LCBInDAT0CWR - LCD Bus Interface data write registers

These data registers contain the data that is to be written.

Writing to a data write register initiates a write access on the LCD bus. All data write registers are 32-bit registers. Depending on the access type, only 8 bits, 16 bits, or 32 bits are used.

For details, refer to Section 36.3.5, Data transfer types.

Access:
- Byte access registers LCBInDAT0SWR/LCBInDAT0CWR:
  These registers can be written in 32-bit units.
- Halfword access registers LCBInDAT0SWRH/LCBInDAT0CWRH:
  These registers can be written in 32-bit units.
- Word access registers LCBInDAT0SWRW/LCBInDAT0CWRW:
  These registers can be written in 32-bit units.

Address: Refer to Table 36.35, Data write registers overview

Initial Value: 0000 0000_H.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCBInDATSWR[31:16]/LCBInDATCWR[31:16]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCBInDATSWR[15:0]/LCBInDATCWR[15:0]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 36.35 Data write registers overview

Data write register	Used bits	Access type	Function	Address
LCBInDAT0SWR	LCBInDATSWR[7:0]	Byte	Write 8-bit display data with A0 active. If CLUT is enabled, the data is converted to RGB values.	<LCBIn_base> + 400_H
LCBInDAT0SWRH	LCBInDATSWR[15:0]	Halfword	Write 16-bit display data with A0 active. If CLUT is enabled, the data is converted to RGB values.	<LCBIn_base> + 404_H
LCBInDAT0SWRW	LCBInDATSWR[31:0]	Word	Write 32-bit display data with A0 active. If CLUT is enabled, the data is converted to RGB values.	<LCBIn_base> + 408_H
LCBInDAT0CWR	LCBInDATCWR[7:0]	Byte	Write 8-bit control data with A0 inactive.	<LCBIn_base> + $40C_H$
LCBInDAT0CWRH	LCBInDATCWR[15:0]	Halfword	Write 16-bit control data with A0 inactive.	<LCBIn_base> + 410_H
LCBInDAT0CWRW	LCBInDATCWR[31:0]	Word	Write 32-bit control data with A0 inactive.	<LCBIn_base> + 414_H

NOTE

The active and inactive level of A0 is determined by LCBInOUTLEV.LCBInTLA0.

Depending on the operation mode, the LCD bus width and the access type (byte, halfword, or word access) certain data write register accesses are forbidden:

Table 36.36 Allowed (OK) and forbidden (X) access to data write registers

LCD Bus Interface operation mode	LCD bus width	LCBInDAT0SWR	LCBInDAT0SWRH	LCBInDAT0SWRW	LCBInDAT0CWR	LCBInDAT0CWRH	LCBInDAT0CWRW
Non-TFT (without CLUT usage)	8	OK	OK	OK	OK	OK	OK
	9	X	OK	OK	X	OK	OK
	16	X	OK	OK	X	OK	OK
	18	X	X	OK	X	X	OK
Non-TFT (with CLUT usage)	9 or 18	OK	OK	OK	OK	OK	OK
TFT (without CLUT usage)	8	OK	OK	OK	OK	OK	OK
	9	X	OK	OK	X	OK	OK
	16	X	OK	OK	X	OK	OK
	18	X	X	OK	X	X	OK
TFT (with CLUT usage)	9 or 18	OK	OK	OK	OK	OK	OK

36.4.5.2 LCBInDAT0SRD, LCBInDAT0CRD - LCD Bus Interface data read registers

In non-TFT operation mode, a data read register contains the data that is to be read.

When read data is available from the LCD bus, this data is accessed by reading a data read register. All data read registers are 32-bit registers. Depending on the access type, only 8 bits, 16 bits, or 32 bits are used for data.

For details, refer to Section 36.3.5, Data transfer types.

Access:
- Byte access registers LCBInDAT0SRD/LCBInDAT0CRD: These registers can be read in 32-bit units.
- Halfword access registers LCBInDAT0SRDH/LCBInDAT0CRDH: These registers can be read in 32-bit units.
- Word access registers LCBInDAT0SRDW/LCBInDAT0CRDW: These registers can be read in 32-bit units.

Address: Refer to Table 36.37, Data read registers overview

Initial Value: 0000 0000_H.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LCBInDATSRD[31:16]/LCBInDATCRD[31:16]/LCBInDATRD[31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCBInDATSRD[15:0]/LCBInDATCRD[15:0]/LCBInDATRD[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 36.37 Data read registers overview

Data read register	Used bits	Access type	Function	Address
LCBInDAT0SRD	LCBInDATSRD[7:0]	Byte	Read 8-bit display data with A0 active.*1	<LCBIn_base> + 418_H
LCBInDAT0SRDH	LCBInDATSRD[15:0]	Halfword	Read 16-bit display data with A0 active.*1	<LCBIn_base> + $41C_H$
LCBInDAT0SRDW	LCBInDATSRD[31:0]	Word	Read 32-bit display data with A0 active.*1	<LCBIn_base> + 420_H
LCBInDAT0CRD	LCBInDATCRD[7:0]	Byte	Read 8-bit control data with A0 inactive.*1	<LCBIn_base> + 424_H
LCBInDAT0CRDH	LCBInDATCRD[15:0]	Halfword	Read 16-bit control data with A0 inactive.*1	<LCBIn_base> + 428_H
LCBInDAT0CRDW	LCBInDATCRD[31:0]	Word	Read 32-bit control data with A0 inactive.*1	<LCBIn_base> + $42C_H$
LCBInDATRDW	LCBInDATRD[31:0]	Word	Read LCD data (32-bit access only).*2	<LCBIn_base> + 430_H

Note 1. If the LCD Bus Interface has been stopped due to a debug break (with LCBInEMU.LCBInSVSDIS = 0), it is not allowed to read these registers.
The LCBInDATRDW register can be used instead to obtain the last read data.

Note 2. When reading register LCBInDATRDW, no read access is triggered on the LCD bus but previously fetched data is reread. Refer to Section 36.3.1, Non-TFT operation mode operation mode for details.

NOTE

The active and inactive level of A0 is determined by LCBInOUTLEV.LCBInTLA0.

Depending on the operation mode, the LCD bus width and the access type (byte, halfword, or word access) certain data read register accesses are forbidden:

Table 36.38 Allowed (OK) and forbidden (X) access to data read registers

LCD Bus Interface operation mode	LCD bus width	LCBInDAT0SRD	LCBInDAT0SRDH	LCBInDAT0SRDW	LCBInDAT0CRD	LCBInDAT0CRDH	LCBInDAT0CRDW	LCBInDATRDW
Non-TFT*1	8	OK	OK	OK	OK	OK	OK	OK
	9	X	OK	OK	X	OK	OK	X
	16	X	OK	OK	X	OK	OK	X
	18	X	X	OK	X	X	OK	X
TFT		X						

Note 1. Reading of data registers in non-TFT operation mode *with* CLUT usage results in reading of data registers in non-TFT operation mode *without* CLUT usage.

36.4.6 CLUT Data RAM registers

36.4.6.1 LCBInCLUTDmW - CLUT data RAM registers (m = 0 to 255)

These registers are used to read and write the data of the color lookup table CLUT. Each LCBInCLUTDmW register is used for the definition of 18-bit color values, for example RGB values.

For details refer to Section 36.3.3, Color usage.

Access: This register can be accessed in 32-bit units.
It can only be read/written in CLUT definition mode (LCBInOPMODE.LCBInOPMODE[1:0] = 11_B).

Address: <LCBIn_base> + m x 4_H (m = 0 to 255)

Initial Value: This register is not initialized.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	LCBIn CD[17:16]	
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCBInCD[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 36.39 LCBInCLUTDmW register contents

Bit position	Bit name	Function
31 to 18	Reserved	When read, the value after reset is returned. When written, write the value after reset.
17 to 0	LCBIn CD[17:0]	CLUT data (18-bit color values)

Section 37 Video and Graphics Functions

37.1 Architecture of the Graphics Subsystem

The Graphics Subsystems comprise following modules:

- Video channels with the Video Input and Output interfaces, the Image Synthesizers and the Video Output Warping Engine.
- Sprite Engine with RLE Units for de-compressing RLE encoded colour data and Sprite Units for sprite layers.

The Graphics Subsystem is connected to the XC0 cross-connect for transfer of video and graphics data to and from the various memories.

The Video Input Interfaces and the Distortion Correction with the Video Output Warping Engine have directly access to the XC0 cross-connect - and thus to the memories.

All memory read accesses of the Image Synthesizers are routed via the XC2 cross-connect to the Sprite Engine, which acquires the colour data from the memories via the cross-connect XC0.

Likewise the other cross-connect systems XC0 and XC1 the XC2 is also a multilayer transaction based bus system.
Refer to Section 14, Bus Architecture for details about the bus architecture.

Refer to the following sections for detailed descriptions about the various functional components:

- Video Input Interface: **Input Controller** in Section 38, Video Data Controller E (VDCE)
- Scaler: **Scaler** in Section 38, Video Data Controller E (VDCE)
- Format Converter: **Image Quality Improver** in Section 38, Video Data Controller E (VDCE)
- Image Synthesizer: **Image Synthesizer** in Section 38, Video Data Controller E (VDCE)
- Distortion Correction: **Output Image Generator** in Section 38, Video Data Controller E (VDCE) and Section 39, Video Output Warping Engine (VOWE)
- Video Output Interface: **Output Controller** in Section 38, Video Data Controller E (VDCE)
- General control functions: **System Controller** in Section 38, Video Data Controller E (VDCE)

CAUTION

Correct operation of the video and graphics functions is not guaranteed if CPU runs operated by the EMCLK.

However, the registers of the video and graphics modules can be accessed.

37.2 Graphics Subsystem Block Diagrams

The Graphics Subsystems of the various devices are different and shown in the following sections.

NOTES

1. The XC0 cross-connect in the following diagrams are not part of the Graphics Subsystem, but are included better understanding of the entire colour data transfers.
2. The arrows in the following diagrams indicate the direction of the colour data flow. For information about the master and slave ports at the cross-connects refer to Section 14, Bus Architecture.

37.2.1 D1L2(H) Graphics Subsystem

The D1L2(H) device features one video output channel and supports four graphic layers.

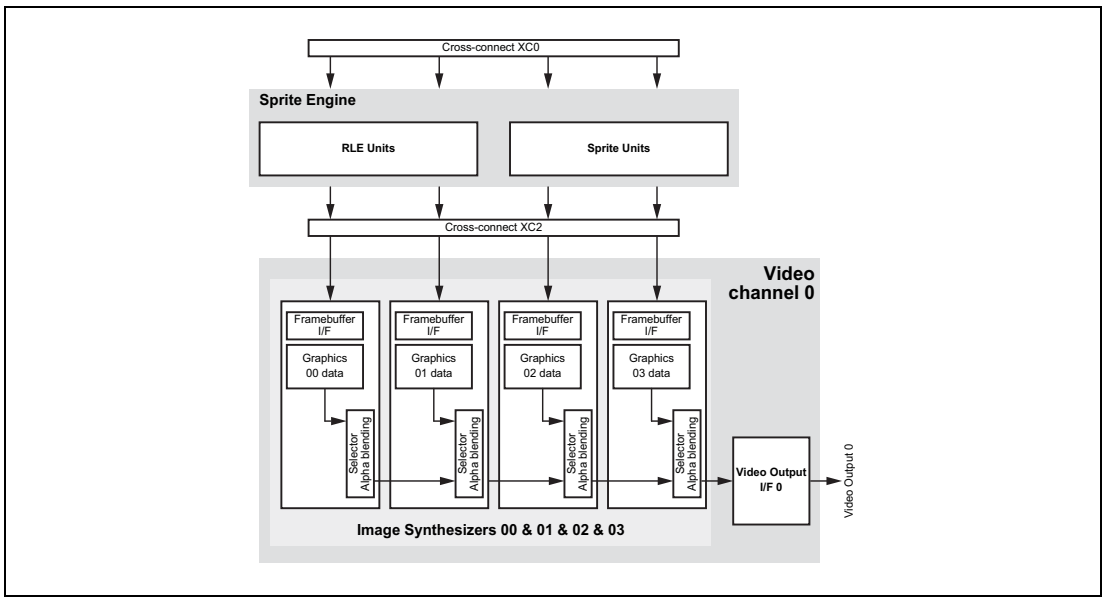


Figure 37.1 D1L2(H) Graphics Subsystem

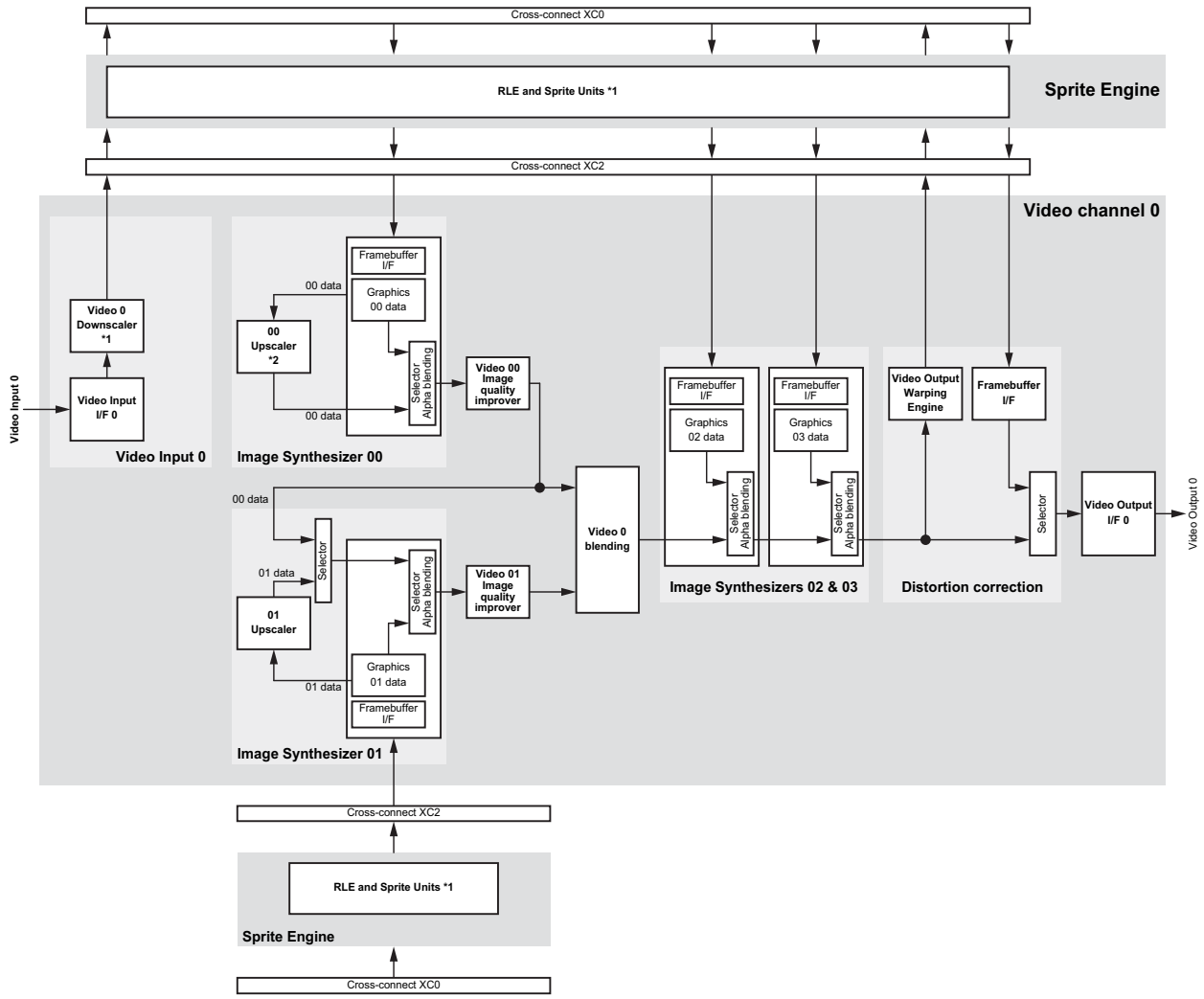
37.2.2 D1M1(H), D1M1-V2, D1M2(H) and D1M1A Graphics Subsystem

The D1M1(H), D1M1-V2devices features one video input and output channel, while the D1M2(H) device provides two video input and two video output channels.

Following layer combinations are possible:

- D1M1(H), D1M1-V2:
  - 4 graphics layers
  - 1 video layer + 3 graphics layers
- D1M2(H) for each of channel 0 and channel 1:
  - 4 graphics layers
  - 1 video layer + 3 graphics layers
  - 2 video layers + 2 graphics layers
- D1M1A:
  - 4 graphics layers (channel 0 and channel 1)
  - 1 video layer + 3 graphics layers (only channel 0)

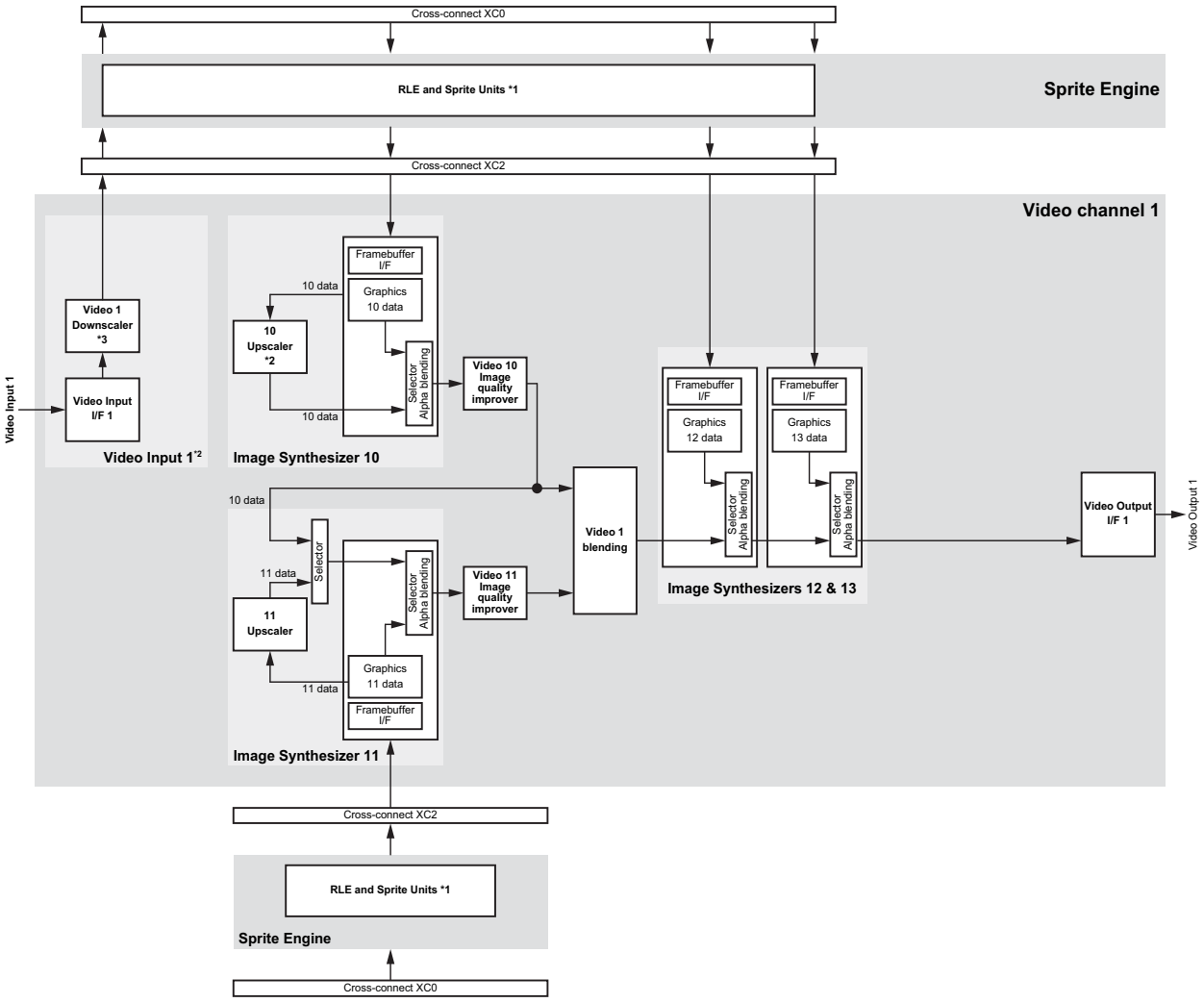
The Video Output channel 0 of the D1M1(H), D1M1-V2, D1M2(H) and D1M1A devices have a Video Output Warping Engine for predistortion of video images.



Note 1. – D1M1(H), D1M2(H): RLE unit can be used for background layer.
– D1M1-V2, D1M1A: RLE units can be used for all layers.

Note 2. The Video 0 Downscaler and 00 Upscaler can not be used concurrently. Either the Downscaler or the Upscaler can be used at the same time.

Figure 37.2 D1M1(H), D1M1-V2, D1M2(H) and D1M1A Graphics Subsystem video channel 0 data flow



Note 1. – D1M2(H): RLE unit can be used for background layer.
– D1M1A: RLE units can be used for all layers.

Note 2. The Video 1 Downscaler and 10 Upscaler can not be used concurrently. Either the Downscaler or the Upscaler can be used at the same time.

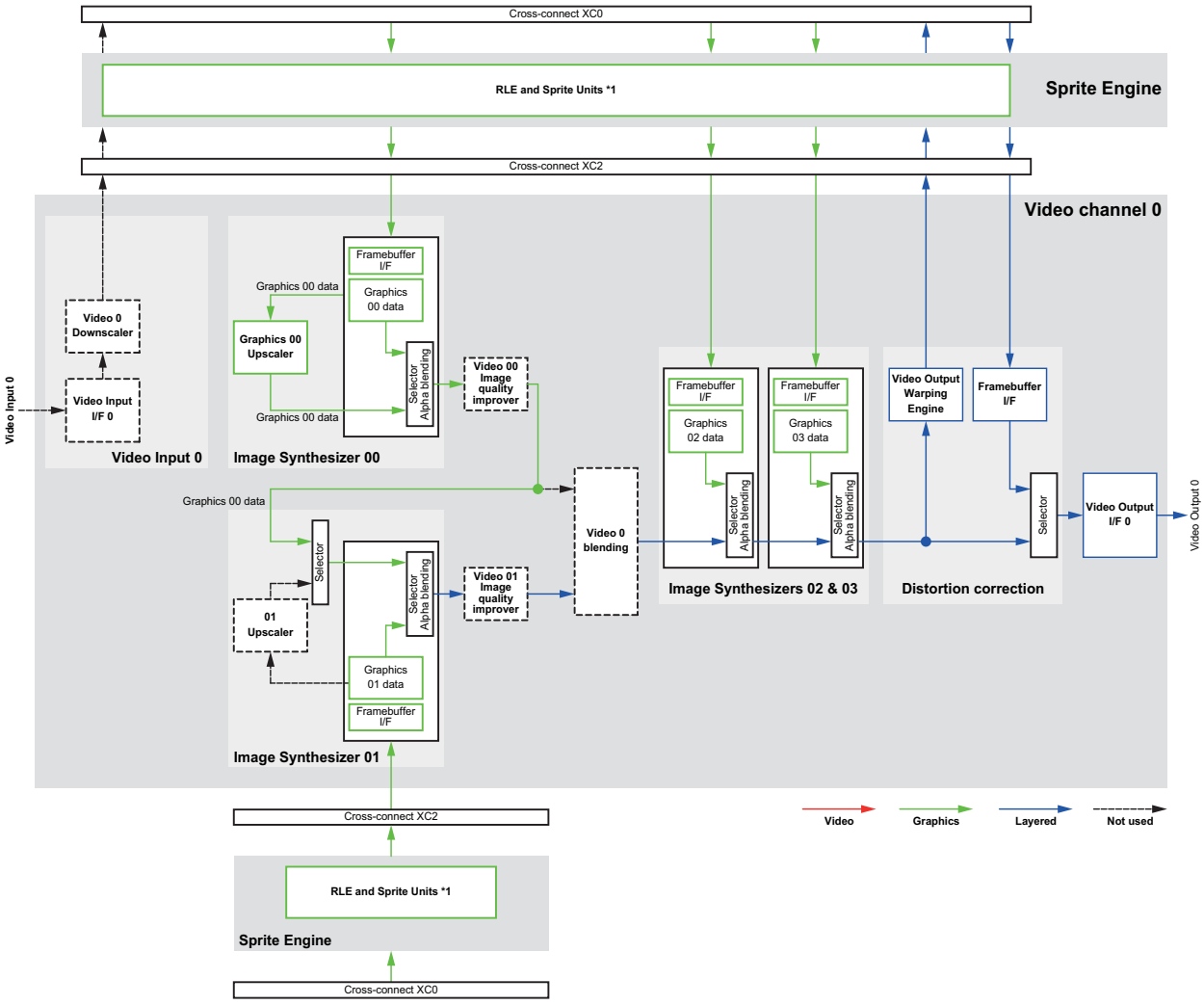
Note 3. The Video Input I/F 1 is not available for D1M1A devices.

Figure 37.3 D1M2(H) and D1M1A Graphics Subsystem video channel 1 data flow

37.3 Video input and output data flow

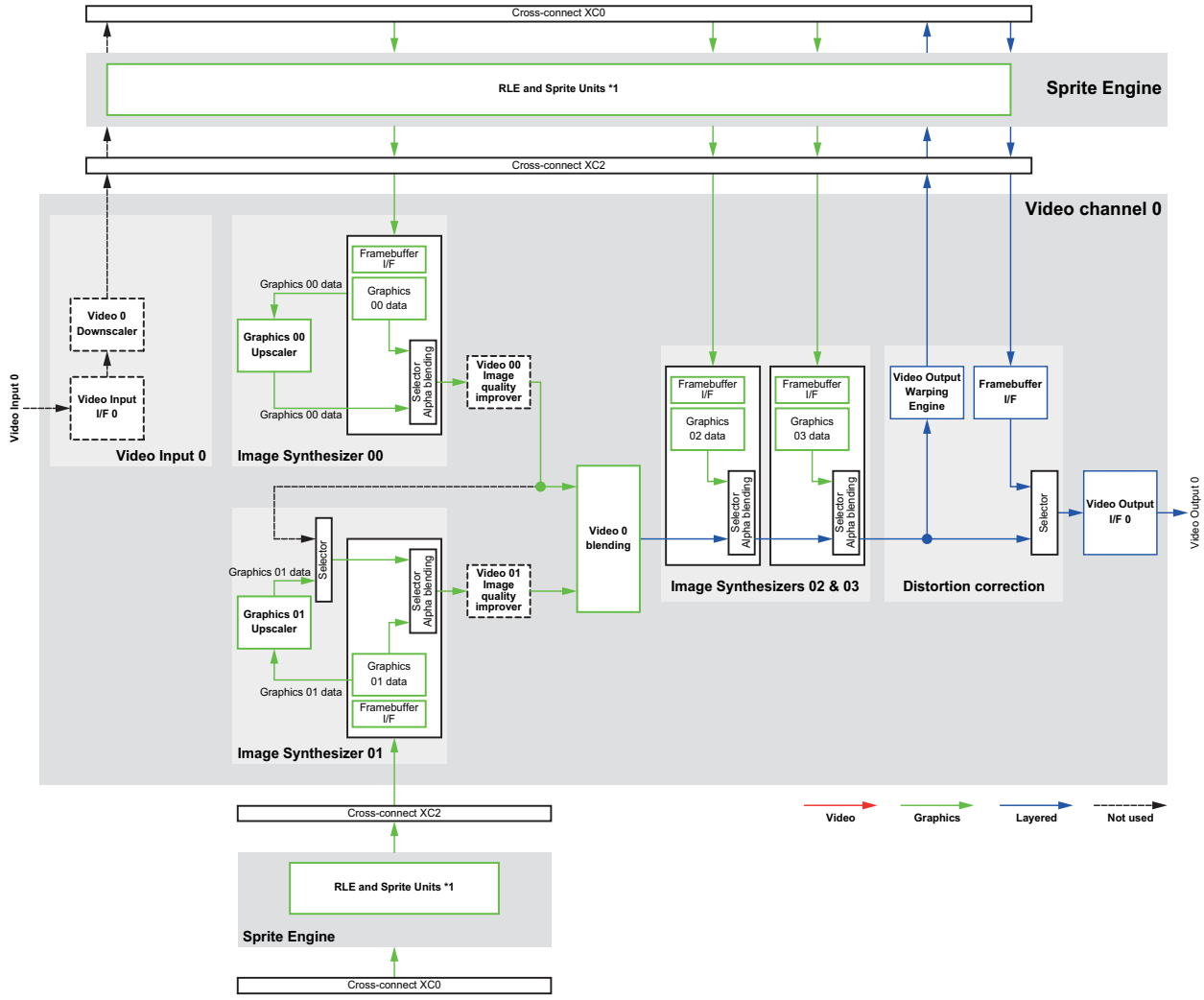
37.3.1 No video layers

If no video layers are used, each of the four Image Synthesizers can generate one graphic layer. The first Image Synthesizer (00 for channel 0, 10 for channel 1 of D1M2(H)) can utilize the upscaler for enlarging the graphics data, which constitutes the lowest layer 0, i.e. the background layer.



Note 1. – D1M1(H), D1M2(H): RLE unit can be used for background layer.
– D1M1-V2, D1M1A: RLE units can be used for all layers.

Figure 37.4 D1M1(H), D1M1-V2, D1M2(H) and D1M1A video channel 0 data flow: four graphics layers - configuration 1

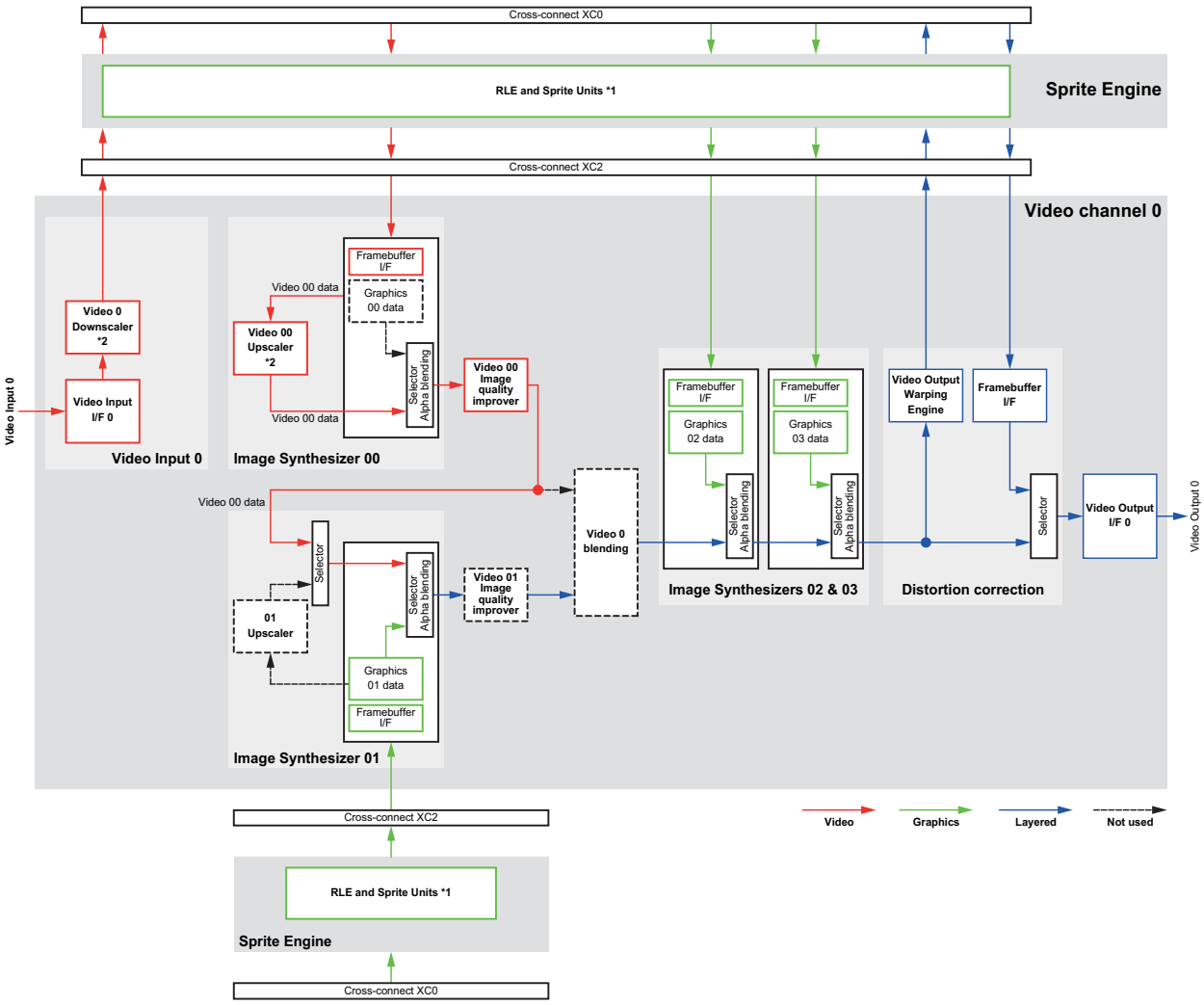


Note 1. – D1M1(H), D1M2(H): RLE unit can be used for background layer.
– D1M1-V2, D1M1A: RLE units can be used for all layers.

Figure 37.5 D1M1(H), D1M1-V2, D1M2(H) and D1M1A video channel 0 data flow: four graphics layers - configuration 2

37.3.2 One video layer

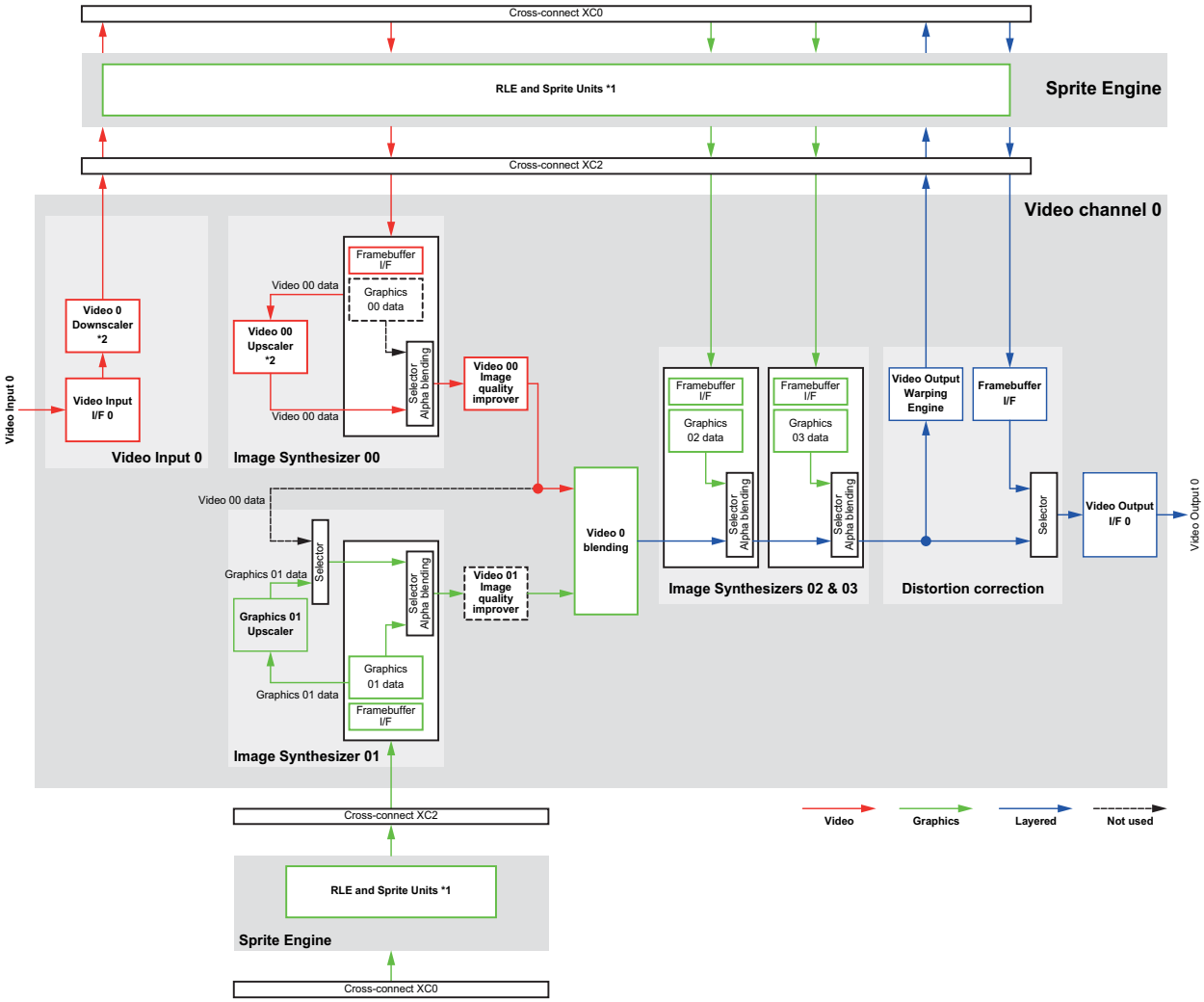
If one video layer is used, the first Image Synthesizer (00 for channel 0, 10 for channel 1 of D1M2(H) and D1M1A) can utilize the upscaler for enlarging the video data, which constitutes the lowest layer 0, i.e. the background layer.
All other Image Synthesizers can add up to three graphics layers.



Note 1. – D1M1(H), D1M2(H): RLE unit can be used for background layer.
– D1M1-V2, D1M1A: RLE units can be used for all layers.
However since captured video data from the Video Input interface is to be used on the background layer in this example, the RLE engines in the RLE units are bypassed anyway.

Note 2. The Video 0 Downscaler and Video 00 Upscaler can not be used concurrently. Either the Downscaler or the Upscaler can be used at the same time.

Figure 37.6 D1M1(H), D1M1-V2, D1M2(H) and D1M1A video channel 0 data flow: one video + up to three graphics layers - configuration 1



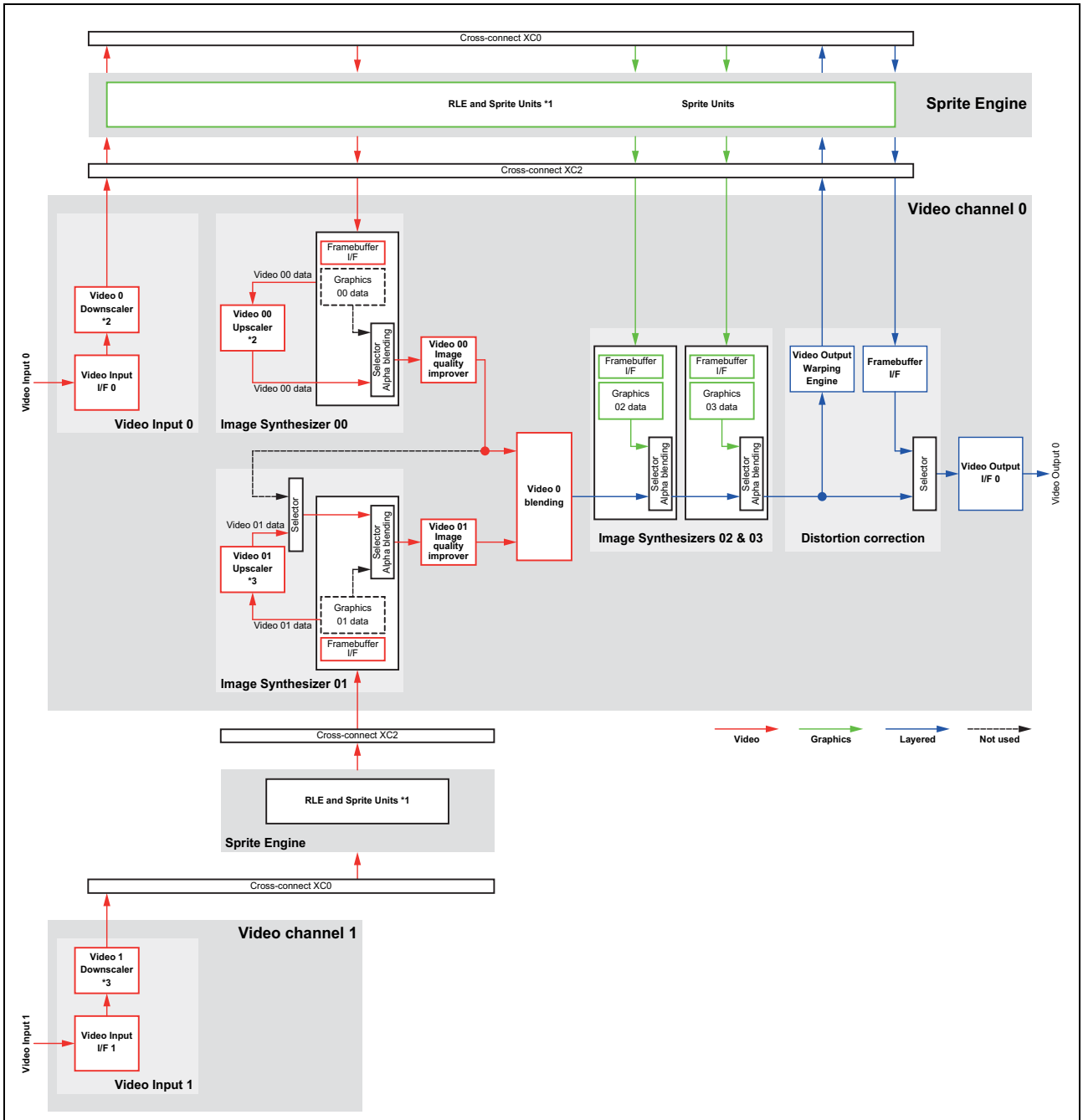
Note 1. – D1M1(H), D1M2(H): RLE unit can be used for background layer.
– D1M1-V2, D1M1A: RLE units can be used for all layers.
However since captured video data from the Video Input interface is to be used on the background layer in this example, the RLE engines in the RLE units are bypassed anyway.

Note 1. The Video 0 Downscaler and Video 00 Upscaler can not be used concurrently. Either the Downscaler or the Upscaler can be used at the same time.

Figure 37.7 D1M1(H), D1M1-V2, D1M2(H) and D1M1A video channel 0 data flow: one video + up to three graphics layers - configuration 2

37.3.3 Two video layers with D1M2H

If the captured data of both D1M2H Video Inputs are used, the Video Input data are scaled up by Image Synthesizer 00 and 01 (10 and 11 for channel 1) and combined by a separate video data blending unit. Blending is performed with a definable constant α value.
The remaining Image Synthesizers can add up to two graphics layers.



Note 1. RLE unit can be used for background layers.
However since captured video data from the Video Input interfaces are to be used on the background layer in this example, the RLE engines in the RLE units are bypassed anyway.

Note 2. The Video 0 Downscaler and Video 00 Upscaler can not be used concurrently. Either the Downscaler or the Upscaler can be used at the same time.

Note 3. The Video 1 Downscaler and Video 01 Upscaler can not be used concurrently. Either the Downscaler or the Upscaler can be used at the same time.

Figure 37.8 D1M2H video channel 0 data flow: two video + up to two graphics layers

37.4 D1M1(H), D1M1-V2, D1M2(H) and D1M1A video input/output synchronization

The D1M1(H), D1M1-V2, D1M2(H) and D1M1A devices provide basically two modes for selecting screen synchronization signals:

- Video output - video input synchronized mode
  In this mode the Video Input and Output is operating frame-synchronous, i.e. the external Video Input vertical synchronization signal is used throughout the entire input - output path.
- Separate mode
  In this mode a separate vertical synchronization signal is generated for the output path, independently from the Video Input.

In either case the horizontal synchronization signals are generated independently from the Video Input. The VSYNC signals are checked and erroneous VSYNC pulses are suppressed and missing VSYNC pulses are substituted in order to ensure stable continuous frame synchronization.

Interrupts

Three interrupts are generated related to the synchronization:

- Video input VSYNC interrupt, marks occurrence of the external VSYNC signal
- Video output VSYNC interrupt, marks occurrence of the VSYNC signals, used in the video output path
- VSYNC substitution interrupts, marks occurrence of a VSYNC substitution

NOTE

For more details refer to “Synchronization Control” in the “Scaler” section of “Video Data Controller (VDCE)”.

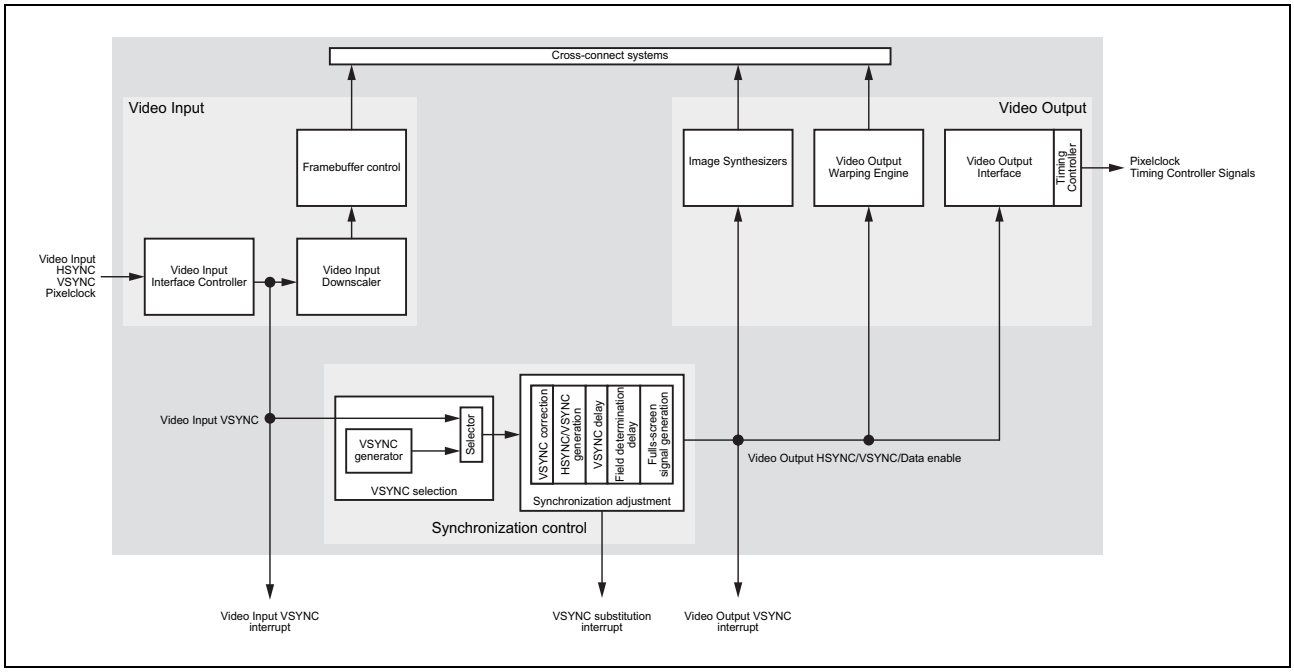


Figure 37.9 D1M1(H), D1M1-V2, D1M2(H) and D1M1A video input/output synchronization

37.5 Video channels clock generation

The clocks for the video channels can be derived from the Spread-Spectrum clock generator PLL0 or from a non-jittered clock of another PLL circuit, in case the connected display does not allow a jittered pixel clock.

The maximum video output pixel clock frequencies C_ISO_VDCEnCLK are:

- D1L2(H): 10 MHz
- D1M1(H), D1M1-V2: 30 MHz
- D1M2(H), D1M1A: 48 MHz

37.5.1 D1L2(H) pixel clock generator

The pixel clock C_ISO_VDCE0CLK of the video output interface is generated by a divider 2..255 from one of the clocks from the clock generators (PLLs).

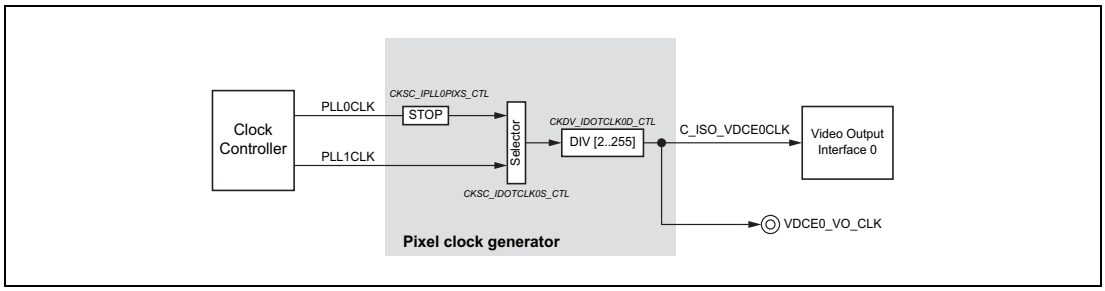


Figure 37.10 D1L2(H) video output pixel clock generation

37.5.2 D1M1(H)pixel clock generator

The pixel clock DOTCLK is generated by a divider 2..255 from one of the clocks from the clock generators (PLLs).

Several options are provided for C_ISO_VI0PIXCLK and C_ISO_VDCE0CLK, depending on the application scenario.

Video input interface used

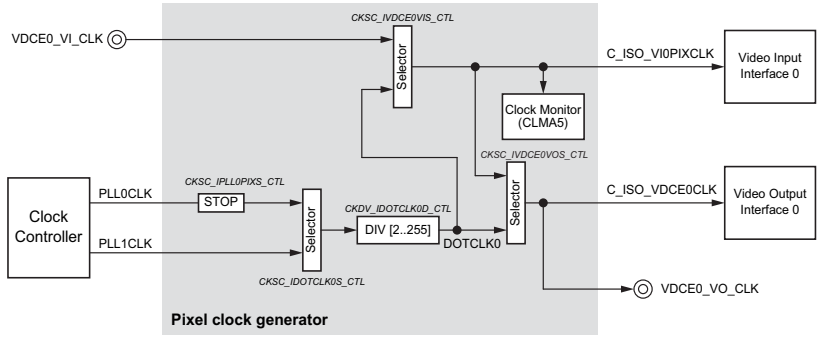
I.e. external video input clock VDCE0_VI_CLK is available.

- C_ISO_VI0PIXCLK = VDCE0_VI_CLK
- C_ISO_VDCE0CLK =
  - DOTCLK: in case video input and output operating with different pixel clocks
  - C_ISO_VI0PIXCLK: in case video input and output operating with same pixel clock

Video input interface not used

I.e. external video input clock VDCE0_VI_CLK is not available.

- C_ISO_VI0PIXCLK = DOTCLK
- C_ISO_VDCE0CLK = DOTCLK



Note 1. The name of the clock control and selection registers are shown in *italics*.

Figure 37.11 D1M1(H) video output pixel clock generation

37.5.3 D1M2(H) video channels clock generator

Video output pixel clock

To generate a video output pixel clock with sufficient granularity the D1M2(H) devices feature an additional dedicated PLL2 circuit, which can generate a jittered (via PLL0) or non-jittered (directly via MainOsc) pixel clock.
However the two Video Output Interfaces share the dedicated PLL2 for pixel clock generation.

Generally the pixel clock C_ISO_VDCEnCLK of the video output interface n is generated by a divider [2..255] from one of the clocks from the clock generators (PLLs).

Additionally each video channel can also select the external video input clock as output pixel clock.

The video output pixel clock generation differs in case a LVTTL/RGB output interface or a RSDS output interface is used.

For operating the RSDS output interface of the D1M2(H) devices a clock with 4 x pixel clock frequency is required by the RSDS logic. The pixel clock of the Video Output Interfaces needs to select a DIV4 of this clock as pixel clock.

If only a LVTTL output interface is used the pixel clock can be derived directly from the DIB [2:255] divider output.

Video input pixel clock

In normal use case the video input pixel clock is taken from the corresponding external signal.

In case of MIPI interface usage the video input clock in generated from the MIPI controller clock (by dividing by 3, 6, 12).

1. MIPI interface used (VDCE0)

   I.e.

   – C_ISO_VI0PIXCLK = C_ISO_MIPIPIXCLK

2. Video input interface used (VDCE0)

   I.e. external video input clock VDCE0_VI_CLK is available:

– C_ISO_VI0PIXCLK = VDCE0_VI_CLK

3. Video input interface not used (VDCE0)

I.e. external video input clock VDCE0_VI_CLK is not available:

– C_ISO_VI0PIXCLK = C_ISO_MIPIPIXCLK

4. Video input interface used (VDCE1)

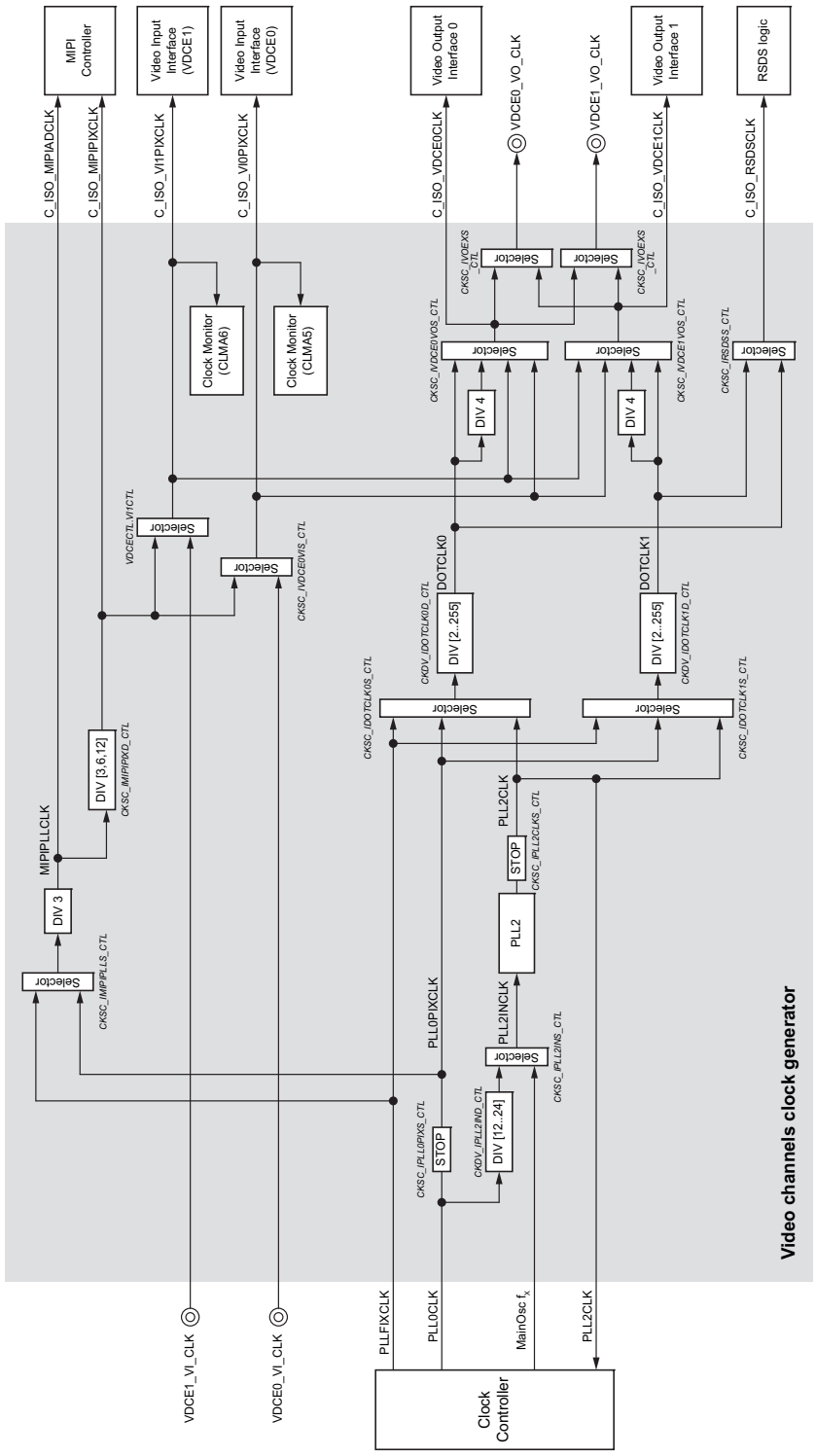
I.e. external video input clock VDCE1_VI_CLK is available:

– C_ISO_VI1PIXCLK = VDCE1_VI_CLK

5. Video input interface not used (VDCE1)

I.e. external video input clock VDCE1_VI_CLK is not available:

– C_ISO_VI1PIXCLK = C_ISO_MIPIPIXCLK



Note 1. The name of the clock control and selection registers are shown in *italics*.

Figure 37.12 D1M2(H) video channels clock generator

37.5.4 D1M1A video channels clock generator

The pixel clock DOTCLKn (n = 0, 1) is generated by a divider [1..255] from one of the clocks from the clock generators (PLLs).

In addition, D1M1A has options for VODDR, Serial RGB and Open LDI output. The video clocks have to be set by following options.

Video input interface used

I.e. external video input clock VDCE0_VI_CLK is available:

- C_ISO_VI0PIXCLK = VDCE0_VI_CLK

Video input interface not used

I.e. external video input clock VDCE0_VI_CLK is not available:

- C_ISO_VI0PIXCLK = DOTCLK

Video output interface clocks selections

The video output clock must be properly selected for the respective output mode.

Regarding the video ouput mode selection, refer to Section 37.9.3.2, D1M1A Video output selection.

(1) Video Output Interface n (VDCEn, n = 0, 1) input clocks selections

Table 37.1 Video Output Interface n input clocks selections

Output mode	C_ISO_VDCEnCLK	C_ISO_VDCEnCLK_OUT
Parallel RGB	CKSC_IVDCEnVOS_CTL*1	
Serial RGB*2	CKSC_IVDCE1VOS_CTL /[3 or 4]*1	CKSC_IVDCE1VOS_CTL*1
VODDR	VOn_PCLK	
Open LDI	DOTCLKn /7	

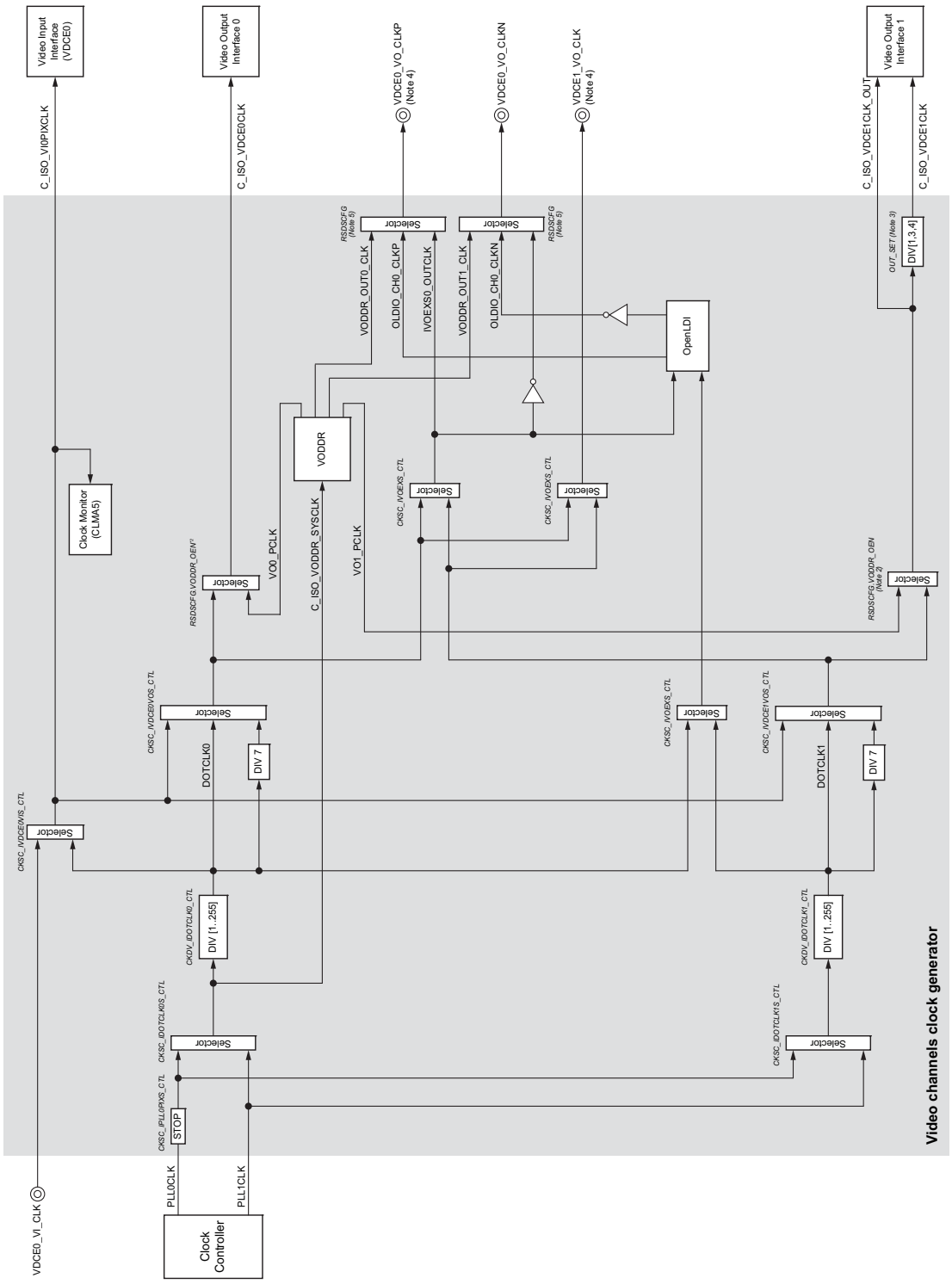
Note 1. The name of a selector register denotes the selected clock.
Note 2. Serial RGB is only available from VDCE1.

(2) External video output clocks selections

Table 37.2 External video output clocks selections

Output mode	VDCE0_VO_CLKP	VDCE1_VO_CLK
Parallel RGB	CKSC_IVDCE0VOS_CTL, (CKSC_IVDCE1VOS_CTL*2)*1	
Serial RGB	CKSC_IVDCE0VOS_CTL, (CKSC_IVDCE1VOS_CTL*2)*1	
VODDR	VODDR_OUT0_CLK	VODDR_OUT1_CLK
Open LDI	OpenLDIO_OUTCLK	–

Note 1. The name of a selector register denotes the selected clock.
Note 2. In case of CKSC_IVDCE0VOS_CTL = 1.



Note 1. The name of the clock control and selection registers are shown in *italics*.

Note 2. Refer to Section 37.10.1.2, RSDSCFG — RSDS and other video output control register.

Note 3. This selector is controlled by OUT_SET.OUT_FRQ_SEL bits of the respective Video Data Controller (VDCE). For details, refer to the OUT_SET register description in Section 38, Video Data Controller E (VDCE).

Note 4. Refer to Table 37.2, External video output clocks selections.

Note 5. Refer to Section 37.9.3, D1M1A Video output configuration.

Figure 37.13 D1M1A video channels clock generator

37.5.5 D1M1-V2 video channels clock generator

The pixel clock DOTCLK0 is generated by a divider [2..255] from one of the clocks from the clock generators (PLLs).

Several options are provided for C_ISO_VI0PIXCLK and C_ISO_VDCE0CLK, depending on the application scenario.

In addition, D1M1-V2 has options for Serial RGB output. The video clocks have to be set by following options.

Video input interface used

I.e. external video input clock VDCE0_VI_CLK is available:

- C_ISO_VI0PIXCLK = VDCE0_VI_CLK
- C_ISO_VDCE0CLK_OUT =
  - DOTCLK0: in case video input and output operating with different pixel clocks
  - C_ISO_VI0PIXCLK: in case video input and output operating with same pixel clock

Video input interface not used

I.e. external video input clock VDCE0_VI_CLK is not available:

- C_ISO_VI0PIXCLK = DOTCLK0
- C_ISO_VDCE0CLK_OUT = DOTCLK0

Video output interface clocks selections

The video output clock must be properly selected for the respective output mode.

Regarding the video output mode selection, refer to Section 37.9.4.1, D1M1-V2 video output mode selection.

(1) Video Output Interface 0 (VDCE0) input clock selections

Table 37.3 Video Output Interface n input clock selections

Output mode	C_ISO_VDCE0CLK	C_ISO_VDCE0CLK_OUT
Parallel RGB	CKSC_IVDCE0VOS_CTL*1	
Serial RGB	CKSC_IVDCE0VOS_CTL /[3 or 4]*1	CKSC_IVDCE0VOS_CTL*1

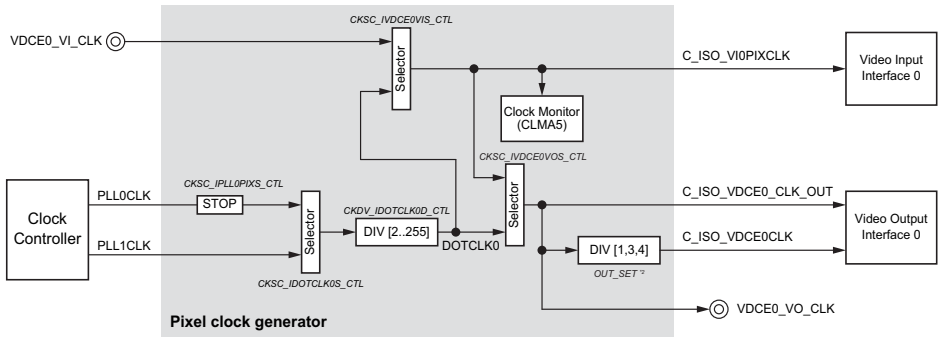
Note 1. The name of a selector register denotes the selected clock.

(2) External video output clock selections

Table 37.4 External video output clock selections

Output mode	VDCE0_VO_CLKP	VDCE0_VO_CLKN
Parallel RGB	CKSC_IVDCE0VOS_CTL*1	
Serial RGB	CKSC_IVDCE0VOS_CTL*1	

Note 1. The name of a selector register denotes the selected clock.



Note 1. The name of the clock control and selection registers are shown in *italics*.

Note 2. This selector is controlled by OUT_SET.OUT_FRQ_SEL bits of the respective Video Data Controller (VDCE).
For details, refer to the OUT_SET register description in Section 38, Video Data Controller E (VDCE).

Figure 37.14 D1M1-V2 video output pixel clock generation

37.5.6 D1M2(H) clock selections for different layer configurations

Depending on the types of layers, the data flow configuration and the usage of scalers certain settings need to be applied in order to supply all engaged modules with appropriate clocks.

The Section 37.3, Video input and output data flow shows diagrams for the various data flow configurations, which are referred to in the following table.

Table 37.5 Clock selections for different layer configurations

Layers		Clock supply for used scalers			
Types	**Configuration*4**	**Video input 0 downscaler**	**Image synthesizer 00 upscaler**	**Video input 1 downscaler**	**Image synthesizer 01 upscaler**
4 graphics	1	–	C_ISO_MIPIPIXCLK	–	–
	2				C_ISO_MIPIPIXCLK*2
1 video input + 3 graphics	1	C_ISO_MIPIPIXCLK or VDCE0_VI_CLK*1*3		–	–
	2				C_ISO_MIPIPIXCLK*2
2 video inputs + 2 graphics	–	C_ISO_MIPIPIXCLK or VDCE0_VI_CLK*1*3		VDCE1_VI_CLK*2*3	

Note 1. Selection by
CKSC_IVDCE0_VIS_CTL = 01_B: port VDCE0_VI_CLK
CKSC_IVDCE0_VIS_CTL = 10_B: C_ISO_MIPIPIXCLK

Note 2. Selection by
VDCECTL.VI1CTL = 0: port VDCE1_VI_CLK
VDCECTL.VI1CTL = 1: C_ISO_MIPIPIXCLK
The VDCECTL register is described in Section 37.10.1.1, VDCECTL — VDCE control register.

Note 3. The video input 0 or 1 downscaler and the image synthesizer 00 or 01 upscaler can not be used concurrently. Either the Downscaler or the Upscaler can be used at the same time.

Note 4. Refer to Section 37.3, Video input and output data flow for diagrams of the different configurations.

37.5.7 Video channels clock generators registers

The video channels clock generators are controlled and operated by the following registers:

Table 37.6 List of video channels clock generators registers

Register name	Shortcut	Address
PLL0PIXCLK clock control register	CKSC_IPLL0PIXS_CTL	FFF8 5640$_H$
PLL0PIXCLK clock active register	CKSC_IPLL0PIXS_ACT	FFF8 5648$_H$
PLL2INCLK clock divider register	CKDV_IPLL2IND_CTL	FFF8 5040$_H$
PLL2INCLK clock divider status register	CKDV_IPLL2IND_STAT	FFF8 5044$_H$
PLL2INCLK source clock selection register	CKSC_IPLL2INS_CTL	FFF8 5080$_H$
PLL2INCLK source clock active register	CKSC_IPLL2INS_ACT	FFF8 5088$_H$
PLL2CLK clock control register	CKSC_IPLL2CLKS_CTL	FFF8 5100$_H$
PLL2CLK clock active register	CKSC_IPLL2CLKS_ACT	FFF8 5108$_H$
DOTCLK0 source clock selection register	CKSC_IDOTCLK0S_CTL	FFF8 5940$_H$
DOTCLK0 source clock active register	CKSC_IDOTCLK0S_ACT	FFF8 5948$_H$
DOTCLK1 source clock selection register	CKSC_IDOTCLK1S_CTL	FFF8 5980$_H$
DOTCLK1 source clock active register	CKSC_IDOTCLK1S_ACT	FFF8 5988$_H$
DOTCLK0 clock divider register	CKDV_IDOTCLK0D_CTL	FFF8 59C0$_H$
DOTCLK0 clock divider status register	CKDV_IDOTCLK0D_STAT	FFF8 59C4$_H$
DOTCLK1 clock divider register	CKDV_IDOTCLK1D_CTL	FFF8 5A00$_H$
DOTCLK1 clock divider status register	CKDV_IDOTCLK1D_STAT	FFF8 5A04$_H$
VDCE0CLK source clock selection register	CKSC_IVDCE0VOS_CTL	FFF8 5880$_H$
VDCE0CLK source clock active register	CKSC_IVDCE0VOS_ACT	FFF8 5888$_H$
VDCE1CLK source clock selection register	CKSC_IVDCE1VOS_CTL	FFF8 58C0$_H$
VDCE1CLK source clock active register	CKSC_IVDCE1VOS_ACT	FFF8 58C8$_H$
Video output pixel clocks exchange register	CKSC_IVOEXS_CTL	FFF8 5900$_H$
Video output pixel clocks exchange active register	CKSC_IVOEXS_ACT	FFF8 5908$_H$
C_ISO_RSDSCLK source clock selection register	CKSC_IRSDSS_CTL	FFF8 5A40$_H$
C_ISO_RSDSCLK source clock active register	CKSC_IRSDSS_ACT	FFF8 5A48$_H$
C_ISO_VI0PIXCLK source clock selection register	CKSC_IVDCE0VIS_CTL	FFF8 5840$_H$
C_ISO_VI0PIXCLK source clock active register	CKSC_IVDCE0VIS_ACT	FFF8 5848$_H$
MIPIPLLCLK source clock selection register	CKSC_IMIPIPLLS_CTL	FFF8 5680$_H$
MIPIPLLCLK source clock active register	CKSC_IMIPIPLLS_ACT	FFF8 5688$_H$
C_ISO_MIPIPIXCLK clock divider register	CKSC_IMIPIPIXD_CTL	FFF8 56C0$_H$
C_ISO_MIPIPIXCLK clock divider active register	CKSC_IMIPIPIXD_ACT	FFF8 56C8$_H$

NOTE

The function of the PLL2 and its control registers are described in Section 12, Clock Controller.

NOTE

In the header files the module name of the above register is defined as:

SYS.

37.5.7.1 CKSC_IPLL0PIXS_CTL — PLL0PIXCLK clock control register (D1L2(H), D1M1(H), D1M1-V2, D1M2(H) and D1M1A only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5640$_H$
Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0PIXSSTP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 37.7 CKSC_IPLL0PIXS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PLL0PIXSSTP	PLL0PIXCLK stop control 0: Stop PLL0PIXCLK. 1: Activate PLL0PIXCLK.

37.5.7.2 CKSC_IPLL0PIXS_ACT — PLL0PIXCLK clock active register (D1L2(H), D1M1(H), D1M1-V2, D1M2(H) and D1M1A only)

Access: This register can be read in 32-bit units.

Address: FFF8 5648$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0 PIXS ACT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.8 CKSC_IPLL0PIXS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PLL0PIXS ACT	PLL0PIXCLK status 0: PLL0PIXCLK is stopped (default). 1: PLL0PIXCLK i s active.

37.5.7.3 CKDV_IPLL2IND_CTL — PLL2INCLK clock divider register (D1M2(H) only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5040$_H$
Initial value: 0000 0018$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	PLL2INDCSID[4:0]				
Initial value	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 37.9 CKDV_IPLL2IND_CTL register contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When written, write the initial value.
4 to 0	PLL2IND CSID[4:0]	Clock divider setting for PLL2INCLK 0: Disabled 12: PLL0CLK /12 ... 24: PLL0CLK /24 (default) All others: Setting prohibited

37.5.7.4 CKDV_IPLL2IND_STAT — PLL2INCLK clock divider status register (D1M2(H) only)

Access: This register can be read in 32-bit units.

Address: FFF8 5044$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL2IN ACT	PLL2IN SYNC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.10 CKDV_IPLL2IND_STAT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	PLL2IN ACT	PLL2IN clock output active state 0: clock is inactive 1: clock is active
0	PLL2IN SYNC	PLL2INCLK clock divider state 0: output clock does not correspond to the current PLL2IND clock divider 1: output clock corresponds to the current PLL2IND clock divider

37.5.7.5 CKSC_IPLL2INS_CTL — PLL2INCLK source clock selection register (D1M2(H) only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5080$_H$
Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL2INS CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 37.11 CKSC_IPLL2INS_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	PLL2INS CSID[1:0]	Source clock selection for PLL2INCLK 01_B: CKDV_IPLL2IND_CTL output clock 10_B: MainOsc f_X (default) All others: Setting prohibited

37.5.7.6 CKSC_IPLL2INS_ACT — PLL2INCLK source clock active register (D1M2(H) only)

Access: This register can be read in 32-bit units.

Address: FFF8 5088$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL2INSACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.12 CKSC_IPLL2INS_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	PLL2INS ACT[1:0]	Current active IPLL2INS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

37.5.7.7 CKSC_IPLL2CLKS_CTL — PLL2CLK clock control register (D1M2(H) only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5100$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL2CLKSSTP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 37.13 CKSC_IPLL2CLKS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PLL2CLKSSTP	PLL2CLK clock stop control 0: Stop PLL2CLK clock (default). 1: Activate PLL2CLK clock.

37.5.7.8 CKSC_IPLL2CLKS_ACT — PLL2CLK clock active register (D1M2(H) only)

Access: This register can be read in 32-bit units.

Address: FFF8 5108$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL2CLKSACT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.14 CKSC_IPLL2CLKS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PLL2CLKSACT	PLL2CLK status 0: PLL2CLK clock is stopped (default). 1: PLL2CLK clock i s active.

37.5.7.9 CKSC_IDOTCLK0S_CTL — DOTCLK0 source clock selection register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 5940$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DOTCLK0S CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 37.15 CKSC_IDOTCLK0S_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	DOTCLK0S CSID[1:0]	Source clock selection for DOTCLK0 • For D1M2(H) devices: 00_B: Disabled 01_B: PLL0PIXCLK (default) 10_B: PLLFIXCLK 11_B: PLL2CLK • For all other devices: 00_B: Disabled 01_B: PLL0PIXCLK (default) 10_B: PLL1CLK 11_B: Setting prohibited

37.5.7.10 CKSC_IDOTCLK0S_ACT — DOTCLK0 source clock active register

Access: This register can be read in 32-bit units.

Address: FFF8 5948$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DOTCLK0S ACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.16 CKSC_IDOTCLK0S_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	DOTCLK0S ACT[1:0]	Current active IDOTCLK0S source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

37.5.7.11 CKSC_IDOTCLK1S_CTL — DOTCLK1 source clock selection register (D1M2(H), D1M1A only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5980$_H$
Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DOTCLK1S CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 37.17 CKSC_IDOTCLK1S_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	DOTCLK1S CSID[1:0]	Source clock selection for DOTCLK1 00_B: Disabled 01_B: PLL0PIXCLK (default) 10_B: PLLFIXCLK 11_B: PLL2CLK (D1M2(H) only)

37.5.7.12 CKSC_IDOTCLK1S_ACT — DOTCLK1 source clock active register (D1M2(H), D1M1A only)

Access: This register can be read in 32-bit units.

Address: FFF8 5988$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DOTCLK1S ACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.18 CKSC_IDOTCLK1S_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	DOTCLK1S ACT[1:0]	Current active IDOTCLK1S source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

37.5.7.13 CKDV_IDOTCLK0D_CTL — DOTCLK0 clock divider register

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.

For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.

Address: FFF8 59C0$_H$

Initial value: 0000 0010$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DOTCLK0DCSID[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.19 CKSC_IDOTCLK0D_CTL register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When written, write the initial value.
7 to 0	DOTCLK0D CSID[7:0]	Clock divider setting for DOTCLK0 0: Disabled 1: CKSC_IDOTCLK0S_CTL selection /1 (D1M1A only) 2: CKSC_IDOTCLK0S_CTL selection /2 ... 16: CKSC_IDOTCLK0S_CTL selection /16 (default) ... 255: CKSC_IDOTCLK0S_CTL selection /255

37.5.7.14 CKDV_IDOTCLK0D_STAT — DOTCLK0 clock divider status register

Access: This register can be read in 32-bit units.

Address: FFF8 59C4$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DOTCLK0 ACT	DOTCLK0 SYNC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.20 CKDV_IDOTCLK0D_STAT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	DOTCLK0 ACT	DOTCLK0 clock output active state 0: clock is inactive 1: clock is active
0	DOTCLK0 SYNC	DOTCLK0 clock divider state 0: output clock does not correspond to the current DOTCLK0D clock divider 1: output clock corresponds to the current DOTCLK0D clock divider

37.5.7.15 CKDV_IDOTCLK1D_CTL — DOTCLK1 clock divider register (D1M2(H), D1M1A only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5A00$_H$
Initial value: 0000 0010$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DOTCLK1DCSID[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.21 CKSC_IDOTCLK1D_CTL register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When written, write the initial value.
7 to 0	DOTCLK1D CSID[7:0]	Clock divider setting for DOTCLK1 0: Disabled 1: CKSC_IDOTCLK1S_CTL selection /1 (D1M1A only) 2: CKSC_IDOTCLK1S_CTL selection /2 ... 16: CKSC_IDOTCLK1S_CTL selection /16 (default) ... 255: CKSC_IDOTCLK1S_CTL selection /255

37.5.7.16 CKDV_IDOTCLK1D_STAT — DOTCLK1 clock divider status register (D1M2(H), D1M1A only)

Access: This register can be read in 32-bit units.

Address: FFF8 5A04$_H$

Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DOT CLK1 ACT	DOT CLK1 SYNC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.22 CKDV_IDOTCLK1D_STAT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1	DOTCLK1 ACT	DOTCLK1 clock output active state 0: clock is inactive 1: clock is active
0	DOTCLK1 SYNC	DOTCLK1 clock divider state 0: output clock does not correspond to the current DOTCLK1D clock divider 1: output clock corresponds to the current DOTCLK1D clock divider

37.5.7.17 CKSC_IVDCE0VOS_CTL — C_ISO_VDCE0CLK source clock selection register (D1M1(H), D1M1-V2, D1M2(H) and D1M1A only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5880$_H$
Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	VDCE0VOSCSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 37.23 CKSC_IVDCE0VOS_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	VDCE0VOS CSID[2:0]	Source clock selection for C_ISO_VDCE0CLK 000_B: Disabled 001_B: DOTCLK0 (default) 010_B: DOTCLK0 /4 (D1M2(H) only) 011_B: Video Input interface 0 pixel clock C_ISO_V0PIXCLK 100_B: Video Input interface 1 pixel clock C_ISO_V1PIXCLK (D1M2(H) only) 101_B: DOTCLK0 /7 (D1M1A only) All others: Setting prohibited

CAUTION

For D1M2(H) only:
If the RSDS video output is used, set "VDCE0VOSCSID[2:0] = 010_B: DOTCLK0 /4".

For D1M1A only:
If the LVDS video output is used, set "VDCE0VOSCSID[2:0] = 101_B: DOTCLK0 /7".

37.5.7.18 CKSC_IVDCE0VOS_ACT — C_ISO_VDCE0CLK source clock active register (D1M1(H), D1M1-V2, D1M2(H) and D1M1A only)

Access: This register can be read in 32-bit units.

Address: FFF8 5888$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	VDCE0VOSACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.24 CKSC_IVDCE0VOS_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	VDCE0VOS ACT[2:0]	Current active IVDCE0VOS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

37.5.7.19 CKSC_IVDCE1VOS_CTL — C_ISO_VDCE1CLK source clock selection register (D1M2(H), D1M1A only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 58C0$_H$
Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	VDCE1VOSCSID[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 37.25 CKSC_IVDCE1VOS_CTL register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	VDCE1VOS CSID[2:0]	Source clock selection for C_ISO_VDCE1CLK 000$_B$: Disabled 001$_B$: DOTCLK1 (default) 010$_B$: DOTCLK1 /4 (D1M2(H) only) 011$_B$: Video Input interface 1 pixel clock C_ISO_V1PIXCLK (D1M2(H) only) 100$_B$: Video Input interface 0 pixel clock C_ISO_V0PIXCLK 101$_B$: DOTCLK1 /7 (D1M1A only) All others: Setting prohibited

CAUTION

For D1M2(H) only:
If the RSDS video output is used, set "VDCE1VOSCSID[2:0] = 010$_B$: DOTCLK1 /4".

For D1M1A only:
If the LVDS video output is used, set "VDCE1VOSCSID[2:0] = 101$_B$: DOTCLK1 /7".

37.5.7.20 CKSC_IVDCE1VOS_ACT — C_ISO_VDCE1CLK source clock active register (D1M2(H), D1M1A only)

Access: This register can be read in 32-bit units.

Address: FFF8 58C8$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	VDCE1VOSACT[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.26 CKSC_IVDCE1VOS_ACT register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	VDCE1VOS ACT[2:0]	Current active IVDCE1VOS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

37.5.7.21 CKSC_IVOEXS_CTL — Video output pixel clocks exchange register (D1M2(H), D1M1A only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5900$_H$
Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VOEXSCSID0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 37.27 CKSC_IVOEXS_CTL register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	VOEXS CSID0	For D1M2(H) Clock selection for VDCE0_VO_CLK and VDCE1_VO_CLK 0: Disable exchange (default) – VDCE0_VO_CLK = C_ISO_VDCE0CLK – VDCE1_VO_CLK = C_ISO_VDCE1CLK 1: Enable exchange – VDCE1_VO_CLK = C_ISO_VDCE0CLK – VDCE0_VO_CLK = C_ISO_VDCE1CLK For D1M1A: • OpenLDI use case 0: OLDICLK = DOTCLK0 1: OLDICLK = DOTCLK1 **Note:** OLDICLK is the operation clock for OpenLDI (maximum is 240 MHz). In the OpenLDI use case, the pixel clock for OpenLDI is also selected by CKSC_IVOEXS_CTL. For example, when CKSC_IVOEXS_CTL is 0, OLDICLK is DOTCLK0. And, pixel clock for OpenLDI is from CKSC_IVDCE0VOS_CTL. • Other use cases 0: Disable exchange – IVOEXS0_OUTCLK = DOTCLK0 – VDCE1_VO_CLK = DOTCLK1 1: Enable exchange – IVOEXS0_OUTCLK = DOTCLK1 – VDCE1_VO_CLK = DOTCLK0

37.5.7.22 CKSC_IVOEXS_ACT — Video output pixel clocks exchange active register (D1M2(H), D1M1A only)

Access: This register can be read in 32-bit units.

Address: FFF8 5908$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VOEXSACT0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.28 CKSC_IVOEXS_ACT register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	VOEXS ACT0	Current active IVOEXS clock selection

37.5.7.23 CKSC_IRSDSS_CTL — C_ISO_RSDSCLK source clock selection register (D1M2(H) only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5A40$_H$
Initial value: 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSDSSCSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 37.29 CKSC_IRSDSS_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	RSDSS CSID[1:0]	Source clock selection for C_ISO_RSDSCLK 01_B: DOTCLK0 10_B: DOTCLK1 (default) All others: Setting prohibited

37.5.7.24 CKSC_IRSDSS_ACT — C_ISO_RSDSCLK source clock active register (D1M2(H) only)

Access: This register can be read in 32-bit units.

Address: FFF8 5A48$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSDSSACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.30 CKSC_IRSDSS_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	RSDSS ACT[1:0]	Current active IRSDSS clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

37.5.7.25 CKSC_IVDCE0VIS_CTL — C_ISO_VI0PIXCLK source clock selection register (D1M1(H), D1M1-V2, D1M2(H) and D1M1A only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5840$_H$
Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDCE0VIS CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 37.31 CKSC_IVDCE0VIS_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	VDCE0VIS CSID[1:0]	Source clock selection for C_ISO_VI0PIXCLK 01_B: Port VDCE0_VI_CLK (default) 10_B: C_ISO_MIPIPIXCLK (D1M2(H)) DOTCLK0 (D1M1(H), D1M1-V2 and D1M1A) All others: Setting prohibited

37.5.7.26 CKSC_IVDCE0VIS_ACT — C_ISO_VI0PIXCLK source clock active register (D1M1(H), D1M1-V2, D1M2(H) and D1M1A only)

Access: This register can be read in 32-bit units.

Address: FFF8 5848$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VDCE0VIS ACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.32 CKSC_IVDCE0VIS_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	VDCE0VIS ACT[1:0]	Current active IVDCE0VIS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

37.5.7.27 CKSC_IMIPIPLLS_CTL — MIPIPLLCLK source clock selection register (D1M2(H) only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 5680$_H$
Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MIPIPLLS CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 37.33 CKSC_IMIPIPLLS_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	MIPIPLLS CSID[1:0]	Source clock selection for MIPIPLLCLK 01_B: PLL0PIXCLK (default) 10_B: PLLFIXCLK All others: Setting prohibited

37.5.7.28 CKSC_IMIPIPLLS_ACT — MIPIPLLCLK source clock active register (D1M2(H) only)

Access: This register can be read in 32-bit units.

Address: FFF8 5688$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MIPIPLLS ACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.34 CKSC_IMIPIPLLS_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	MIPIPLLS ACT[1:0]	Current active IMIPIPLLS source clock selection 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

37.5.7.29 CKSC_IMIPIPIXD_CTL — C_ISO_MIPIPIXCLK clock divider register (D1M2(H) only)

Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDD1.
For details, refer to **Section 4, Write-Protected Registers**.

Access: This register can be read/written in 32-bit units.
Address: FFF8 56C0$_H$
Initial value: 0000 0001$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MIPIPIXD CSID[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 37.35 CKSC_IMIPIPIXD_CTL register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	MIPIPIXD CSID[1:0]	Clock divider setting for C_ISO_MIPIPIXCLK 00_B: Disabled 01_B: MIPIPLLCLK /3 (default) 10_B: MIPIPLLCLK /6 11_B: MIPIPLLCLK /12

37.5.7.30 CKSC_IMIPIPIXD_ACT — C_ISO_MIPIPIXCLK clock divider active register (D1M2(H) only)

Access: This register can be read in 32-bit units.

Address: FFF8 56C8$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MIPIPIXD ACT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.36 CKSC_IMIPIPIXD_ACT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1 to 0	MIPIPIXD ACT[1:0]	Current active IMIPIPIXD clock divider 0: selected clock inactive All others: selected and active clock

NOTE

As long as the clock, selected by the source clock selection register, is not active, this source clock active register remains 0.
Its shows the correct value, when the selected clock becomes active.

37.5.8 Video Input Pixel Clock Monitoring

The video input pixel clocks are supervised by the Clock Monitors CLMA5 and CLMA6.

Table 37.37 Video Input pixel Clock Monitors

Module	Clock	Connected to
CLMA5:		
CLMATSMP	CLMA5 sampling clock	Low Speed IntOsc f_{RL} (240 kHz)
CLMATMON	CLMA5 monitored clock	C_ISO_VI0PIXCLK (Video Input Interface 0 pixel clock)
CLMA6 (D1M2(H) only):		
CLMATSMP	CLMA6 sampling clock	Low Speed IntOsc f_{RL} (240 kHz)
CLMATMON	CLMA6 monitored clock	C_ISO_VI1PIXCLK (Video Input Interface 1 pixel clock)

In case of a pixel clock fail the clock monitor generates an error signal, that is input to the Error Control Module.

For the functional description of the Clock Monitors CLMA5 and CLMA6 refer to **Section 12.7, Clock Monitor A (CLMA)**.

37.6 Sprite Engine registers update control

The sprite definition registers of the Sprite Engine are updated, i.e. new sprite definition values become effective, with internal vertical enable from VDCE and software control, depending which signal the update for a certain sprite is assigned to.
Refer to “Sprite definition registers modification” in Section 43, Sprite Engine (SPEA) for details.

The following diagram shows how the Sprite Engine VUPDATEn (n=0,1) inputs are generated.

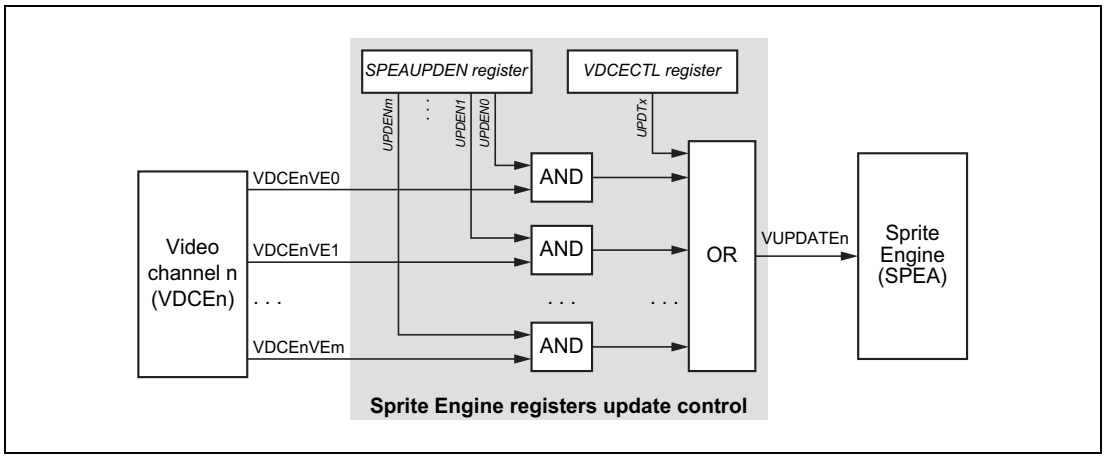


Figure 37.15 Sprite Engine registers update control for video channel n

Each stage of a video channel n issues vertical enable signals VDCEnVEm, which indicate active data processing of an image synthesizer stage.

These signals turn active with the first visible line of image data displayed by a layer and turns inactive after the last visible line has been displayed by that layer.

The Sprite Engine’s VUPDATEn input changes to low level when all VDCEnVEm signals turned to inactive (low level). This triggers the update of the sprite registers.

Each VDCEnVEm signal can be masked, i.e. disregarded, via mask bits in the Sprite Engine update timing control register SPEAUPDEN:

- UPDENm = 0: VDCEnVEm does not impact VUPDATEn signal generation
- UPDENm = 1: VDCEnVEm impacts VUPDATEn signal generation

The following table lists all available VDCEnVEm signals.

Table 37.38 VDCEnVEm signals (1/2)

VDCEn	VDCEnVEm signal		Mask bit
VDCE0	VDCE0VE0	VE of Image Synthesizer 00	SPEAUPDEN.UPDEN0
	VDCE0VE1	VE of Image Synthesizer 01	SPEAUPDEN.UPDEN1
	VDCE0VE2	VE of Image Synthesizer 02	SPEAUPDEN.UPDEN2
	VDCE0VE3	VE of Image Synthesizer 03	SPEAUPDEN.UPDEN3
	VDCE0VE4	VE of Output Image Generator	SPEAUPDEN.UPDEN4
	VDCE0VE5	VE after alpha blending of Image Synthesizer 02 and Image Synthesizer 03	SPEAUPDEN.UPDEN5

Table 37.38 VDCEnVEm signals (2/2)

VDCEn	VDCEnVEm signal		Mask bit
VDCE1 (D1M1A, D1M2(H) only)	VDCE1VE0	VE of Image Synthesizer 10	SPEAUPDEN.UPDEN8
	VDCE1VE1	VE of Image Synthesizer 11	SPEAUPDEN.UPDEN9
	VDCE1VE2	VE of Image Synthesizer 12	SPEAUPDEN.UPDEN10
	VDCE1VE3	VE of Image Synthesizer 13	SPEAUPDEN.UPDEN11
	VDCE1VE4	VE after alpha blending of Image Synthesizer 12 and Image Synthesizer 13	SPEAUPDEN.UPDEN12

Hardware and software update

By use of the VDCECTL.UPDTn bit and the mask bits SPEAUPDEN.UPDENm it can be selected whether the Sprite Engine registers are update by hardware, i.e. by the VDCEnVEm signals, or by the application program.

- Hardware update: VDCECTL.UPDTn = 0

  SPEAUPDEN.UPDENm = 1 selects the VDCEnVEm signals to generate VUPDATEn

- Software update: all SPEAUPDEN.UPDENm = 0

  The VSYNCn signal is directly controlled by the VDCECTL.UPDTn bit.
  Following procedure triggers the Sprite Engine registers update:

  – Set UPDTn = 1: VUPDATEn → high
  – Set UPDTn = 0: VUPDATEn → low: triggers Sprite Engine registers update.

37.7 Video Input selection

Depending on the data format of the video input signals and the ports, which are used to connect these, several control registers have to be set:

- correct alternative port function by use of the port control registers PFCn and PFCEn
- VDCE0: video input selection via VDCECTL.PXSL and VDCECTL.VISL0
- VDCE1: video input selection via VDCECTL.VISL1

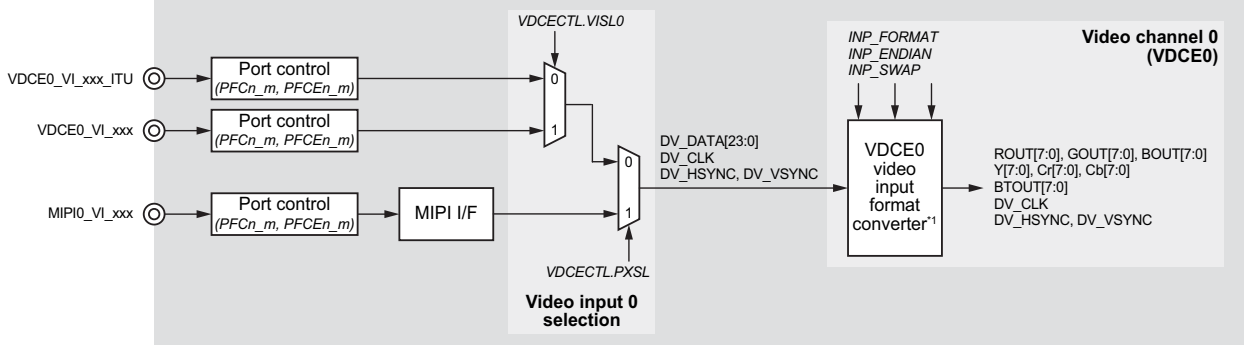
Refer to Section 2, Pins for detailed information about alternative port functions.

The tables in the following sections summarize the correct selection of the video input data format.

37.7.1 Video input channel 0 selection

Table 37.39 Video input channel 0 selection

Alternative port function	2nd alternative input (PFCn_m = 1, PFCEn_m = 0)	3rd alternative input (PFCn_m = 0, PFCEn_m = 1)		MIPI
Input ports	VDCE0_VI_xxx_ITU	VDCE0_VI_xxx_ITU	VDCE0_VI_xxx	MIPI0_VI_xxx
Data format	BT656, BT601	BT656, BT601	RGB666, RGB565, YCbCr422, BT656, BT601	RGB888, RGB565, YCbCr422
VISL0	0		1	–
PXSL	0			1



Note 1. The Video Data Controllers VDCEn include facilities to change the mapping of the color input data DV_DATA[23:0] to the VDCEn input data ROUT[7:0], GOUT[7:0], BOUT[7:0], BTOUT[7:0], Y[7:0], Cr[7:0] and Cb[7:0]. Refer to Section 38, Video Data Controller E (VDCE) for details about the INP_FORMAT, INP_ENDIAN and INP_SWAP settings.

Figure 37.16 Video channel 0 video input selections

The following table shows the assignment between the various video input signals for following settings:

- VDCEn video input format converter settings:
  - INP_EXT_SYNC_CNT.INP_ENDIAN_ON = 0
  - INP_EXT_SYNC_CNT.INP_SWAP_ON = 0

  For other bit allocations of DV_DATA[23:0] to ROUT[7:0], GOUT[7:0], BOUT[7:0], Y[7:0], Cr[7:0], Cb[7:0] and BTOUT[7:0] refer to Section 38, Video Data Controller E (VDCE).
- Video input selection:

– VDCECTL.PXSL = 0: input via VDCE0_VI signals (no MIPI)

Table 37.40 Video channel 0 input format selection examples

	VDCE0 video input format converter output			
	INP_FORMAT[2:0] = 001_B	INP_FORMAT[2:0] = 010_B	INP_FORMAT[2:0] = 011_B, 100_B	INP_FORMAT[2:0] = 101_B
Port signal name	**RGB666**	**RGB565**	**ITU formats**	**YCrCb422**
Color data signals				
VDCE0_VI_DATA17	ROUT[7]	–	–	–
VDCE0_VI_DATA16	ROUT[6]	–	–	–
VDCE0_VI_DATA15	ROUT[5]	ROUT[7]	–	Y[7]
VDCE0_VI_DATA14	ROUT[4]	ROUT[6]	–	Y[6]
VDCE0_VI_DATA13	ROUT[3]	ROUT[5]	–	Y[5]
VDCE0_VI_DATA12	ROUT[2]	ROUT[4]	–	Y[4]
VDCE0_VI_DATA11	GOUT[7]	ROUT[3]	–	Y[3]
VDCE0_VI_DATA10	GOUT[6]	GOUT[7]	–	Y[2]
VDCE0_VI_DATA9	GOUT[5]	GOUT[6]	–	Y[1]
VDCE0_VI_DATA8	GOUT[4]	GOUT[5]	–	Y[0]
VDCE0_VI_DATA7	GOUT[3]	GOUT[4]	BTOUT[7]	Cb[7] / Cr[7]
VDCE0_VI_DATA6	GOUT[2]	GOUT[3]	BTOUT[6]	Cb[6] / Cr[6]
VDCE0_VI_DATA5	BOUT[7]	GOUT[2]	BTOUT[5]	Cb[5] / Cr[5]
VDCE0_VI_DATA4	BOUT[6]	BOUT[7]	BTOUT[4]	Cb[4] / Cr[4]
VDCE0_VI_DATA3	BOUT[5]	BOUT[6]	BTOUT[3]	Cb[3] / Cr[3]
VDCE0_VI_DATA2	BOUT[4]	BOUT[5]	BTOUT[2]	Cb[2] / Cr[2]
VDCE0_VI_DATA1	BOUT[3]	BOUT[4]	BTOUT[1]	Cb[1] / Cr[1]
VDCE0_VI_DATA0	BOUT[2]	BOUT[3]	BTOUT[0]	Cb[0] / Cr[0]
Pixel clock and synchronization signals				
VDCE0_VI_CLK	DV_CLK			
VDCE0_VI_HSYNC	DV_HSYNC			
VDCE0_VI_VSYNC	DV_VSYNC			

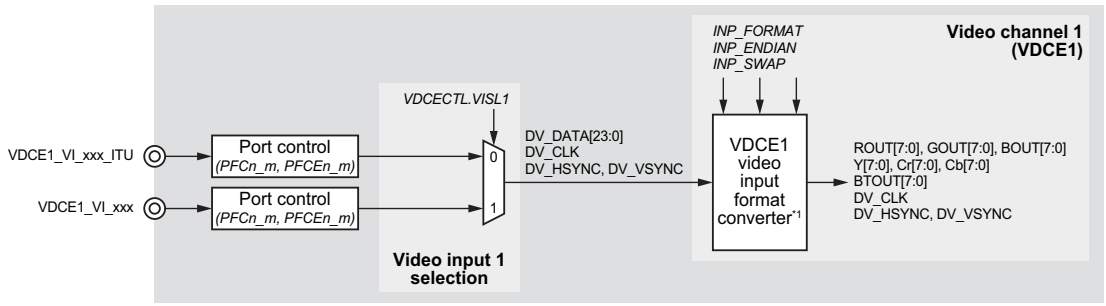
NOTE

If the MIPI interface is used for video data input to video channel 0 (VDCECTL.PXSL = 1) the separate MIPI ports MIPI0_VI_xxx are used. The mapping of the MIPI interface output to the VDCE0 video input format converter is fixed and occupies all 24 bit DV_DATA[23:0] for the RGB888 format.

37.7.2 Video input channel 1 selection (D1M2(H) only)

Table 37.41 Video input channel 1 selection

Alternative port function	2nd alternative input (PFCn_m = 1, PFCEn_m = 0)	1st alternative input (PFCn_m = 0, PFCEn_m = 0)
Input ports	VDCE1_VI_xxx_ITU	VDCE1_VI_xxx
Data format	BT656, BT601	RGB888, RGB666, RGB565, YCbCr444, YCbCr422, BT656, BT601
VISL1	0	1



Note 1. The Video Data Controllers VDCEn include facilities to change the mapping of the color input data DV_DATA[23:0] to the VDCEn input data ROUT[7:0], GOUT[7:0], BOUT[7:0], BTOUT[7:0], Y[7:0], Cr[7:0] and Cb[7:0]. Refer to Section 38, Video Data Controller E (VDCE) for details about the INP_FORMAT, INP_ENDIAN and INP_SWAP settings.

Figure 37.17 Video channel 1 video input selections

The following table shows the assignment between the various video input signals for following settings:

- VDCEn video input format converter settings:
  - INP_EXT_SYNC_CNT.INP_ENDIAN_ON = 0
  - INP_EXT_SYNC_CNT.INP_SWAP_ON = 0

  For other bit allocations of DV_DATA[23:0] to ROUT[7:0], GOUT[7:0], BOUT[7:0], Y[7:0], Cr[7:0], Cb[7:0] and BTOUT[7:0] refer to Section 38, Video Data Controller E (VDCE).

Table 37.42 Video channel 1 input format selection examples (1/2)

	VDCE1 video input format converter output				
	INP_FORMAT[2:0] = 000_B	INP_FORMAT[2:0] = 001_B	INP_FORMAT[2:0] = 010_B	INP_FORMAT[2:0] = 011_B, 100_B	INP_FORMAT[2:0] = 101_B
Port signal name	RGB888, YCrCb444	RGB666	RGB565	ITU formats	YCrCb422
Color data signals					
VDCE1_VI_DATA23	ROUT[7] / Cr[7]	–	–	–	–
VDCE1_VI_DATA22	ROUT[6] / Cr[6]	–	–	–	–
VDCE1_VI_DATA21	ROUT[5] / Cr[5]	–	–	–	–
VDCE1_VI_DATA20	ROUT[4] / Cr[4]	–	–	–	–
VDCE1_VI_DATA19	ROUT[3] / Cr[3]	–	–	–	–
VDCE1_VI_DATA18	ROUT[2] / Cr[2]	–	–	–	–
VDCE1_VI_DATA17	ROUT[1] / Cr[1]	ROUT[7]	–	–	–

Table 37.42 Video channel 1 input format selection examples (2/2)

	VDCE1 video input format converter output				
	INP_FORMAT[2:0] = 000_B	INP_FORMAT[2:0] = 001_B	INP_FORMAT[2:0] = 010_B	INP_FORMAT[2:0] = 011_B, 100_B	INP_FORMAT[2:0] = 101_B
Port signal name	RGB888, YCrCb444	RGB666	RGB565	ITU formats	YCrCb422
VDCE1_VI_DATA16	ROUT[0] / Cr[0]	ROUT[6]	–	–	–
VDCE1_VI_DATA15	GOUT[7] / Y[7]	ROUT[5]	ROUT[7]	–	Y[7]
VDCE1_VI_DATA14	GOUT[6] / Y[6]	ROUT[4]	ROUT[6]	–	Y[6]
VDCE1_VI_DATA13	GOUT[5] / Y[5]	ROUT[3]	ROUT[5]	–	Y[5]
VDCE1_VI_DATA12	GOUT[4] / Y[4]	ROUT[2]	ROUT[4]	–	Y[4]
VDCE1_VI_DATA11	GOUT[3] / Y[3]	GOUT[7]	ROUT[3]	–	Y[3]
VDCE1_VI_DATA10	GOUT[2] / Y[2]	GOUT[6]	GOUT[7]	–	Y[2]
VDCE1_VI_DATA9	GOUT[1] / Y[1]	GOUT[5]	GOUT[6]	–	Y[1]
VDCE1_VI_DATA8	GOUT[0] / Y[0]	GOUT[4]	GOUT[5]	–	Y[0]
VDCE1_VI_DATA7	BOUT[7] / Cb[7]	GOUT[3]	GOUT[4]	BTOUT[7]	Cb[7] / Cr[7]
VDCE1_VI_DATA6	BOUT[6] / Cb[6]	GOUT[2]	GOUT[3]	BTOUT[6]	Cb[6] / Cr[6]
VDCE1_VI_DATA5	BOUT[5] / Cb[5]	BOUT[7]	GOUT[2]	BTOUT[5]	Cb[5] / Cr[5]
VDCE1_VI_DATA4	BOUT[4] / Cb[4]	BOUT[6]	BOUT[7]	BTOUT[4]	Cb[4] / Cr[4]
VDCE1_VI_DATA3	BOUT[3] / Cb[3]	BOUT[5]	BOUT[6]	BTOUT[3]	Cb[3] / Cr[3]
VDCE1_VI_DATA2	BOUT[2] / Cb[2]	BOUT[4]	BOUT[5]	BTOUT[2]	Cb[2] / Cr[2]
VDCE1_VI_DATA1	BOUT[1] / Cb[1]	BOUT[3]	BOUT[4]	BTOUT[1]	Cb[1] / Cr[1]
VDCE1_VI_DATA0	BOUT[0] / Cb[0]	BOUT[2]	BOUT[3]	BTOUT[0]	Cb[0] / Cr[0]
Pixel clock and synchronization signals					
VDCE1_VI_CLK	DV_CLK				
VDCE1_VI_HSYNC	DV_HSYNC				
VDCE1_VI_VSYNC	DV_VSYNC				

37.8 D1M2(H) MIPI Video Input Interface (MIPI)

37.8.1 Overview of the RH850/D1L/D1M MIPI Video Input Interface

37.8.1.1 Number of Units

This microcontroller has the following number of units of the MIPI.

Each MIPI unit has one channel interface. “Number of channels” is used with the same meaning as “number of units” in this section.

Table 37.43 Number of Units

Product Name	D1L1	D1L2(H)	D1M1(H) D1M1-V2 D1M1A	D1M2(H)
Units	–	–	–	1
Names	–	–	–	MIPIn (n = 0)

Table 37.44 Index

Index	Meaning
n	Throughout this section, the individual MIPI units are identified by the index “n” (n = 0): for example, MIPInMODE is the MIPIn mode register.

37.8.1.2 Register Base Addresses

MIPI base addresses are listed in the following table.

MIPI register addresses are given as offsets from the base addresses in general.

Table 37.45 Register Base Address

Base Address Name	Base Address
<MIPI0_base>	FFFD 3400_H

37.8.1.3 Clock supply

The MIPI clock supply is shown in the following table.

Table 37.46 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
MIPI0	MIPIPCLK	Clock Controller C_ISO_PCLK
	MIPIADCLK	Clock Controller C_ISO_MIPIADCLK
	MIPIPIXCLK	Clock Controller C_ISO_MIPIPIXCLK

37.8.1.4 Interrupts

MIPI interrupt requests are listed in the following table.

Table 37.47 MIPIn interrupt requests

MIPIn signals	Function	Connected to
MIPI0:		
	MIPI buffer overflow interrupt	Interrupt Controller INTMIPI0OVF
	MIPI controller interrupt	Interrupt Controller INTMIPI0CTL

37.8.1.5 Reset sources

MIPI reset sources are listed in the following table. MIPI is initialized by these reset sources.

Table 37.48 Reset sources

Unit Name	Reset Source
MIPI0	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

37.8.1.6 External Input/Output Signals

External input/output signals of MIPI are listed below.

Table 37.49 I/O signals connections

Unit Signal Name	Outline	Alternative Port Pin Signal
MIPI0:		
MIPI0_VI_DATA0P	Data lane 0 signals	Port MIPI0_VI_DATA0P
MIPI0_VI_DATA0N		Port MIPI0_VI_DATA0N
MIPI0_VI_DATA1P	Data lane 1 signals	Port MIPI0_VI_DATA1P
MIPI0_VI_DATA1N		Port MIPI0_VI_DATA1N
MIPI0_VI_CLKP	Clock lane signals	Port MIPI0_VI_CLKP
MIPI0_VI_CLKN		Port MIPI0_VI_CLKN

37.8.2 Functional Overview

The MIPI Video Input Interface captures video data from a MIPI CSI-2 compliant Camera Serial Interface and passes the video data on to the Video Input Interface of the Video Data Controller.

Features summary

- Input and output color formats: 16-bit YUV422, 24-bit RGB888, 16-bit RGB565
- Maximum frame size: 1024 x 1024 pixels (24-bit color)
- Maximum frame rate: 60 frames/s
- MIPI CSI-2 receiver
- Lane module type:

  - MIPI CSI-2 receiver
  - The external MIPI CSI-2 master can send trigger events over the MIPI interface, which can issue interrupts.
- Data lane number: up to two lanes
- Data transfer bandwidth
  - Maximum data clock: max. 240 MHz in DDR mode
  - Transfer speed: max. 480 Mbit/s per lane, max. 960 Mbit/s in total
- Data consistency checks
  - ECC check on MIPI packet header (single bit errors correction, double bit errors detection)
  - CRC check on data packets
- Video output signal to the Video Data Controller
  - Regular video timing with HSYNC and VSYNC
  - Horizontal blank period: fixed to 16 pixel clock cycles
  - Burst output via two 4 KB output buffers
- Following MIPI CSI-2 features are not supported by this implementation of the MIPI Video Input Interface:
  - Virtual channel: This feature is not supported.
  - Embedded data: Transmitting embedded data to the Video Data Controller VDCE0 is not supported.
  - Low power data transfer (LPDT): This feature is not supported.

37.8.3 Functional Description

The following figure provides a functional block diagram of the MIPI Video Input Interface.

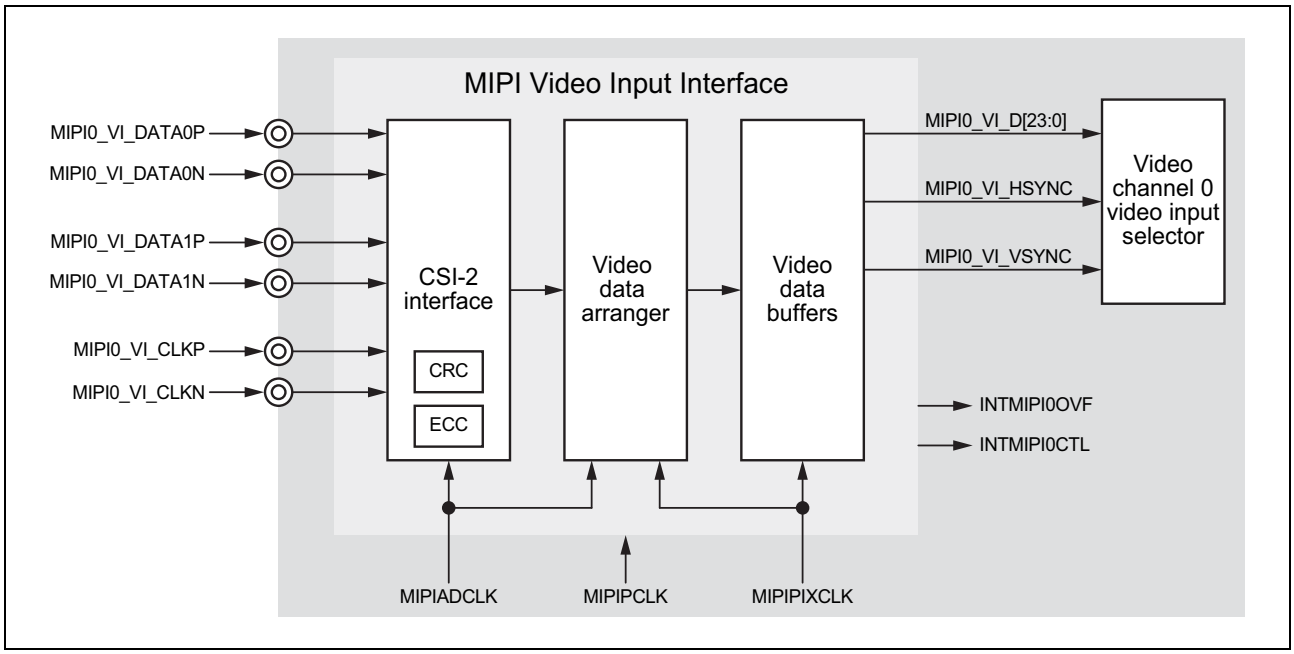


Figure 37.18 MIPI Video Input Interface block diagram

The video data, acquired via the CSI-2 interface, is arranged to be forwarded to the video input selector for the video channel 0 of the Video Data Controller.

The color format of the video data is passed through to the video channel 0, i.e. the color format is not changed.

The video signals (MIPI0_VI_D[23:0], MIPI0_VI_HSYNC and MIPI0_VI_VSYNC) are passed to the video input selector for the video channel 0 of the Video Data Controller via video data buffers.

The clock for the video channel 0 signals (MIPI0_VI_D[23:0], MIPI0_VI_HSYNC and MIPI0_VI_VSYNC) is the MIPIPIXCLK..
Refer to Section 37.7, Video Input selection for details about the video input selector.

37.8.4 MIPI Video Input Interface registers

The MIPI Video Input Interface is controlled and operated by the following registers:

For details on <MIPIn_base>, see Section 37.8.1.2, Register Base Addresses.

Table 37.50 MIPI register overview

Register name	Shortcut	Address
Operation control register	MIPInON	<MIPIn_base> + 00_H
Mode register	MIPInMODE	<MIPIn_base> + 04_H
Data delay register	MIPInDATA_DLY_CTL	<MIPIn_base> + 08_H
Reset register	MIPInRST_CTL	<MIPIn_base> + $0C_H$
SubLVDS buffer control	MIPInBUF_CTL	<MIPIn_base> + 10_H
$T_{CLK-SETTLE}$ and $T_{HS-SETTLE}$ register	MIPInSOT_COUNT	<MIPIn_base> + 14_H
PHY layer status register	MIPInRX_STATE	<MIPIn_base> + 18_H
Long packet word count definition register	MIPInWORD_COUNT	<MIPIn_base> + $1C_H$
LP buffer and noise removal timing register	MIPInLP_EN_ON_WC	<MIPIn_base> + 20_H
Line blanking period register	MIPInLINE_BLANK	<MIPIn_base> + 28_H
LP11 reset delay adjustment register	MIPInRESET_DLY_CTL0	<MIPIn_base> + $2C_H$
Interrupt status register	MIPInINTSTATUS	<MIPIn_base> + 40_H
Interrupt enable set register	MIPInINTENSET	<MIPIn_base> + 48_H
Interrupt enable clear register	MIPInINTENCLR	<MIPIn_base> + $4c_H$
Interrupt factor clear register	MIPInINTFFCLR	<MIPIn_base> + 50_H
T_{EOT} setting	MIPInEOT_COUNT	<MIPIn_base> + 74_H
Video input color format register	MIPInVIN_MODE	<MIPIn_base> + $9C_H$

37.8.4.1 MIPInON – Operation control register

This register controls start and stop of the MIPI interface.

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + 00_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIPI_ON
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 37.51 MIPInON register contents

Bit position	Bit name	Function
31 to 1	Reserved	These bits are always read as 0. When written, write the initial value.
0	MIPI_ON	MIPI interface start/stop control 0: Stop the MIPI interface immediately 1: Start the MIPI interface immediately

37.8.4.2 MIPInMODE – Mode register

This register controls the mode of the MIPI interface.

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + 04_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	DEL_LP11_NOISE	LANE_NUM[1:0]		0	SEL_PLIF	RSTN_CLK_ON	RSTN_DATA_ON	SOT_CLKSEL	0	0	0	0	DATA_TYPE[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	VTIM	0	0	ECC_EN	CRC_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W

Table 37.52 MIPInMODE register contents (1/2)

Bit position	Bit name	Function
31 to 30	Reserved	These bits are always read as 0. When written, write the initial value.
29	DEL_LP11_NOISE	Noise filter for clock line. If this is switched on, noise occurring during transition from HS state to LP11 state is removed. 0: Noise removal disabled 1: Noise removal enabled
28 to 27	LANE_NUM[1:0]	Number of lanes. 00_B: 1 lane 01_B: 2 lanes All others: Setting prohibited
26	Reserved	These bits are always read as 0. When written, write the initial value.
25	SEL_PLIF	Select PHY layer interface synchronization method 0: Select FIFO method 1: Setting prohibited
24	RSTN_CLK_ON	Reset enable for clock lane for $T_{CLK\text{-}SETTLE}$ and T_{EOT} When this bit is 1(enable), the reset of clock lane is controlled after $T_{CLK\text{-}SETTLE}$ and T_{EOT}. This reset is available for noise removal from HS buffer during LP mode. This bit must be set to 1 before the MIPI transmission is enabled by MIPInON.MIPI_ON = 1. 0: Disable (prohibited when MIPInON.MIPI_ON = 1) Clock lane is not controlled after $T_{CLK\text{-}SETTLE}$ and T_{EOT}. Also, clock lane is not reset during LP mode. 1: Enable The reset of clock lane is released after $T_{CLK\text{-}SETTLE}$. $T_{CLK\text{-}SETTLE}$ is defined in MIPInSOT_COUNT.CLK_COUNT[7:0]. The reset of clock lane is activated after T_{EOT}. T_{EOT} is defined in MIPInEOT_COUNT.EOT_COUNT[7:0].

Table 37.52 MIPInMODE register contents (2/2)

Bit position	Bit name	Function
23	RSTN_ DATA_ON	Reset enable for data lane for $T_{HS-SETTLE}$ and T_{EOT} When this bit is 1(enable), the reset of data lane is controlled after $T_{HS-SETTLE}$ and T_{EOT}. This reset is available for noise removal from HS buffer during LP mode. This bit must be set to 1 before the MIPI transmission is enabled by MIPInON.MIPI_ON = 1. 0: Disable (prohibited when MIPInON.MIPI_ON = 1) Data lane is not controlled after $T_{HS-SETTLE}$ and T_{EOT}. Also, data lane is not reset during LP mode. 1: Enable The reset of data lane is released after $T_{HS-SETTLE}$. $T_{HS-SETTLE}$ is defined in MIPInSOT_COUNT.DATA_COUNT[7:0]. The reset of data lane is activated after T_{EOT}. T_{EOT} is defined in MIPInEOT_COUNT.EOT_COUNT[7:0].
22	SOT_CLKSEL	Select the clock that counts $T_{HS-SETTLE}$ 0: MIPIADCLK 1: External clock divided by 4
21 to 18	Reserved	These bits are always read as 0. When written, write the initial value.
17 to 16	DATA_ TYPE[1:0]	Input format specification 10_B: RAW8 All others: Setting prohibited
15 to 5	Reserved	These bits are always read as 0. When written, write the initial value.
4	VTIM	Timing of MIPI0_VI_VSYNC changes for using VDCE vertical position shifting 0: Enabled 1: Disabled
3 to 2	Reserved	This bit is always read as 0. When written, write the initial value.
1	ECC_EN	ECC check and correction of single bit errors of the header 0: Disabled 1: Enabled
0	CRC_EN	CRC check of the packet data 0: Disabled 1: Enabled

37.8.4.3 MIPInDATA_DLY_CTL – Data delay register

The register is for input data delay setting.

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + 08_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	DLY_LANE1[2:0]			DLY_LANE0[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.53 MIPInDATA_DLY_CTL register contents

Bit position	Bit name	Function
31 to 6	Reserved	These bits are always read as 0. When written, write the initial value.
5 to 3	DLY_ LANE1[2:0]	Delay adjustment for data lane 1 Delay time = DLY_LANE1[2:0] x 100 ps
2 to 0	DLY_ LANE0[2:0]	Delay adjustment for data lane 0 Delay time = DLY_LANE0[2:0] x 100 ps

37.8.4.4 MIPInRST_CTL – Reset register

This register is used to reset the MIPI interface.

The reset must be released only if MIPI receiver is in STOP state.
So, please set the MIPI transceiver to STOP state (LP11) before releasing MIPI reset.
Thus follow the procedure below to reset the MIPI interface:

- Write MIPInRST_CTL = 0000 0003_H
- Set MIPI transceiver to STOP state (LP11)
- Write MIPInRST_CTL = 0000 0000_H

NOTE

In order to reduce power consumption it is recommended to set the MIPI interface into reset if it is not used.

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + $0C_H$

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OTHER_RST	PHY_RST
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 37.54 MIPInRST_CTL register contents

Bit position	Bit name	Function
31 to 2	Reserved	These bits are always read as 0. When written, write the initial value.
1	OTHER_RST	Reset MIPI interface, except PHY layer and PHY layer interface 0: Release reset 1: Reset
0	PHY_RST	Reset PHY layer and PHY layer interface 0: Release reset 1: Reset

CAUTION

All parts of the MIPI Interface must be reset or released from reset at the same time.

Thus set

- **MIPInRST_CTL = 0000 0003_H to set the MIPI Interface in reset**
- **MIPInRST_CTL = 0000 0000_H to release the MIPI Interface from reset**

37.8.4.5 MIPInBUF_CTL – SubLVDS buffer control

This register is used to control the lane buffers.

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + 10_H

Initial value: 0000 $7FFF_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MIPInBUF_CTL[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MIPInBUF_CTL[15:0]															
Initial value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.55 MIPInBUF_CTL register contents

Bit position	Bit name	Function
31 to 0	MIPInBUF _CTL[31:0]	Lane buffer control • Set this register to 0000 0000_H if the MIPI interface is used. • Set this register to its initial value 0000 $7FFF_H$ if the MIPI interface is not used.

37.8.4.6 MIPInSOT_COUNT – $T_{CLK\text{-}SETTLE}$ and $T_{HS\text{-}SETTLE}$ register

This register determines the duration of $T_{CLK\text{-}SETTLE}$ and $T_{HS\text{-}SETTLE}$.

At the beginning of the HS transmission the LP receive buffers are switched off and the HS receive buffers and the line termination is switched on.

To mask possible analog effects that could occur during the switching a time is defined where the output from the HS receiver is ignored. Thus the time bridges the time until the switch from LP to HS has been settled.

For the data lane this time is $T_{HS\text{-}SETTLE}$ and is defined by MIPInSOT_COUNT.DATA_COUNT[7:0]. For the clock lane this time is $T_{CLK\text{-}SETTLE}$ and is defined by MIPInSOT_COUNT.CLK_COUNT[7:0].

(1) CLK_COUNT[7:0]

CLK_COUNT[7:0] must satisfy the following condition:

$$(95\ ns / T_{MIPIADCLK}) - 2 \leq CLK\_COUNT[7:0] \leq (300\ ns / T_{MIPIADCLK}) - 3$$

with

$T_{MIPIADCLK}$ $= 1/f_{MIPIADCLK}$ = cycle duration of the MIPIADCLK in ns

Example

for $f_{MIPIADCLK}$ = 160 MHz, i.e. $T_{MIPIADCLK}$ = 6.25 ns

$$(95\ ns / 6.25\ ns) - 2 \leq CLK\_COUNT[7:0] \leq (300\ ns / 6.25\ ns) - 3$$

$$13.2 \leq CLK\_COUNT[7:0] \leq 45$$

Thus CLK_COUNT[7:0] must be set between 14 and 45.

(2) DATA_COUNT[7:0]

DATA_COUNT[7:0] must satisfy the following condition

$$[(85\ ns + 6\ UI) / T_{HS\_SET\_CLK}] - 2 \leq DATA\_COUNT[7:0] \leq [(145\ ns + 10\ UI) / T_{HS\_SET\_CLK}] - 3$$

with

$T_{HS\_SET\_CLK}$ Count clock for T_{HS_SETTLE}, which is selected by MIPInMODE.SOT_CLKSEL:

- SOT_CLKSEL = 0: count clock is MIPIADCLK
  $T_{HS\_SET\_CLK} = T_{MIPIADCLK} = 1/f_{MIPI\_VI\_CLK}$
  = cycle duration of the MIPIADCLK in ns
- SOT_CLKSEL = 1: count clock is the MIPI interface input clock MIPI_VI_CLK/4
  $T_{HS\_SET\_CLK} = 4 \times T_{MIPI\_VI\_CLK} = 4/f_{MIPIADCLK}$
  = cycle duration of the MIPI_VI_CLK/4 in ns

UI Unit Interval, equal to the duration of any HS state on the clock lane, which is 1/2 clock cycle duration of the MIPI interface input clock MIPI_VI_CLK.

- $UI = T_{MIPI\_VI\_CLK}/2 = 2 \times f_{MIPI\_VI\_CLK}$
  = cycle duration of the 2 × MIPI_VI_CLK in ns

Example 1

for

- SOT_CLKSEL = 0
- $f_{MIPIADCLK}$ = 160 MHz → $T_{HS\_SET\_CLK}$ = 6.25 ns
- MIPI_VI_CLK = 240 MHz → UI = $T_{MIPI\_VI\_CLK}/2$ = 2.08 ns)

$$[(85\ ns + 6 \times 2.08\ ns) / 6.25\ ns] - 2 \leq DATA\_COUNT[7:0] \leq [(145\ ns + 10 \times 2.08\ ns) / 6.25] - 3$$

$$13.6 \leq DATA\_COUNT[7:0] \leq 23.5$$

Thus DATA_COUNT[7:0] must be set between 14 and 23.

Example 2

for

- SOT_CLKSEL = 1
- MIPI_VI_CLK = 240 MHz → $T_{HS\_SET\_CLK} = 4 \times T_{MIPI\_VI\_CLK}$ = 16.67 ns
- MIPI_VI_CLK = 240 MHz → UI = $T_{MIPI\_VI\_CLK}/2$ = 2.08 ns)

$$[(85\ ns + 6 \times 2.08\ ns) / 16.67\ ns] - 2 \leq DATA\_COUNT[7:0] \leq [(145\ ns + 10 \times 2.08\ ns) / 6.25] - 3$$

$$3.85 \leq DATA\_COUNT[7:0] \leq 6.95$$

Thus DATA_COUNT[7:0] must be set between 4 and 6.

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + 14_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	DATA_COUNT[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	CLK_COUNT[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.56 MIPInSOT_COUNT register contents

Bit position	Bit name	Function
31 to 24	Reserved	These bits are always read as 0. When written, write the initial value.
23 to 16	DATA_ COUNT[7:0]	$T_{HS-SETTLE}$ setting of data lane
15 to 8	Reserved	These bits are always read as 0. When written, write the initial value.
7 to 0	CLK_ COUNT[7:0]	$T_{CLK-SETTLE}$ setting of clock lane

37.8.4.7 MIPInRX_STATE – PHY layer status register

This register indicates the STOP state of the PHY.

PHY STOP state is indicated by

RX_STOP_STATE_0 = RX_STOP_STATE_1 = RX_STOP_STATE_C = 1.

Verification of the STOP state is used to check release timing for MIPInRST_CTL.
Refer to Section 37.8.4.4, MIPInRST_CTL – Reset register for details.

Access: This register can be read in 32-bit units.

Address: <MIPIn_base> + 18_H

Initial value: 0000 0492_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Bit17	RX_STOP_STATE_C
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	Bit11	Bit10	0	Bit8	Bit7	0	Bit5	RX_STOP_STATE_1	0	Bit2	RX_STOP_STATE_0	0
Initial value	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.57 MIPInRX_STATE register contents

Bit position	Bit name	Function
31 to 18	Reserved	These bits are always read as 0.
17	Bit17	The read value of this bit is undefined.
16	RX_STOP_STATE_C	This bit indicates the clock lane status of PHY layer. 0: Not STOP 1: STOP **Note:** This status can not be confirmed when MIPInBUF_CTL = 0000 $7FFF_H$ or MIPInRST_CTL = 0000 0003_H.
15 to 12	Reserved	These bits are always read as 0.
11, 10	Bit11, Bit10	The read values of these bits are undefined.
9	Reserved	These bits are always read as 0.
8, 7	Bit8, Bit7	The read values of these bits are undefined.
6	Reserved	This bit is always read as 0.
5	Bit5	The read value of this bit is undefined.
4	RX_STOP_STATE_1	This bit indicates the data lane 1 status of PHY layer. 0: Not STOP 1: STOP
3	Reserved	This bit is always read as 0.
2	Bit2	The read value of this bit is undefined.
1	RX_STOP_STATE_0	This bit indicates the data lane 0 status of PHY layer. 0: Not STOP 1: STOP
0	Reserved	This bit is always read as 0. When written, write the initial value.

37.8.4.8 MIPInWORD_COUNT – Long packet word count definition register

This register allows to define the maximum size of long packets to be transferred to the Video Data Controller.

If enabled (FORCE_WC_EN = 1) the maximum packet word count of Long Packet will be fixed to FORCE_WC_NUM[15:0].
If packet word count is larger than FORCE_WC_NUM[15:0], FORCE_WC_NUM[15:0] size will be sent to Video Data Controller VDCE0.
In such case a CRC error will be generated. Therefore CRC check should be disabled (MIPInMODE.CRC_EN = 0), when FORCE_WC_EN = 1.

NOTE

By shortening Long Packets, while the image is sent to the Video Data Controller, the right side of the image can be cut.
Alternatively the image can also be cut by the Video Data Controller.

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + $1C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FORCE_WC_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FORCE_WC_NUM[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.58 MIPInWORD_COUNT register contents

Bit position	Bit name	Function
31 to 17	Reserved	These bits are always read as 0. When written, write the initial value.
16	FORCE_ WC_EN	Enable Long Packet size limitation 0: Disable 1: Enable
15 to 0	FORCE_ WC_NUM[15:0]	Word count value of Long Packet, if enabled by FORCE_WC_EN = 1. One word has 8 bit.

37.8.4.9 MIPInLP_EN_ON_WC – LP buffer and noise removal timing register

This register adjusts the timing and the width of noise removal when LP buffer is turned on.

LP buffer is turned on again after high speed transmission has ended.
Each high speed packet contains a word count value that indicates the length of the packet.

Switching on of LP buffer can be adjusted absolute to the beginning of the packet (LP_EN_ON_MODE = 1) or relative to the end of the packet (LP_EN_ON_MODE = 0). The adjustment value is given by LP_EN_ON_WC[15:0].

Noise reduction is then defined in an interval centered around this switching point. The width of the interval is given by 2* LP_DEL_NOISE_TIM[6:0].

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	LP_DEL_NOISE_MODE	LP_DEL_NOISE_TIM[6:0]							LP_EN_ON_MODE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LP_EN_ON_WC[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.59 MIPInLP_EN_ON_WC register contents

Bit position	Bit name	Function
31 to 25	Reserved	These bits are always read as 0. When written, write the initial value.
24	LP_DEL_ NOISE_MODE	Enable noise removal mode. 1: Noise removal enabled 0: Noise removal disabled
23 to 17	LP_DEL_ NOISE_ TIM[6:0]	The count value for noise removal when LP_DEL_NOISE_MODE = 1. The recommended value of LP_DEL_NOISE_TIM[6:0] is 05_H.
16	LP_EN_ON_ MODE	Setting LP buffer on timing 0: LP buffer turns on when word count reached the calculated value (Word count in packet header) -LP_EN_ON_WC[15:0]. 1: LP buffer turns on after LP_EN_ON_WC[15:0] read words.
15 to 0	LP_EN_ON_ WC[15:0]	Set the word count (WC) when to (IO_MIPI_LP_EN_ * _OUT == 1) switch the LP BUFFER on. The recommended value of LP_EN_ON_WC[15:0] is 0020_H.

The noise reduction interval must not leave the data interval of the packet.
Thus LP_EN_ON_WC[15:0] and LP_DEL_NOISE_TIM[6:0] must comply to the following conditions:

- If noise removal enabled (LP_EN_ON_MODE = 1):

  - LP_EN_ON_WC[15:0] + LP_DEL_NOISE_TIM[6:0] + 16 < (wordcount in packet header)
  - LP_EN_ON_WC[15:0] – LP_DEL_NOISE_TIM[6:0] > 0
- If noise removal disabled (LP_EN_ON_MODE = 0):
  - LP_EN_ON_WC[15:0] – LP_DEL_NOISE_TIM[6:0] - 16 > 0
  - LP_EN_ON_WC[15:0] + LP_DEL_NOISE_TIM[6:0] < (wordcount in packet header)

NOTE

The word count in packet header is changed to MIPInWORD_COUNT.FORCE_WC_NUM[15:0], if Long Packet size limitation is enabled (MIPInWORD_COUNT.FORCE_WC_EN = 1).

37.8.4.10 MIPInLINE_BLANK – Line blanking period register

This register generates EOT (End of Transmission) timing.

MIPInLINE_BLANK controls the duration between the current line data (current Long Packet) and the next line data (next Long Packet).
This control is not related to MIPInWORD_COUNT control.

EOT timing is used in the following cases

- Initialization timing after a line data is transmitted.
- When MIPI mode moves from HS to LP under the condition that the number of words is less than WORD COUNT, EOT timing can be used to adjust the timing LP buffer turn on.

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + 28_H

Initial value: 0000 0004_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

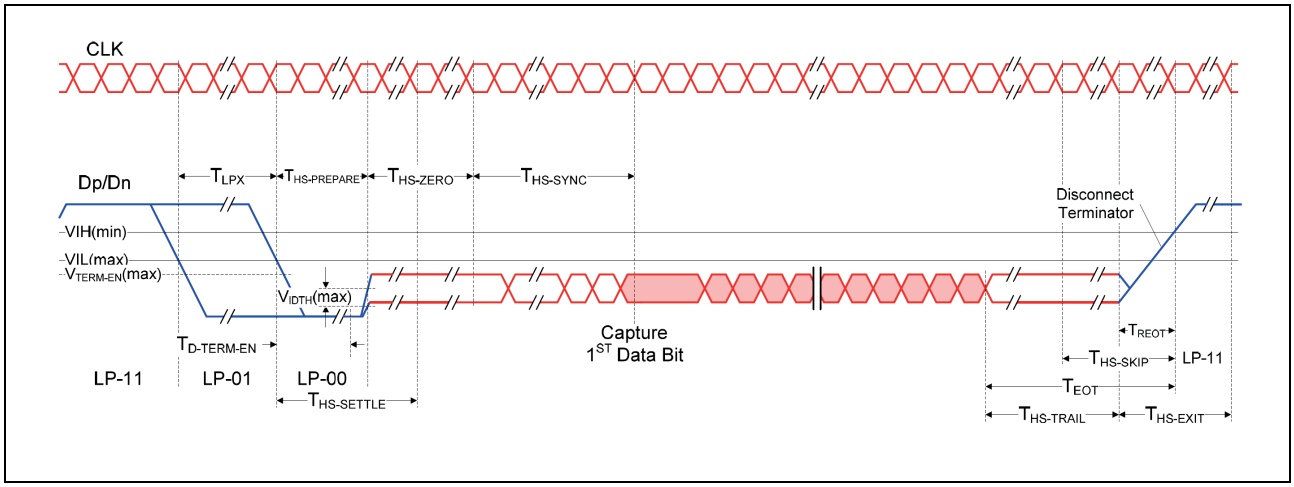
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	LINE_BLANK[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.60 MIPInLINE_BLANK register contents

Bit position	Bit name	Function
31 to 8	Reserved	These bits are always read as 0. When written, write the initial value.
7 to 0	LINE_ BLANK[7:0]	Interval until EOT in MIPIADCLK cycles

LINE_BLANK[7:0] must stay smaller than the minimum time until next HS transfer.
Thus it must satisfy the following condition:

- 12 * UI < MIPIADCLK cycles * (LINE_BLANK[7:0] + 2) < $T_{HS-EXIT}$ + T_{LPX} + $T_{HS-PREPARE}$ + $T_{HS-ZERO}$ − 12 * MIPIADCLK cycles
  UI means clock cycle / 2 in the HS mode (1 Data Bit Time).
- The setting of MIPInLINE_BLANK = 0 is prohibit.


CLK
Dp/Dn
VIH(min)
VIL(max)
VTERM-EN(max)
TLPX
THS-PREPARE
THS-ZERO
THS-SYNC
VIDTH(max)
TD-TERM-EN
THS-SETTLE
LP-11
LP-01
LP-00
Capture
1ST Data Bit
Disconnect
Terminator
TREOT
THS-SKIP
LP-11
TEOT
THS-TRAIL
THS-EXIT

37.8.4.11 MIPInRESET_DLY_CTL0 – LP11 reset delay adjustment register

This register selects the LP11 reset delay.

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + $2C_H$

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	RESET_DLY_LANE1[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	RESET_DLY_LANE0[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.61 MIPInRESET_DLY_CTL0 register contents

Bit position	Bit name	Function
31 to 28	Reserved	These bits are always read as 0. When written, write the initial value.
27 to 16	RESET_DLY_LANE1[11:0]	Control reset delay for data lane 1. **CAUTION** The initial value of RESET_DLY_LANE1[11:0] must be changed to $03F_H$ during the MIPI I/F initialization.
15 to 12	Reserved	These bits are always read as 0. When written, write the initial value.
11 to 0	RESET_DLY_LANE0[11:0]	Control reset delay for data lane 0. **CAUTION** The initial value of RESET_DLY_LANE0[11:0] must be changed to $03F_H$ during the MIPI I/F initialization.

37.8.4.12 MIPInINTSTATUS – Interrupt status register

This register shows the status of interrupt.

For each interrupt a dedicated status bit is provided with the following meaning:

- interrupt status bit = 0: interrupt has not occurred
- interrupt status bit = 1: interrupt has occurred

De-assertion of an interrupt and clearing of its status bit can be executed via the MIPInINTFFCLR register.

NOTE

If an interrupt is disabled (by writing 1 to its corresponding mask bit in the MIPInINTENCLR register), its status bit in MIPInSTATUS remains 0, even if the respective interrupt condition is fulfilled.

Access: This register can be read in 32-bit units.

Address: <MIPIn_base> + 40_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	ESC_TRIG_UK5	ESC_TRIG_UK4	ESC_TRIG_UK3	ESC_TRIG_RESET	0	0	0	0	0	0	PHWC_ERR	CTL_ERR_1
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ESC_ERR_1_3	SOT_SYNC_ERR_1	SOT_ERR_1	CTL_ERR_0	ESC_ERR_0	SOT_SYNC_ERR_0	SOT_ERR_0	BUF_OR_ERR	0	0	FRAME SYNC_ERR	0	CRC_ERR	ECC_SINGLE_ERR	ECC_MULTI_ERR	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 37.62 MIPInINTSTATUS register contents (1/2)

Bit position	Bit name	Function
31 to 28	Reserved	These bits are always read as 0. When written, write the initial value.
27	ESC_TRIG_UK5	Unknown-5 trigger
26	ESC_TRIG_UK4	Unknown-4 trigger
25	ESC_TRIG_UK3	Unknown-3 trigger
24	ESC_TRIG_RESET	Detect reset-trigger (remote application)
23 to 18	Reserved	These bits are always read as 0. When written, write the initial value.
17	PHWC_ERR	Detect packet header WORDCOUNT error
16	CTL_ERR_1	False Control Error for data lane 1
15	ESC_ERR_1_3	Escape mode entry command error for data lane 1
14	SOT_SYNC_ERR_1	Start-of-Transmission (SoT) synchronization error for data lane 1
13	SOT_ERR_1	Start-of-Transmission (SoT) error for data lane 1

Table 37.62 MIPInINTSTATUS register contents (2/2)

Bit position	Bit name	Function
12	CTL_ERR_0	False Control Error for data lane 0
11	ESC_ERR_0	Escape mode entry command error for data lane 0
10	SOT_SYNC_ ERR_0	Start-of-Transmission (SoT) synchronization error for data lane 0
9	SOT_ERR_0	Start-of-Transmission (SoT) error for data lane 0
8	BUF_OR_ERR	BUFFER overrun error
7 to 6	Reserved	These bits are always read as 0. When written, write the initial value.
5	FRAMESYNC_ ERR	FRAME SYNC error
4	Reserved	These bits are always read as 0. When written, write the initial value.
3	CRC_ERR	CRC error detection
2	ECC_SINGLE_ ERR	Single bit ECC error detection
1	ECC_MULTI_ ERR	Multi bit ECC error detection
0	Reserved	These bits are always read as 0. When written, write the initial value.

Interrupt details

- Unknown-5,4,3 trigger (bit[27:25])
  - If Unknown-5,4,3 trigger commands are received, these interrupts are asserted.
  - Entry Command Patterns are 10100000, 00100001 and 01011101.
- Reset-trigger (Remote Application, bit[24])
  - If Reset-trigger commands are received, this interrupt are asserted.
  - Entry Command Pattern is 01100010.
- Packet Header word count error (bit[17])
  - Asserted when word count is lager than
    Actual Data Size + ((Ths-trail + Ths-exit) / (UI * 8)) * Number of Lane
    and receive next packet with the LP buffer tuerned on status. (LP buffer turn on timing is controllable by MIPInLP_EN_ON_WC register).
- False Control Error (bit[16], [12])
  - If a LP-Rqst (LP-10) is not followed by the remainder of a valid Escape or Turnaround sequence, a False Control Error is indicated. This error is also indicated if a HS-Rqst (LP-01) is not correctly followed by a Bridge State (LP-00).
- Escape Mode Entry Command Error (bit[15], [11])
  - If the receiving Lane Module does not recognize the received Entry Command for Escape mode an Escape mode Entry Command Error is indicated.
- SoT Sync Error (bit[14], [10])

  - If the SoT Leader sequence is corrupted in a way that proper synchronization cannot be expected, a SoT Sync Error is indicated.

  When SOT_SYNC_ERR is detected the MIPI module must be reset by using the MIPInRST_CTL register. See Section 37.8.4.4, MIPInRST_CTL – Reset register for details.

- SoT Error (bit[13], [9])
  - The Leader sequence for Start of High-Speed Transmission is fault tolerant for any single-bit error and some multi bit errors. Therefore, the synchronization may be usable, but confidence in the payload data is lower. If this situation occurs an SoT Error is indicated.
- BUFFER overrun error(bit[8])
  - Asserted when the overrun is occurred in the internal data FIFO of MIPI module.
- FRAMESYNC_ERR (bit[5])
  - Asserted when a Frame End (FE) is not paired with a Frame Start (FS) on the same virtual channel.
- CRC_ERR (bit[3])
  - Asserted when the computed CRC code is different than the received CRC code.
- ECC_SINGLE_ERR (bit[2])
  - Asserted when an ECC syndrome was computed and a single bit-error in the packet header was detected and corrected.
- ECC_MULTI_ERR (bit[1])
  - Asserted when an ECC syndrome was computed and two bit-errors are detected in the received packet header.

37.8.4.13 MIPInINTENSET – Interrupt enable set register

This register is used to enable the interrupts.

For each interrupt a dedicated interrupt enable bit is provided with the following function:

- at write:
  - interrupt enable bit = 0: no function
  - interrupt enable bit = 1: enable interrupt
- at read:
  - interrupt enable bit = 0: interrupt disabled
  - interrupt enable bit = 1: interrupt enabled

Disabling an interrupt can be executed via the MIPInINTENCLR register.

Enabling the interrupts

- Unknown-5,4,3 trigger (ESC_TRIG_UK5_EN, ESC_TRIG_UK4_EN, ESC_TRIG_UK3_EN = 1)
- Reset-trigger (ESC_TRIG_RESET_EN = 1)
- False Control Error (CTL_ERR_1_EN = 1, CTL_ERR_0_EN = 1)
- Escape Mode Entry Command Error (ESC_ERR_1_3_EN = 1, ESC_ERR_0_EN = 1)

should be used with the noise removal of clock line enabled (MIPInMODE.DEL_LP11_NOISE = 1).

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + 48_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	ESC_TRIG_UK5_EN	ESC_TRIG_UK4_EN	ESC_TRIG_UK3_EN	ESC_TRIG_RESET_EN	0	0	0	0	0	0	PHWC_ERR_EN	CTL_ERR_1_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ESC_ERR_13_EN	SOT_SYNC_ERR_1_EN	SOT_ERR_1_EN	CTL_ERR_0_EN	ESC_ERR_0_EN	SOT_SYNC_ERR_0_EN	SOT_ERR_0_EN	BUF_OR_ERR_EN	0	0	FRAMESYNC_ERR_EN	0	CRC_ERR_EN	ECC_SINGLE_ERR_EN	ECC_MULTI_ERR_EN	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R	R/W	R/W	R/W	R

Table 37.63 MIPInINTENSET register contents (1/2)

Bit position	Bit name	Function
31 to 28	Reserved	These bits are always read as 0. When written, write the initial value.
27	ESC_TRIG_UK5_EN	Unknown-5 trigger enable
26	ESC_TRIG_UK4_EN	Unknown-4 trigger enable
25	ESC_TRIG_UK3_EN	Unknown-3 trigger enable

Table 37.63 MIPInINTENSET register contents (2/2)

Bit position	Bit name	Function
24	ESC_TRIG_RESET_EN	Detect reset-trigger (remote application) enable This is only a regular interrupt triggered by a MIPI escape sequence. It is similar to the unknown trigger events and does not trigger any kind of reset.
23 to 18	Reserved	These bits are always read as 0. When written, write the initial value.
17	PHWC_ERR_EN	Detect packet header WORDCOUNT error enable
16	CTL_ERR_1_EN	False Control Error for data lane 1 enable
15	ESC_ERR_1_3_EN	Escape mode entry command error for data lane 1 enable
14	SOT_SYNC_ERR_1_EN	Start-of-Transmission (SoT) synchronization error for data lane 1 enable
13	SOT_ERR_1_EN	Start-of-Transmission (SoT) error for data lane 1 enable
12	CTL_ERR_0_EN	False Control Error for data lane 0 enable
11	ESC_ERR_0_EN	Escape mode entry command error for data lane 0 enable
10	SOT_SYNC_ERR_0_EN	Start-of-Transmission (SoT) synchronization error for data lane 0 enable
9	SOT_ERR_0_EN	Start-of-Transmission (SoT) error for data lane 0 enable
8	BUF_OR_ERR_EN	BUFFER overrun error enable
7 to 6	Reserved	These bits are always read as 0. When written, write the initial value.
5	FRAMESYNC_ERR_EN	FRAME SYNC error enable
4	Reserved	These bits are always read as 0. When written, write the initial value.
3	CRC_ERR_EN	CRC error detection enable
2	ECC_SINGLE_ERR_EN	Single bit ECC error detection enable
1	ECC_MULTI_ERR_EN	Multi bit ECC error detection enable
0	Reserved	These bits are always read as 0. When written, write the initial value.

37.8.4.14 MIPInINTENCLR – Interrupt enable clear register

This register is used to mask the interrupts.

For each interrupt a dedicated interrupt mask bit is provided with the following function:

- at write:
  - interrupt mask bit = 0: no function
  - interrupt mask bit = 1: disable (mask) interrupt
- at read: always 0

Enabling an interrupt can be executed via the MIPInINTENSET register.

Access: This register can be written in 32-bit units.

Address: <MIPIn_base> + $4C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	ESC_TRIG_UK5_MASK	ESC_TRIG_UK4_MASK	ESC_TRIG_UK3_MASK	ESC_TRIG_RESET_MASK	0	0	0	0	0	0	PHWC_ERR_MASK	CTL_ERR_1_MASK
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W	R	R	R	R	R	R	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ESC_ERR_1_3_MASK	SOT_SYNC_ERR_1_MASK	SOT_ERR_1_MASK	CTL_ERR_0_MASK	ESC_ERR_0_MASK	SOT_SYNC_ERR_0_MASK	SOT_ERR_0_MASK	BUF_OR_ERR_MASK	0	0	FRAMESYNC_ERR_MASK	0	CRC_ERR_MASK	ECC_SINGLE_ERR_MASK	ECC_MULTI_ERR_MASK	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	R	R	W	R	W	W	W	R

Table 37.64 MIPInINTENCLR register contents (1/2)

Bit position	Bit name	Function
31 to 28	Reserved	These bits are always read as 0. When written, write the initial value.
27	ESC_TRIG_UK5_MASK	Unknown-5 trigger mask
26	ESC_TRIG_UK4_MASK	Unknown-4 trigger mask
25	ESC_TRIG_UK3_MASK	Unknown-3 trigger mask
24	ESC_TRIG_RESET_MASK	Detect reset-trigger (remote application) mask
23 to 18	Reserved	These bits are always read as 0. When written, write the initial value.
17	PHWC_ERR_MASK	Detect packet header WORDCOUNT error mask
16	CTL_ERR_1_MASK	False Control Error for data lane 1 mask
15	ESC_ERR_1_3_MASK	Escape mode entry command error for data lane 1 mask
14	SOT_SYNC_ERR_1_MASK	Start-of-Transmission (SoT) synchronization error for data lane 1 mask
13	SOT_ERR_1_MASK	Start-of-Transmission (SoT) error for data lane 1 mask

Table 37.64 MIPInINTENCLR register contents (2/2)

Bit position	Bit name	Function
12	CTL_ERR_0_ MASK	False Control Error for data lane 0 mask
11	ESC_ERR_0_ MASK	Escape mode entry command error for data lane 0 mask
10	SOT_SYNC_ ERR_0_MASK	Start-of-Transmission (SoT) synchronization error for data lane 0 mask
9	SOT_ERR_0_ MASK	Start-of-Transmission (SoT) error for data lane 0 mask
8	BUF_OR_ERR _MASK	BUFFER overrun error mask
7 to 6	Reserved	These bits are always read as 0. When written, write the initial value.
5	FRAMESYNC_ ERR_MASK	FRAME SYNC error mask
4	Reserved	These bits are always read as 0. When written, write the initial value.
3	CRC_ERR_ MASK	CRC error detection mask
2	ECC_SINGLE_ ERR_MASK	Single bit ECC error detection mask
1	ECC_MULTI_ ERR_MASK	Multi bit ECC error detection mask
0	Reserved	These bits are always read as 0. When written, write the initial value.

37.8.4.15 MIPInINTFFCLR – Interrupt factor clear register

This register is used to de-assert an interrupt and clear its status flag in the MIPInINTSTATUS register..

For each interrupt a dedicated interrupt bit is provided with the following function:

- at write:
  - interrupt factor clear bit = 0: no function
  - interrupt factor clear bit = 1: de-assert interrupt and clear its status bit in MIPInINTSTATUS
- at read: always 0

Access: This register can be written in 32-bit units.

Address: <MIPIn_base> + 50_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	ESC_TRIG_UK5_CLR	ESC_TRIG_UK4_CLR	ESC_TRIG_UK3_CLR	ESC_TRIG_RESET_CLR	0	0	0	0	0	0	PHWC_ERR_CLR	CTL_ERR_1_CLR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	W	W	W	W	R	R	R	R	R	R	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ESC_ERR_1_3_CLR	SOT_SYNC_ERR_1_CLR	SOT_ERR_1_CLR	CTL_ERR_0_CLR	ESC_ERR_0_CLR	SOT_SYNC_ERR_0_CLR	SOT_ERR_0_CLR	BUF_OR_ERR_CLR	0	0	FRAME SYNC_ERR_CLR	0	CRC_ERR_CLR	ECC_SINGLE_ERR_CLR	ECC_MULTI_ERR_CLR	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	R	R	W	R	W	W	W	R

Table 37.65 MIPInINTFFCLR register contents (1/2)

Bit position	Bit name	Function
31 to 28	Reserved	These bits are always read as 0. When written, write the initial value.
27	ESC_TRIG_UK5_CLR	Unknown-5 trigger clear
26	ESC_TRIG_UK4_CLR	Unknown-4 trigger clear
25	ESC_TRIG_UK3_CLR	Unknown-3 trigger clear
24	ESC_TRIG_RESET_CLR	Detect reset-trigger (remote application) clear
23 to 18	Reserved	These bits are always read as 0. When written, write the initial value.
17	PHWC_ERR_CLR	Detect packet header WORDCOUNT error clear
16	CTL_ERR_1_CLR	False Control Error for data lane 1 enable clear
15	ESC_ERR_1_3_CLR	Escape mode entry command error for data lane 1 enable clear
14	SOT_SYNC_ERR_1_CLR	Start-of-Transmission (SoT) synchronization error for data lane 1 enable clear
13	SOT_ERR_1_CLR	Start-of-Transmission (SoT) error for data lane 1 enable clear
12	CTL_ERR_0_CLR	False Control Error for data lane 0 enable clear

Table 37.65 MIPInINTFFCLR register contents (2/2)

Bit position	Bit name	Function
11	ESC_ERR_0_CLR	Escape mode entry command error for data lane 0 enable clear
10	SOT_SYNC_ERR_0_CLR	Start-of-Transmission (SoT) synchronization error for data lane 0 enable clear
9	SOT_ERR_0_CLR	Start-of-Transmission (SoT) error for data lane 0 enable clear
8	BUF_OR_ERR_CLR	BUFFER overrun error enable clear
7 to 6	Reserved	These bits are always read as 0. When written, write the initial value.
5	FRAMESYNC_ERR_CLR	FRAME SYNC error clear
4	Reserved	These bits are always read as 0. When written, write the initial value.
3	CRC_ERR_CLR	CRC error detection clear
2	ECC_SINGLE_ERR_CLR	Single bit ECC error detection clear
1	ECC_MULTI_ERR_CLR	Multi bit ECC error detection clear
0	Reserved	These bits are always read as 0. When written, write the initial value.

37.8.4.16 MIPInEOT_COUNT – T_{EOT} setting

This register sets the reset timing of DES according T_{EOT}.

EOT_COUNT[7:0] refers to the number of MIPIADCLK cycles. It is the period from changing LP11 to beginning reset.

In the case that the data rate is more than 266 Mbps per one lane, this register should be set 0.

Otherwise, it is necessary to satisfy the following condition.

- MIPInMODE.SOT_CLKSEL = 0 (MIPIADCLK is $T_{HS\text{-}SETTLE}$ count clock):
  16 UI < $T_{MIPIADCLK}$ x EOT_COUNT[7:0]
  ($T_{MIPIADCLK}$ = MIPIADCLK cycle duration)
- MIPInMODE.SOT_CLKSEL = 1(external clock divided by 4 is $T_{HS\text{-}SETTLE}$ count clock):
  EOT_COUNT[7:0] = 3

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + 74_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	EOT_COUNT[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.66 MIPInEOT_COUNT register contents

Bit position	Bit name	Function
31 to 8	Reserved	These bits are always read as 0. When written, write the initial value.
7 to 0	EOT_ COUNT[7:0]	Counter value indicating reset timing for EoT.

37.8.4.17 MIPInVIN_MODE – Video input color format register

This register selects the color format of the video data.

Access: This register can be read/written in 32-bit units.

Address: <MIPIn_base> + $9C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	VIN_MODE[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 37.67 MIPInVIN_MODE register contents

Bit position	Bit name	Function
31 to 2	Reserved	These bits are always read as 0. When written, write the initial value.
1 to 0	VIN_ MODE[1:0]	Selection of the video data color format 00_B: YUV422 01_B: RGB888 10_B: RGB565 11_B: Setting prohibited

37.8.5 MIPI Operation

37.8.5.1 Start and stop sequence

MIPI port setup

(1) Select alternative port function: i.e. PMC40.PMC40_[5:0] = 11 1111_B for two data lanes

(2) Set ports as input: i.e. PM40.PM40_[5:0] = 11 1111_B for two data lanes

(3) Wait 20 µsec buffer stabilization time

(4) MIPI video source and clock setup

- Select video input channel 0 for MIPI CSI-2: VDCECTL.PXSL = 1
- Setup clocks
  – C_ISO_MIPIADCLK
  – C_ISO_MIPIPIXCLK

Start sequence

1. Reset by MIPInRST_CTL = 0000 0003_H
2. Set up following registers:
   - MIPInMODE
   - MIPInDATA_DLY_CTL
   - MIPInBUF_CTL
   - MIPInSOT_COUNT
   - MIPInWORD_COUNT
   - MIPInLP_EN_ON_WC
   - MIPInLINE_BLANK
   - MIPInRESET_DLY_CTL0
   - MIPInEOT_COUNT
   - MIPInVIN_MODE
3. Release reset by MIPInRST_CTL = 0000 0000_H
4. Enable MIPI interrupts by set up to the following registers:
   – MIPInINTENCLR
   – MIPInINTFFCLR
   – MIPInINTENSET
   Activate the following interrupts by the Interrupt Controller:
   – INTMIPI0OVF
   – INTMIPI0CTL
5. Enable MIPI transmission by MIPInON.MIPI_ON = 1.

Example start sequence

In the following a register set-up example is given for a MIPI CSI-2 video data stream with following parameters:

- 640 x 480 pixel size (VGA with VESA timing)
- 24 bpp (RGB888)

- using 2 data lanes of MIPI CSI-2
- using clock C_ISO_MIPIPIXCLK = 26.67 MHz
  (from C_ISO_MIPIADCLK = MIPIPLLCLK = PLL0 / 3 = 160 MHz;
  C_ISO_MIPIPIXCLK = MIPIPLLCLK / 6 = 26.67 MHz)

Register settings:

1. MIPInMODE = 2982 0003_H
2. MIPInDATA_DLY_CTL = 0000 0000_H
3. MIPInBUF_CTL = 0000 0000_H
4. MIPInSOT_COUNT = 0014 0014_H
5. MIPInWORD_COUNT = 0000 0000_H
6. MIPInLP_EN_ON_WC = 010A 0020_H
7. MIPInLINE_BLANK = 0000 0004_H
8. MIPInRESET_DLY_CTL0 = 003F $003F_H$
9. MIPInEOT_COUNT = 0000 0000_H
10. MIPInVIN_MODE = 0000 0001_H

CAUTION

Do not change the registers MIPInMODE, MIPInDATA_DLY_CTL, MIPInBUF_CTL, MIPInSOT_COUNT, MIPInWORD_COUNT, MIPInLP_EN_ON_WC, MIPInLINE_BLANK, MIPInRESET_DLY_CTL0, MIPInEOT_COUNT while the MIPI interface is active.

These registers may only be changed after the stop sequence.

Stop sequence

1. Disable MIPI Interrupts by the Interrupt Controller:
   – INTMIPI0OVF
   – INTMIPI0CTL
2. Disable the MIPI interface by MIPInON.MIPI_ON = 0.

37.8.5.2 Interrupt handling sequence

- INTMIPI0OVF interrupt handling:
  When MIPI buffer overflow is detected the MIPI module must be reset by using the MIPInRST_CTL register. See Section 37.8.4.4, MIPInRST_CTL – Reset register for details.
- INTMIPI0CTL interrupt handling:
  Evaluate interrupt cause by MIPInINTSTATUS interrupt status register and handle cause as described in Section 37.8.4.12, MIPInINTSTATUS – Interrupt status register.

37.8.5.3 Video Input Setup for MIPI

The MIPI Video Input Interface provides the video data to the Video Input channel 0 with the following attributes:

1. sync and data capture point
   - horizontal- and vertical- sync is to be captured on rising clock edge
   - data is to be captured on riding clock edge
2. horizontal blanking:
   - sync width fixed to 1 pixel clock
   - back porch fixed to 16 pixel clocks
   - active high horizontal sync
3. vertical blanking:
   - sync width fixed to 1 line
   - back porch fixed to 3 lines
   - active high vertical sync
   - first data line is the line with active sync

That means the following register settings have to be performed for the VDCE Video Input channel 0:

- DEMODE0.DE_4HS_EN = 1 when MIPInMODE.VTIM is 0.
- INP_SEL_CNT.INP_PXD_EDGE = 0 (rising edge)
- INP_SEL_CNT.INP_VS_EDGE = 0 (rising edge)
- INP_SEL_CNT.INP_HS_EDGE = 0 (rising edge)
- SC0_SCL0_DS2.SC0_RES_VS = 4 (refer to note below)
  (1 line vsync + 3 line backporch = 4 lines)
- SC0_SCL0_DS2.SC0_RES_VW = 484 (refer to note below)
  (height of video signal + 4 = 484 lines at VGA)
- SC0_SCL0_DS3.SC0_RES_HS = 17 (1 pixel hsync + 16 pixel backporch = 17 pixel)
- SC0_SCL0_DS3.SC0_RES_HW = 640 (width of video signal = 640 pixels at VGA)

NOTE

The frame sync (vertical sync and horizontal sync) information is contained in the actual data packets (long packets) of the MIPI CSI-2 data stream.

By this fact the vertical sync is provided to the VDCE video input channel 0 together with the first line of video data. As the VDCE video input requires 4 lines from the reception of VSYNC to the capture of the first line (min. value of SC0_SCL0_DS2.SC0_RES_VS register is 4) the first 4 lines can't be captured.

However, this issue can be avoided by setting DEMODE0.DE_4HS_EN to 1 in VDCE0 and MIPInMODE.VTIM to 0. By this setting, VDC automatically shift 4 lines after VSYNC detection between VSYNC and the first HSYNC from MIPI.

Also, when DEMODE0.DE_4HS_EN is set to 1, the detection of the end of line is available by setting DEMODE0.DE_VLAST_EN. For the detail, please refer DEMODE0 and DEMODE1 in VDCE chapter.

When MIPInMODE.VTIM is 1, the MIPI CSI-2 video data stream needs to contain 4 lines of dummy data as first active data lines.

I.e. when a 640 x 480 pixels size video input source (VGA) shall be captured by the VDCE video input via MIPI CSI-2, provide a 640 x 484 pixels size video source with lines 1 to 4 containing dummy data and lines 5 to 484 containing the actual video data.

Application hint: This can be done by providing a data enable signal to the MIPI CSI-2 generating video converter, that starts 4 lines earlier than the actual video data.

37.8.5.4 Conditions for MIPI video clock selection

The MIPI Video Input Interface can select the pixel clock to the Video Input channel 0 (C_ISO_MIPIPIXCLK) generated from the clock source C_ISO_MIPIADCLK by division.

The selection of this divider (CKSC_IMIPIPIXD_CTL) needs to fulfill the following two conditions.

(1) Condition 1: VIN pixel clock (restriction of MIPI Video data buffers)

- C_ISO_MIPIPIXCLK > VIN pixel clock
- (C_ISO_MIPIADCLK / DIV) > VIN pixel clock

with DIV is 3, 6 or 12. (Refer to Section 37.5.7.29, CKSC_IMIPIPIXD_CTL — C_ISO_MIPIPIXCLK clock divider register (D1M2(H) only))

Example:

- C_ISO_MIPIADCLK = 160 MHz
- QVGA 240 x 320 (native video data timing of data embedded into MIPI CSI-2 video stream)

⇒ VIN pixel clock = 5 MHz
(160 MHz / 12) = 13.34 MHz > 5 MHz

⇒ Select DIV = 12

(2) Condition 2: max. HSYNC width (restriction of VDCE)

- (C_ISO_MIPIPIXCLK × VIN total width) / VIN pixel clock ≤ 2015 pixels
- ((C_ISO_MIPIADCLK / DIV) × VIN total width) / VIN pixel clock ≤ 2015 pixels

with DIV is 3, 6 or 12. (Refer to Section 37.5.7.29, CKSC_IMIPIPIXD_CTL — C_ISO_MIPIPIXCLK clock divider register (D1M2(H) only))

Example:

- C_ISO_MIPIADCLK = 160 MHz
- QVGA 240 x 320 (native video data timing of data embedded into MIPI CSI-2 video stream)

⇒ VIN total width = 320 + 84 = 404 pixel

⇒ VIN pixel clock = 5 MHz
((160MHz / 12) * 404 pixel) / 5MHz = 1078 <= 2015 pixels

⇒ Select DIV = 12

37.8.5.5 VSYNC Timing from MIPI to VDCE

The following figure show the difference of VSYNC output timing by MIPInMODE.VTIM.

When MIPInMODE.VTIM is 0, VSYNC is asserted before almost 1 line from the first line output. Also, VSYNC is asserted until next data come to data buffer.

When MIPInMODE.VTIM is 1, VSYNC is output at the same timing as HSYNC to VDCE.

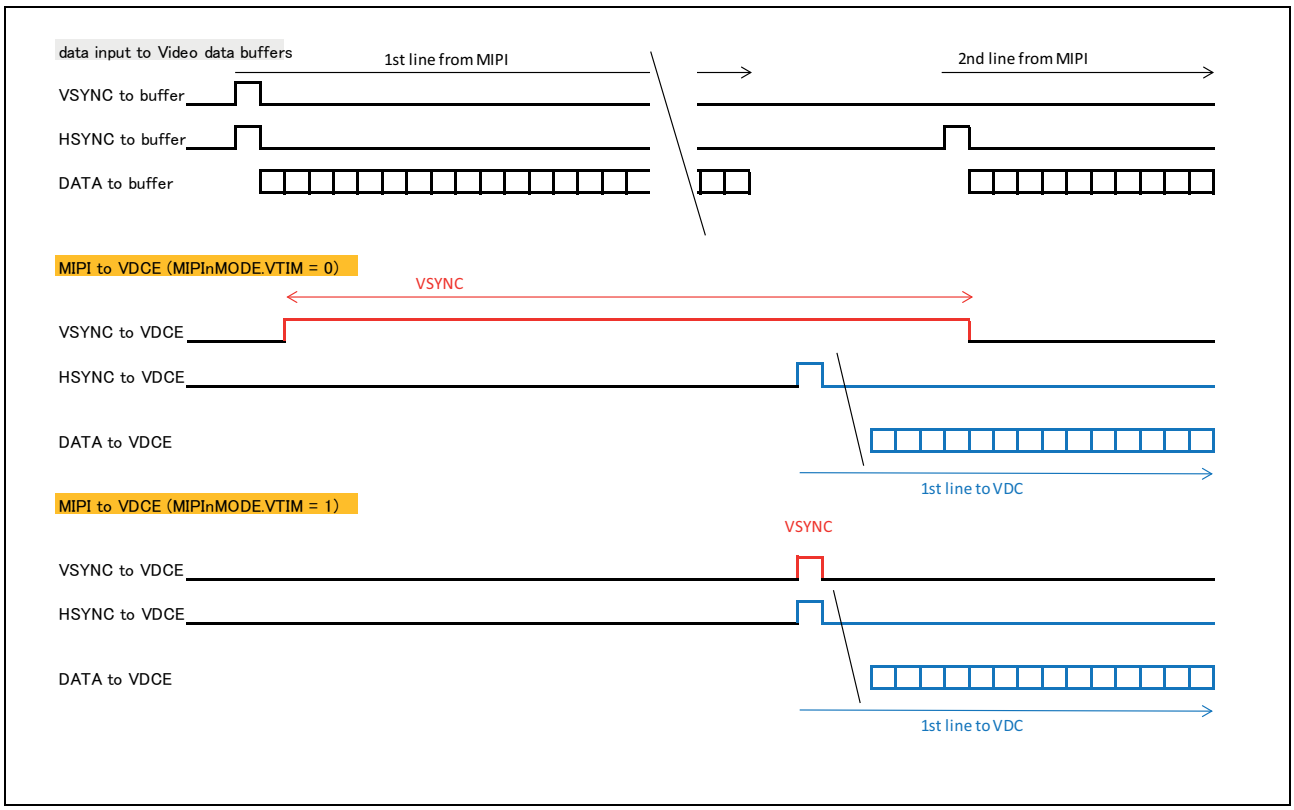


Figure 37.19 VSYNC Timing from MIPI to VDCE

NOTE

- MIPInMODE.VTIM is 1, DEMODE0 and 1 in VDCE0 register can be available.
- VDCE setting except DEMODE0 and 1 for MIPI is the same regardless MIPInMODE.VTIM setting.

MIPI setting

1. Fix the capturing position
   - SC_SCL0_DS2.SC_RES_VS = 4
   - SC_SCL0_DS3.SC_RES_HS = 17
2. DEMODE setting

   [No line detection]
   - DEMODE0=0x00000010

   [line detection after the last line finished]

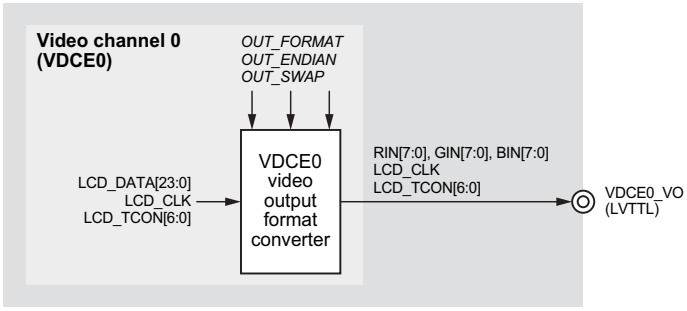
- DEMODE0=0x00000014
- DEMODE1= active line number
  e.g) when active vertical width is 480, DEMODE1 is 480.
- SC_SCL0_INT.SC_RES_LINE = active line number + 4
  e.g) when active vertical width is 480, SC_RES_LINE is 484.

3. Other VDCE setting is the same as video input setting with HSYNC

37.9 Video Output selection and RSDS control

37.9.1 D1L2(H) and D1M1(H) Video output selection

The diagram below illustrates the video output signal connections.



Note: The Video Data Controllers VDCEn include facilities to change the mapping of the color output data LCD_DATA[23:0] to the VDCEn output data RIN[7:0], GIN[7:0] and BIN[7:0]. Refer to Section 38, Video Data Controller E (VDCE) for details about the OUT_FORMAT, OUT_ENDIAN and OUT_SWAP settings.

Figure 37.20 D1L2(H) and D1M1(H) video output signals

The following table shows the assignment between the various video output signals for following settings:

- VDCEn video output format converter settings:
  - OUT_SET.OUT_FORMAT[1:0] = 00_B
  - OUT_SET.OUT_ENDIAN_ON = 0
  - OUT_SET.OUT_SWAP_ON = 0

  For other bit allocations of LCD_DATA[23:0] to RIN[7:0], GIN[7:0] and BIN[7:0] refer to Section 38, Video Data Controller E (VDCE).

Table 37.68 D1L2(H) and D1M1(H) Video output format selection example (1/2)

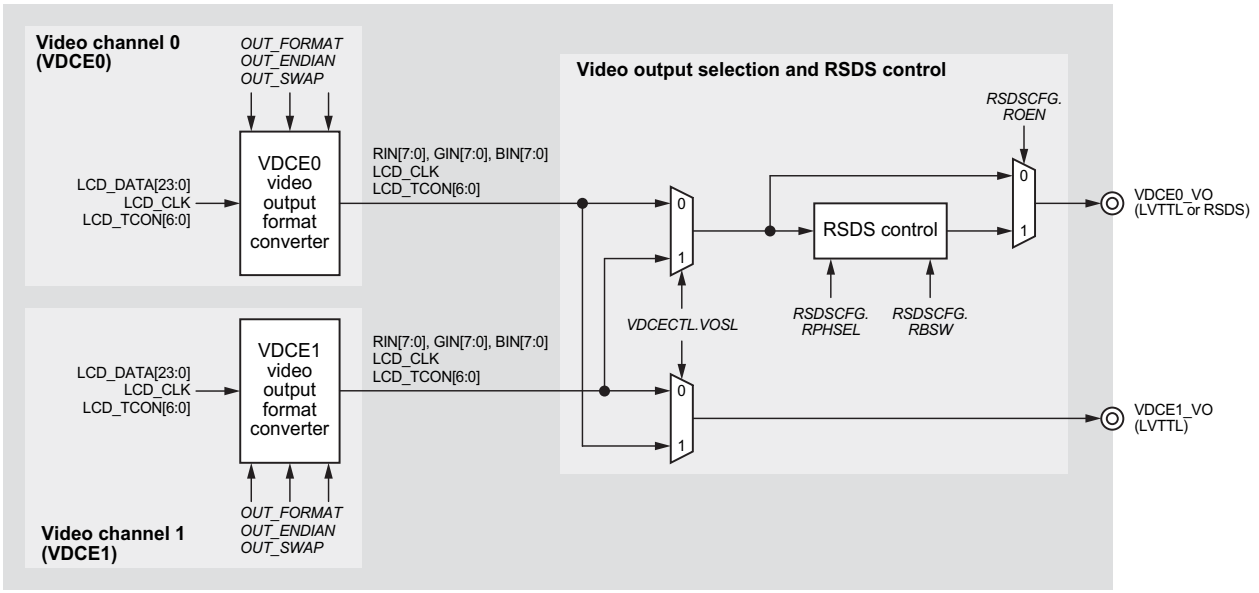
VDCE0 video output data	VDCE0 video output format converter output	Port signal name	LVTTL signal
Color data signals			
LCD_DATA23	RIN[7]	VDCE0_VO_DATA23	R[7]
LCD_DATA22	RIN[6]	VDCE0_VO_DATA22	R[6]
LCD_DATA21	RIN[5]	VDCE0_VO_DATA21	R[5]
LCD_DATA20	RIN[4]	VDCE0_VO_DATA20	R[4]
LCD_DATA19	RIN[3]	VDCE0_VO_DATA19	R[3]
LCD_DATA18	RIN[2]	VDCE0_VO_DATA18	R[2]
LCD_DATA17	RIN[1]	VDCE0_VO_DATA17	R[1]
LCD_DATA16	RIN[0]	VDCE0_VO_DATA16	R[0]
LCD_DATA15	GIN[7]	VDCE0_VO_DATA15	G[7]
LCD_DATA14	GIN[6]	VDCE0_VO_DATA14	G[6]
LCD_DATA13	GIN[5]	VDCE0_VO_DATA13	G[5]
LCD_DATA12	GIN[4]	VDCE0_VO_DATA12	G[4]
LCD_DATA11	GIN[3]	VDCE0_VO_DATA11	G[3]

Table 37.68 D1L2(H) and D1M1(H) Video output format selection example (2/2)

VDCE0 video output data	VDCE0 video output format converter output	Port signal name	LVTTL signal
LCD_DATA10	GIN[2]	VDCE0_VO_DATA10	G[2]
LCD_DATA9	GIN[1]	VDCE0_VO_DATA9	G[1]
LCD_DATA8	GIN[0]	VDCE0_VO_DATA8	G[0]
LCD_DATA7	BIN[7]	VDCE0_VO_DATA7	B[7]
LCD_DATA6	BIN[6]	VDCE0_VO_DATA6	B[6]
LCD_DATA5	BIN[5]	VDCE0_VO_DATA5	B[5]
LCD_DATA4	BIN[4]	VDCE0_VO_DATA4	B[4]
LCD_DATA3	BIN[3]	VDCE0_VO_DATA3	B[3]
LCD_DATA2	BIN[2]	VDCE0_VO_DATA2	B[2]
LCD_DATA1	BIN[1]	VDCE0_VO_DATA1	B[1]
LCD_DATA0	BIN[0]	VDCE0_VO_DATA0	B[0]
Pixel clock			
LCD_CLK	LCD_CLK	VDCE0_VO_CLKP	VDCE0_VO_CLKP
Timing Controller signals			
LCD_TCON0	LCD_TCON0	VDCE0_VO_TCON0	
LCD_TCON1	LCD_TCON1	VDCE0_VO_TCON1	
LCD_TCON2	LCD_TCON2	VDCE0_VO_TCON2	
LCD_TCON3	LCD_TCON3	VDCE0_VO_TCON3	
LCD_TCON4	LCD_TCON4	VDCE0_VO_TCON4	
LCD_TCON5	LCD_TCON5	VDCE0_VO_TCON5	
LCD_TCON6	LCD_TCON6	VDCE0_VO_TCON6	

37.9.2 D1M2(H) Video output selection

The diagram below illustrates the video output selection options.



Note: The Video Data Controllers VDCEn include facilities to change the mapping of the color output data LCD_DATA[23:0] to the VDCEn output data RIN[7:0], GIN[7:0] and BIN[7:0]. Refer to Section 38, Video Data Controller E (VDCE) for details about the OUT_FORMAT, OUT_ENDIAN and OUT_SWAP settings.

Figure 37.21 D1M2(H) video output selections

By use of the VDCECTL.VOSL selector the output signals from the video channels 0 and 1 can be swapped, so that any of both can be used for the RSDS output. The other video output is output as LVTTL signals.

Table 37.69 D1M2(H) Video output selections

VDCECTL.VOSL	RSDSCFG.ROEN	Ports VDCE0_VO	Ports VDCE1_VO
0	0	Video channel 0: LVTTL	Video channel 1: LVTTL
	1	Video channel 0: RSDS	
1	0	Video channel 1: LVTTL	Video channel 0: LVTTL
	1	Video channel 1: RSDS	

37.9.2.1 Video data and clock output settings

The following tables show the assignment between the video data and clock output signals for different settings endian and R – B swap (red channel – blue channel swapping).

Basic video output selection and RSDS:

- RSDSCFG.ROEN = 1: VDCE0_VO as RSDS
- VDCECTL.VOSL determines which video output channel is output as RSDS via VDCE0_VO (no impact on the tables)
- The tables show the assignment for RGB888 output format, selected by OUT_SET.OUT_FORMAT[1:0] = 00_B. For other formats refer to Section 38, Video Data Controller E (VDCE).

(a) VDCEn video output format converter settings (no R – B swap, little endian)

– OUT_SET.OUT_FORMAT[1:0] = 00_B (RGB888 output)

– OUT_SET.OUT_ENDIAN_ON = 0 (little endian)

– OUT_SET.OUT_SWAP_ON = 0 (no R – B swap)

Table 37.70 D1M2(H) Video output format selection (no R – B swap, little endian)

VDCEn video output data	VDCEn video output format converter output	Port signal name	LVTTL	RSDS
Color data signals (LCD_DATA[23:0] from video data channel selected by VDCECTL.VOSL)				
LCD_DATA23	RIN[7]	VDCE0_VO_DATA23	R[7]	RSDS_R[3]_P
LCD_DATA22	RIN[6]	VDCE0_VO_DATA22	R[6]	RSDS_R[3]_N
LCD_DATA21	RIN[5]	VDCE0_VO_DATA21	R[5]	RSDS_R[2]_P
LCD_DATA20	RIN[4]	VDCE0_VO_DATA20	R[4]	RSDS_R[2]_N
LCD_DATA19	RIN[3]	VDCE0_VO_DATA19	R[3]	RSDS_R[1]_P
LCD_DATA18	RIN[2]	VDCE0_VO_DATA18	R[2]	RSDS_R[1]_N
LCD_DATA17	RIN[1]	VDCE0_VO_DATA17	R[1]	RSDS_R[0]_P
LCD_DATA16	RIN[0]	VDCE0_VO_DATA16	R[0]	RSDS_R[0]_N
LCD_DATA15	GIN[7]	VDCE0_VO_DATA15	G[7]	RSDS_G[3]_P
LCD_DATA14	GIN[6]	VDCE0_VO_DATA14	G[6]	RSDS_G[3]_N
LCD_DATA13	GIN[5]	VDCE0_VO_DATA13	G[5]	RSDS_G[2]_P
LCD_DATA12	GIN[4]	VDCE0_VO_DATA12	G[4]	RSDS_G[2]_N
LCD_DATA11	GIN[3]	VDCE0_VO_DATA11	G[3]	RSDS_G[1]_P
LCD_DATA10	GIN[2]	VDCE0_VO_DATA10	G[2]	RSDS_G[1]_N
LCD_DATA9	GIN[1]	VDCE0_VO_DATA9	G[1]	RSDS_G[0]_P
LCD_DATA8	GIN[0]	VDCE0_VO_DATA8	G[0]	RSDS_G[0]_N
LCD_DATA7	BIN[7]	VDCE0_VO_DATA7	B[7]	RSDS_B[3]_P
LCD_DATA6	BIN[6]	VDCE0_VO_DATA6	B[6]	RSDS_B[3]_N
LCD_DATA5	BIN[5]	VDCE0_VO_DATA5	B[5]	RSDS_B[2]_P
LCD_DATA4	BIN[4]	VDCE0_VO_DATA4	B[4]	RSDS_B[2]_N
LCD_DATA3	BIN[3]	VDCE0_VO_DATA3	B[3]	RSDS_B[1]_P
LCD_DATA2	BIN[2]	VDCE0_VO_DATA2	B[2]	RSDS_B[1]_N
LCD_DATA1	BIN[1]	VDCE0_VO_DATA1	B[1]	RSDS_B[0]_P
LCD_DATA0	BIN[0]	VDCE0_VO_DATA0	B[0]	RSDS_B[0]_N
Pixel clock (LCD_CLK from video data channel selected by VDCECTL.VOSL)*1				
LCD_CLK	LCD_CLK	VDCE0_VO_CLKP	VDCE0_VO_CLKP	RSDS_CLK_P
		VDCE0_VO_CLKN	–	RSDS_CLK_N

Note 1. The clock signals are not affected by R – B swap and endian selection.

(b) VDCEn video output format converter settings (R – B swap, little endian)

– OUT_SET.OUT_FORMAT[1:0] = 00_B (RGB888 output)

– OUT_SET.OUT_ENDIAN_ON = 0 (little endian)

– OUT_SET.OUT_SWAP_ON = 1 (R – B swap)

Table 37.71 D1M2(H) Video output format selection (R – B swap, little endian)

VDCEn video output data	VDCEn video output format converter output	Port signal name	LVTTL	RSDS
Color data signals (LCD_DATA[23:0] from video data channel selected by VDCECTL.VOSL)				
LCD_DATA23	BIN[7]	VDCE0_VO_DATA23	B[7]	RSDS_B[3]_P
LCD_DATA22	BIN[6]	VDCE0_VO_DATA22	B[6]	RSDS_B[3]_N
LCD_DATA21	BIN[5]	VDCE0_VO_DATA21	B[5]	RSDS_B[2]_P
LCD_DATA20	BIN[4]	VDCE0_VO_DATA20	B[4]	RSDS_B[2]_N
LCD_DATA19	BIN[3]	VDCE0_VO_DATA19	B[3]	RSDS_B[1]_P
LCD_DATA18	BIN[2]	VDCE0_VO_DATA18	B[2]	RSDS_B[1]_N
LCD_DATA17	BIN[1]	VDCE0_VO_DATA17	B[1]	RSDS_B[0]_P
LCD_DATA16	BIN[0]	VDCE0_VO_DATA16	B[0]	RSDS_B[0]_N
LCD_DATA15	GIN[7]	VDCE0_VO_DATA15	G[7]	RSDS_G[3]_P
LCD_DATA14	GIN[6]	VDCE0_VO_DATA14	G[6]	RSDS_G[3]_N
LCD_DATA13	GIN[5]	VDCE0_VO_DATA13	G[5]	RSDS_G[2]_P
LCD_DATA12	GIN[4]	VDCE0_VO_DATA12	G[4]	RSDS_G[2]_N
LCD_DATA11	GIN[3]	VDCE0_VO_DATA11	G[3]	RSDS_G[1]_P
LCD_DATA10	GIN[2]	VDCE0_VO_DATA10	G[2]	RSDS_G[1]_N
LCD_DATA9	GIN[1]	VDCE0_VO_DATA9	G[1]	RSDS_G[0]_P
LCD_DATA8	GIN[0]	VDCE0_VO_DATA8	G[0]	RSDS_G[0]_N
LCD_DATA7	RIN[7]	VDCE0_VO_DATA7	R[7]	RSDS_R[3]_P
LCD_DATA6	RIN[6]	VDCE0_VO_DATA6	R[6]	RSDS_R[3]_N
LCD_DATA5	RIN[5]	VDCE0_VO_DATA5	R[5]	RSDS_R[2]_P
LCD_DATA4	RIN[4]	VDCE0_VO_DATA4	R[4]	RSDS_R[2]_N
LCD_DATA3	RIN[3]	VDCE0_VO_DATA3	R[3]	RSDS_R[1]_P
LCD_DATA2	RIN[2]	VDCE0_VO_DATA2	R[2]	RSDS_R[1]_N
LCD_DATA1	RIN[1]	VDCE0_VO_DATA1	R[1]	RSDS_R[0]_P
LCD_DATA0	RIN[0]	VDCE0_VO_DATA0	R[0]	RSDS_R[0]_N
Pixel clock (LCD_CLK from video data channel selected by VDCECTL.VOSL)*1				
LCD_CLK	LCD_CLK	VDCE0_VO_CLKP	VDCE0_VO_CLKP	RSDS_CLK_P
		VDCE0_VO_CLKN	–	RSDS_CLK_N

Note 1. The clock signals are not affected by R – B swap and endian selection.

(c) VDCEn video output format converter settings (no R – B swap, big endian)

– OUT_SET.OUT_FORMAT[1:0] = 00_B (RGB888 output)

– OUT_SET.OUT_ENDIAN_ON = 1 (big endian)

– OUT_SET.OUT_SWAP_ON = 0 (no R – B swap)

Table 37.72 D1M2(H) Video output format selection (no R – B swap, big endian)

VDCEn video output data	VDCEn video output format converter output	Port signal name	LVTTL	RSDS
Color data signals (LCD_DATA[23:0] from video data channel selected by VDCECTL.VOSL)				
LCD_DATA23	RIN[0]	VDCE0_VO_DATA23	R[0]	RSDS_R[0]_P
LCD_DATA22	RIN[1]	VDCE0_VO_DATA22	R[1]	RSDS_R[0]_N
LCD_DATA21	RIN[2]	VDCE0_VO_DATA21	R[2]	RSDS_R[1]_P
LCD_DATA20	RIN[3]	VDCE0_VO_DATA20	R[3]	RSDS_R[1]_N
LCD_DATA19	RIN[4]	VDCE0_VO_DATA19	R[4]	RSDS_R[2]_P
LCD_DATA18	RIN[5]	VDCE0_VO_DATA18	R[5]	RSDS_R[2]_N
LCD_DATA17	RIN[6]	VDCE0_VO_DATA17	R[6]	RSDS_R[3]_P
LCD_DATA16	RIN[7]	VDCE0_VO_DATA16	R[7]	RSDS_R[3]_N
LCD_DATA15	GIN[0]	VDCE0_VO_DATA15	G[0]	RSDS_G[0]_P
LCD_DATA14	GIN[1]	VDCE0_VO_DATA14	G[1]	RSDS_G[0]_N
LCD_DATA13	GIN[2]	VDCE0_VO_DATA13	G[2]	RSDS_G[1]_P
LCD_DATA12	GIN[3]	VDCE0_VO_DATA12	G[3]	RSDS_G[1]_N
LCD_DATA11	GIN[4]	VDCE0_VO_DATA11	G[4]	RSDS_G[2]_P
LCD_DATA10	GIN[5]	VDCE0_VO_DATA10	G[5]	RSDS_G[2]_N
LCD_DATA9	GIN[6]	VDCE0_VO_DATA9	G[6]	RSDS_G[3]_P
LCD_DATA8	GIN[7]	VDCE0_VO_DATA8	G[7]	RSDS_G[3]_N
LCD_DATA7	BIN[0]	VDCE0_VO_DATA7	B[0]	RSDS_B[0]_P
LCD_DATA6	BIN[1]	VDCE0_VO_DATA6	B[1]	RSDS_B[0]_N
LCD_DATA5	BIN[2]	VDCE0_VO_DATA5	B[2]	RSDS_B[1]_P
LCD_DATA4	BIN[3]	VDCE0_VO_DATA4	B[3]	RSDS_B[1]_N
LCD_DATA3	BIN[4]	VDCE0_VO_DATA3	B[4]	RSDS_B[2]_P
LCD_DATA2	BIN[5]	VDCE0_VO_DATA2	B[5]	RSDS_B[2]_N
LCD_DATA1	BIN[6]	VDCE0_VO_DATA1	B[6]	RSDS_B[3]_P
LCD_DATA0	BIN[7]	VDCE0_VO_DATA0	B[7]	RSDS_B[3]_N
Pixel clock (LCD_CLK from video data channel selected by VDCECTL.VOSL)*1				
LCD_CLK	LCD_CLK	VDCE0_VO_CLKP	VDCE0_VO_CLKP	RSDS_CLK_P
		VDCE0_VO_CLKN	–	RSDS_CLK_N

Note 1. The clock signals are not affected by R – B swap and endian selection.

(d) VDCEn video output format converter settings (R – B swap, big endian)

– OUT_SET.OUT_FORMAT[1:0] = 00_B (RGB888 output)

– OUT_SET.OUT_ENDIAN_ON = 1 (big endian)

– OUT_SET.OUT_SWAP_ON = 1 (R – B swap)

Table 37.73 D1M2(H) Video output format selection (R – B swap, big endian)

VDCEn video output data	VDCEn video output format converter output	Port signal name	LVTTL	RSDS
Color data signals (LCD_DATA[23:0] from video data channel selected by VDCECTL.VOSL)				
LCD_DATA23	BIN[0]	VDCE0_VO_DATA23	B[0]	RSDS_B[0]_P
LCD_DATA22	BIN[1]	VDCE0_VO_DATA22	B[1]	RSDS_B[0]_N
LCD_DATA21	BIN[2]	VDCE0_VO_DATA21	B[2]	RSDS_B[1]_P
LCD_DATA20	BIN[3]	VDCE0_VO_DATA20	B[3]	RSDS_B[1]_N
LCD_DATA19	BIN[4]	VDCE0_VO_DATA19	B[4]	RSDS_B[2]_P
LCD_DATA18	BIN[5]	VDCE0_VO_DATA18	B[5]	RSDS_B[2]_N
LCD_DATA17	BIN[6]	VDCE0_VO_DATA17	B[6]	RSDS_B[3]_P
LCD_DATA16	BIN[7]	VDCE0_VO_DATA16	B[7]	RSDS_B[3]_N
LCD_DATA15	GIN[0]	VDCE0_VO_DATA15	G[0]	RSDS_G[0]_P
LCD_DATA14	GIN[1]	VDCE0_VO_DATA14	G[1]	RSDS_G[0]_N
LCD_DATA13	GIN[2]	VDCE0_VO_DATA13	G[2]	RSDS_G[1]_P
LCD_DATA12	GIN[3]	VDCE0_VO_DATA12	G[3]	RSDS_G[1]_N
LCD_DATA11	GIN[4]	VDCE0_VO_DATA11	G[4]	RSDS_G[2]_P
LCD_DATA10	GIN[5]	VDCE0_VO_DATA10	G[5]	RSDS_G[2]_N
LCD_DATA9	GIN[6]	VDCE0_VO_DATA9	G[6]	RSDS_G[3]_P
LCD_DATA8	GIN[7]	VDCE0_VO_DATA8	G[7]	RSDS_G[3]_N
LCD_DATA7	RIN[0]	VDCE0_VO_DATA7	R[0]	RSDS_R[0]_P
LCD_DATA6	RIN[1]	VDCE0_VO_DATA6	R[1]	RSDS_R[0]_N
LCD_DATA5	RIN[2]	VDCE0_VO_DATA5	R[2]	RSDS_R[1]_P
LCD_DATA4	RIN[3]	VDCE0_VO_DATA4	R[3]	RSDS_R[1]_N
LCD_DATA3	RIN[4]	VDCE0_VO_DATA3	R[4]	RSDS_R[2]_P
LCD_DATA2	RIN[5]	VDCE0_VO_DATA2	R[5]	RSDS_R[2]_N
LCD_DATA1	RIN[6]	VDCE0_VO_DATA1	R[6]	RSDS_R[3]_P
LCD_DATA0	RIN[7]	VDCE0_VO_DATA0	R[7]	RSDS_R[3]_N
Pixel clock (LCD_CLK from video data channel selected by VDCECTL.VOSL)				
LCD_CLK	LCD_CLK	VDCE0_VO_CLKP	VDCE0_VO_CLKP	RSDS_CLK_P
		VDCE0_VO_CLKN	–	RSDS_CLK_N

37.9.2.2 Timing Controller output signals assignment

The following table shows the relation between the VDECn TCON output signals and the port signal names.

Table 37.74 D1M2(H) Timing Controller output signals assignment

VDCEn video output data	VDCEn video output format converter output	Port signal name
LCD_TCON0	LCD_TCON0	VDCEn_VO_TCON0
LCD_TCON1	LCD_TCON1	VDCEn_VO_TCON1
LCD_TCON2	LCD_TCON2	VDCEn_VO_TCON2
LCD_TCON3	LCD_TCON3	VDCEn_VO_TCON3
LCD_TCON4	LCD_TCON4	VDCEn_VO_TCON4
LCD_TCON5	LCD_TCON5	VDCEn_VO_TCON5
LCD_TCON6	LCD_TCON6	VDCEn_VO_TCON6

37.9.3 D1M1A Video output configuration

37.9.3.1 D1M1A video output mode selection

Parallel 16-/18-/24-bit RGB, VODDR, Open LDI and Serial 8-bit RGB are assigned to the same alternative ports.

Therefore, these output functions must be selected by RSDSCFG register.

NOTE

P42_[15:11] functions (VDCE0_VO_TCONx) are not changed by RSDSCFG register.

Table 37.75 Pin assignment of P43_0-1, P44_0-11, P45_0-13 (1/2)

		Use case			
		Parallel RGB*1	**VODDR**	**Open LDI*2**	**Serial RGB*2*4**
Configuration					
RSDSCFG register	VODDR_OEN	0	1	0	0
	OLDI0_OEN	0	0	1	–
	SRGB1_EN	0	0	–*3	1
	Pin name	**Pin functions**			
1st alternative assignment	P43_0	VDCE0_TCON2 (HS)	VODDR_TCON2 (HS)		VDCE1_VO_TCON2 (HS)
	P43_1	VDCE0_TCON0 (VS)	VODDR_TCON0 (VS)		VDCE1_VO_TCON0 (VS)
	P44_0	VDCE0_VO_DATA23	VODDR_LCDOUT23		VDCE1_VO_DATA7
	P44_1	VDCE0_VO_DATA22	VODDR_LCDOUT22		VDCE1_VO_DATA6
	P44_2	VDCE0_VO_DATA21	VODDR_LCDOUT21		VDCE1_VO_DATA5
	P44_3	VDCE0_VO_DATA20	VODDR_LCDOUT20		VDCE1_VO_DATA4
	P44_4	VDCE0_VO_DATA19	VODDR_LCDOUT19		VDCE1_VO_DATA3
	P44_5	VDCE0_VO_DATA18	VODDR_LCDOUT18		VDCE1_VO_DATA2
	P44_6	VDCE0_VO_DATA17	VODDR_LCDOUT17		VDCE1_VO_DATA1
	P44_7	VDCE0_VO_DATA16	VODDR_LCDOUT16		VDCE1_VO_DATA0
	P44_8	VDCE0_VO_DATA15	VODDR_LCDOUT15		VDCE1_VO_CLK
	P44_9	VDCE0_VO_DATA14	VODDR_LCDOUT14		
	P44_10	VDCE0_VO_DATA13	VODDR_LCDOUT13		
	P44_11	VDCE0_VO_DATA12	VODDR_LCDOUT12		
	P45_0	VDCE0_VO_CLKP	VODDR_OUT0_CLK	OLDI0_CH0_CLKP	
	P45_1	VDCE0_VO_CLKN	VODDR_OUT1_CLK	OLDI0_CH0_CLKN	
	P45_2	VDCE0_VO_DATA11	VODDR_LCDOUT11	OLDI0_CH1_P	
	P45_3	VDCE0_VO_DATA10	VODDR_LCDOUT10	OLDI0_CH1_N	
	P45_4	VDCE0_VO_DATA9	VODDR_LCDOUT9	OLDI0_CH2_P	
	P45_5	VDCE0_VO_DATA8	VODDR_LCDOUT8	OLDI0_CH2_N	
	P45_6	VDCE0_VO_DATA7	VODDR_LCDOUT7	OLDI0_CH3_P	
	P45_7	VDCE0_VO_DATA6	VODDR_LCDOUT6	OLDI0_CH3_N	
	P45_8	VDCE0_VO_DATA5	VODDR_LCDOUT5	OLDI0_CH4_P	
	P45_9	VDCE0_VO_DATA4	VODDR_LCDOUT4	OLDI0_CH4_N	
	P45_10	VDCE0_VO_DATA3	VODDR_LCDOUT3		
	P45_11	VDCE0_VO_DATA2	VODDR_LCDOUT2		
	P45_12	VDCE0_VO_DATA1	VODDR_LCDOUT1		
	P45_13	VDCE0_VO_DATA0	VODDR_LCDOUT0		

Table 37.75 Pin assignment of P43_0-1, P44_0-11, P45_0-13 (2/2)

		Use case			
		Parallel RGB*1	**VODDR**	**Open LDI*2**	**Serial RGB*2*4**
2nd alternative assignment	P43_0	VDCE0_VO_TCON3 (DE)	VODDR_VO_TCON3 (DE)*5		VDCE1_VO_TCON3 (DE)

Note 1. VDCE0 and VDCE1 output can be swapped by VDCECTL.VOSL.

Note 2. Pins not used for the selected video output mode can be used for other functions.

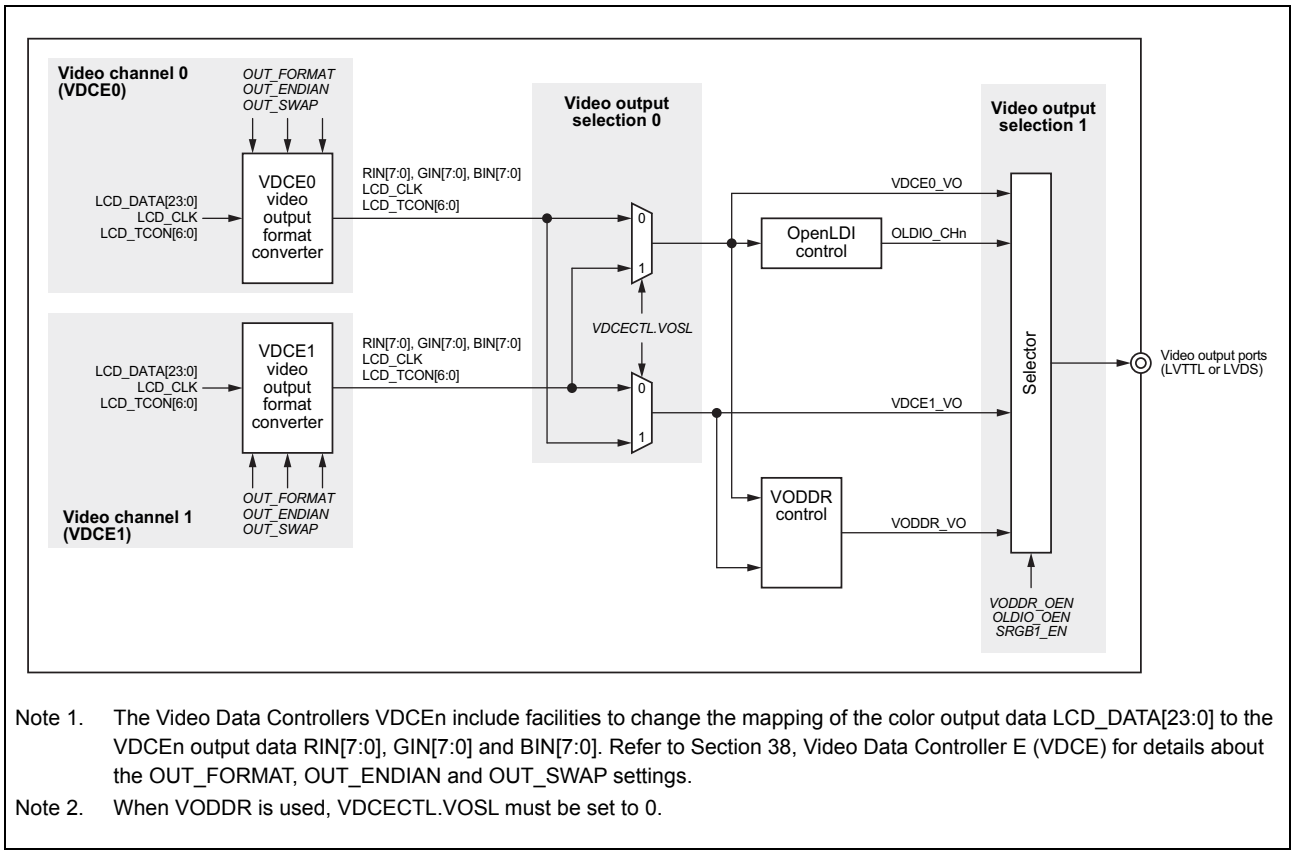
Note 3. When OpenLDI is not used, OLDI0_OEN must be set to 0.

Note 4. Serial RGB is only available for VDCE1.

Note 5. Another alternative, when all 3 sync signals (HS+VS and DE) are required for VODDR at the same time, is to use VDCE0_VO_TCON3 (DE) on P42_11 (AF1 Out). This port pin selection is not controlled by the RSDSCFG register.

37.9.3.2 D1M1A Video output selection

The diagram below illustrates the video output selection options.



Note 1. The Video Data Controllers VDCEn include facilities to change the mapping of the color output data LCD_DATA[23:0] to the VDCEn output data RIN[7:0], GIN[7:0] and BIN[7:0]. Refer to Section 38, Video Data Controller E (VDCE) for details about the OUT_FORMAT, OUT_ENDIAN and OUT_SWAP settings.

Note 2. When VODDR is used, VDCECTL.VOSL must be set to 0.

Figure 37.22 D1M1A video output selections

By use of the VDCECTL.VOSL selector the output signals from the video channels 0 and 1 can be swapped, so that any of both can be used for the OpenLDI output. The other video output is output as LVTTL signals.

Table 37.76 D1M1A video output selections

VDCECTL.	RSDSCFG.			
VOSL	VODDR_OEN	OLDIO_OEN	SRGB1_OEN	Video output ports
0	0	0	0	Video Channel 0: Parallel RGB
	1			VODDR: VODDR output
	0	1	– *1	OpenLDI (Video Channel 0): OpenLDI (LVDS) output
		– *1	1	Video Channel 1: Serial RGB
1	0	0	0	Video Channel 1: Parallel RGB
	1			VODDR: VODDR output
	0	1	– *1	OpenLDI (Video Channel 1): OpenLDI (LVDS) output

Note 1. OpenLDI output ports and Serial RGB ports from VDCE1_VO are not overlapping.

37.9.4 D1M1-V2 Video output configuration

37.9.4.1 D1M1-V2 video output mode selection

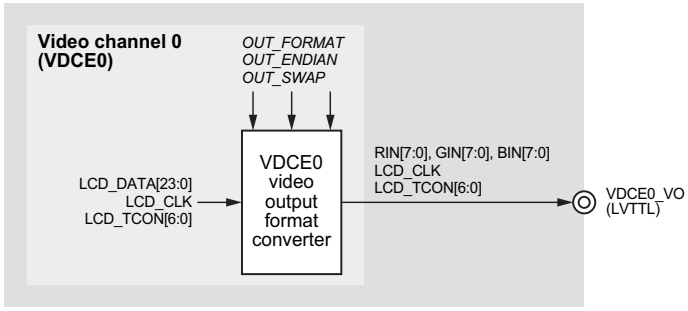
Parallel 16-/18-/24-bit RGB and Serial 8-bit RGB are assigned to the same alternative ports. These output function must be selected by the OUT_SET.OUT_FRQ_SEL[1:0] register.

Table 37.77 Pin assignment of P44_0-11, P45_0-13

		Use case	
		Parallel RGB	**Serial RGB**
Configuration			
OUT_SET register	OUT_FRQ_SEL[1:0]	0	1, 2
	Pin name	**Pin functions**	
1st alternative assignment	P44_0	VDCE0_VO_DATA23	
	P44_1	VDCE0_VO_DATA22	
	P44_2	VDCE0_VO_DATA21	
	P44_3	VDCE0_VO_DATA20	
	P44_4	VDCE0_VO_DATA19	
	P44_5	VDCE0_VO_DATA18	
	P44_6	VDCE0_VO_DATA17	
	P44_7	VDCE0_VO_DATA16	
	P44_8	VDCE0_VO_DATA15	
	P44_9	VDCE0_VO_DATA14	
	P44_10	VDCE0_VO_DATA13	
	P44_11	VDCE0_VO_DATA12	
	P45_0	VDCE0_VO_CLKP	VDCE0_VO_CLKP
	P45_1	VDCE0_VO_CLKN	VDCE0_VO_CLKN
	P45_2	VDCE0_VO_DATA11	
	P45_3	VDCE0_VO_DATA10	
	P45_4	VDCE0_VO_DATA9	
	P45_5	VDCE0_VO_DATA8	
	P45_6	VDCE0_VO_DATA7	VDCE0_VO_DATA7
	P45_7	VDCE0_VO_DATA6	VDCE0_VO_DATA6
	P45_8	VDCE0_VO_DATA5	VDCE0_VO_DATA5
	P45_9	VDCE0_VO_DATA4	VDCE0_VO_DATA4
	P45_10	VDCE0_VO_DATA3	VDCE0_VO_DATA3
	P45_11	VDCE0_VO_DATA2	VDCE0_VO_DATA2
	P45_12	VDCE0_VO_DATA1	VDCE0_VO_DATA1
	P45_13	VDCE0_VO_DATA0	VDCE0_VO_DATA0

37.9.4.2 D1M1-V2 Video output selection

The diagram below illustrates the video output selection options.



Note: The Video Data Controllers VDCEn include facilities to change the mapping of the color output data LCD_DATA[23:0] to the VDCEn output data RIN[7:0], GIN[7:0] and BIN[7:0]. Refer to Section 38, Video Data Controller E (VDCE) for details about the OUT_FORMAT, OUT_ENDIAN and OUT_SWAP settings.

Figure 37.23 D1M1-V2 video output signals

37.9.5 Video output clock inversion in LVTTL mode

If the video data VDCEn_VO_DATA[23:0] is output in LVTTL mode, the video data changes at the rising edge of the video output clock VDCEn_VO_CLK.

By use of the port output inversion function of the port control the video output clock can be inverted, thus the video data changes at the falling edge of the video output clock.

- VDCE0_VO_CLK
  - PINV45.PINV45_0 = 0: not inverted
    VDCE0_VO_DATA[23:0] changes at the rising edge of VDCE0_VO_CLK
  - PINV45.PINV45_0 = 1: inverted
    VDCE0_VO_DATA[23:0] changes at the falling edge of VDCE0_VO_CLK
- VDCE1_VO_CLK
  - PINV47.PINV47_8 = 0: not inverted
    VDCE1_VO_DATA[23:0] changes at the rising edge of VDCE1_VO_CLK
  - PINV47.PINV47_8 = 1: inverted
    VDCE1_VO_DATA[23:0] changes at the falling edge of VDCE1_VO_CLK

37.9.6 D1M2(H) RSDS control

37.9.6.1 Phase shift

RSDSCFG.RPHSL[1:0] selects a phase shift in 90 ° steps between the RSDS output clock VDCE0_VO_CLKP and the RSDS data output.

While RSDS data are output to terminal with 1 cycle delay from VDCE output, TCON signals are not delayed to terminal.
Therefore, when RSDS output is used, TCON signals should be delayed for 1 cycle. For this adjustment, please refer to Section 38.8.1.9, LCD TCON.

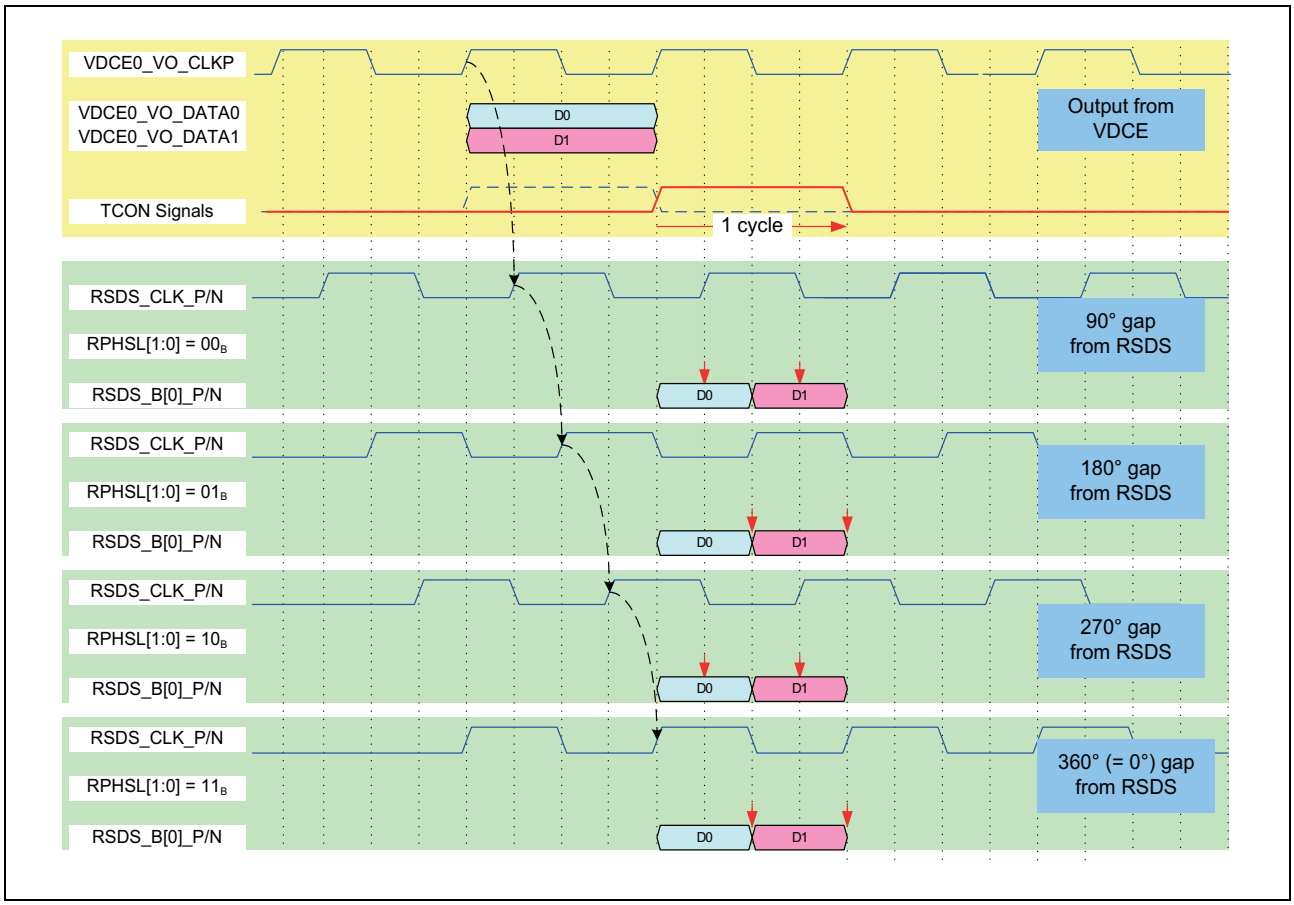


Figure 37.24 RSDS clock phase shift

37.9.6.2 RSDS data bit swap

RSDSCFG.RBSW selects a bit swap of the bits on a single RSDS data line:

- RBSW = 0: no swap, even number bit (i.e. data bits 0, 2, 4, 6) first
- RBSW = 1: swap, odd number bit (i.e. data bits 1, 3, 5, 7) first

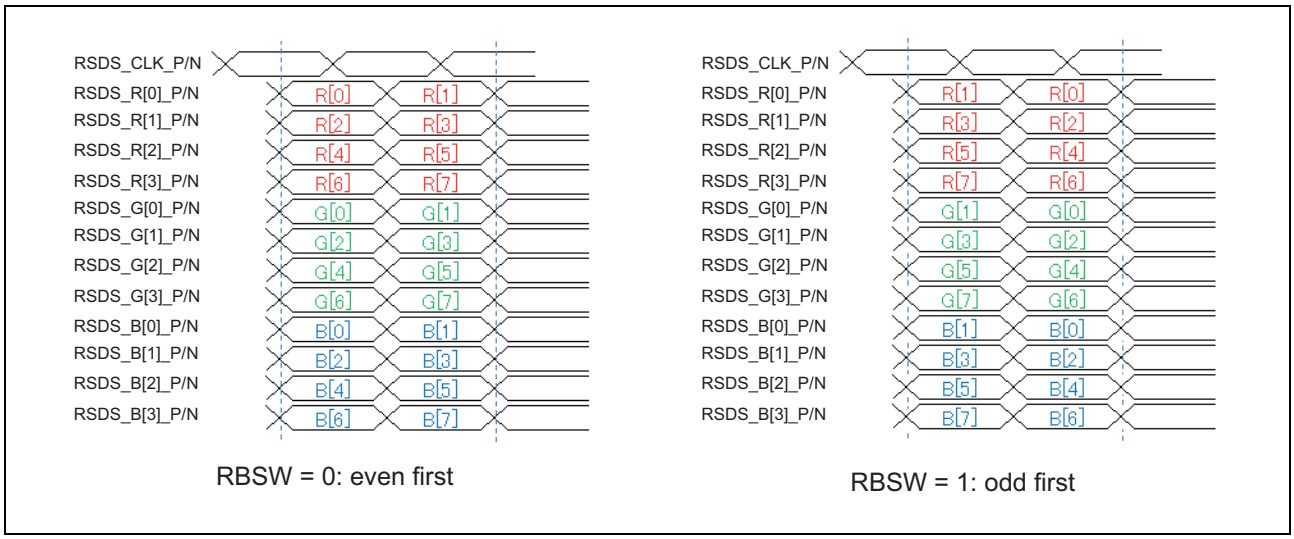


Figure 37.25 RSDS data bitswap

37.9.6.3 RSDS setup

1. Select alternative port function:
   - PMC44.PM44_[11:0] = FFF_H
   - PMC45.PM45_[13:0] = $3FFF_H$
2. Set ports as output:
   - PM44.PM44_[11:0] = FFF_H
   - PM45.PM45_[13:0] = $3FFF_H$
3. Set RSDSCFG.ROEN = 1 to select RSDS output
4. Set RSSCFG.RPHSL[1:0] and RSSCFG.RBSW as required.
5. Wait 10 µsec RSDS buffer stabilization time.

37.10 Video and graphics functions control registers

Table 37.78 List of video and graphics functions control registers

Module Name	Register Name	Symbol	Address
SELB	VDCE control registers	VDCECTL	FFC0 601C$_H$
SELB	RSDS configuration register	RSDSCFG	FFC0 6020$_H$
SELB	Sprite Engine update timing control register	SPEAUPDEN	FFC0 6048$_H$

NOTES

1. In the header files the names of the above registers are defined in the following format: <ModuleName> + <Symbol>.
   <ModuleName> and <Symbol> are defined in the above table.
2. Memory access right after register access has possibility to cause the racing between two accesses via different routes.
   Therefore, the last accessed register value must be confirmed at least before memory access. Refer to Section 3.4.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation for details.

37.10.1 Register details

37.10.1.1 VDCECTL — VDCE control register

Access: This register can be read/written in 32-bit units.

Address: FFC0 601C$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*1
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	VI1CTL	UPDT1	UPDT0	PXSL	VOSL	VISL1	VISL0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The initial value “0” of bit 16 must be changed to “1” and must not be changed afterwards.

Table 37.79 VDCECTL register contents

Bit Position	Bit Name	Function
31 to 17	Reserved	When written, write the initial value.
16	–	**CAUTION** The initial value “0” of bit 16 must be changed to “1” and must not be changed afterwards.
15 to 7	Reserved	When written, write the initial value.
6	VI1CTL	Selection of C_ISO_VI1PIXCLK (for D1M2(H) devices only) 0: C_ISO_VI1PIXCLK = VDCE1_VI_CLK 1: C_ISO_VI1PIXCLK = C_ISO_MIPIPIXCLK
5	UPDT1	Sprite Engine software update control This bit is used to generate the VUPDATE1 signal towards the Sprite Engine in sprite definition registers in software mode.
4	UPDT0	Sprite Engine software update control This bit is used to generate the VUPDATE0 signal towards the Sprite Engine in sprite definition registers in software mode.
3	PXSL	Video Input selection for VDCE0 (D1M2(H) only) 0: LVTTL 1: MIPI
2	VOSL	Video Output swap (for D1M2(H) and D1M1A devices only) 0: VDCE0 -> VO0 (LVTTL/RSDS) VDCE1 -> VO1 (LVTTL) 1: VDCE0 -> VO1 (LVTTL) VDCE1 -> VO0 (LVTTL/RSDS)
1	VISL1	Video Input color format for video channel 1 (for D1M2(H) devices only) 0: 8-bit ITU656 1: 24-bit RGB888
0	VISL0	Video Input color format for video channel 0 0: 8-bit ITU656 1: 24-bit RGB888

37.10.1.2 RSDSCFG — RSDS and other video output control register

Access: This register can be read/written in 32-bit units.

Address: FFC0 6020$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	SRGB1_OEN	0	0	OLDI_OEN	VODDR_OEN	ROEN	RPHSL[1:0]		RBSW
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.80 RSDSCFG register contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When written, write the initial value.
8	SRGB1_OEN	VDCE1 serial RGB output enabled/disable (D1M1A only) 0: VDCE1 serial RGB output disabled 1: VDCE1 serial RGB enabled **Note:** When this bit is set to 1, VODDR_OEN must be set to 0.
7 to 6	Reserved	When written, write the initial value.
5	OLDI_OEN	Open LDI output enabled/disable (D1M1A only) 0: Open LDI output disabled 1: Open LDI output enabled **Note:** When this bit is set to 1, VODDR_OEN must be set to 0. When Open LDI is not used, OLDI0_OEN must be set to 0.
4	VODDR_OEN	VODDR output enabled/disable (D1M1A only) VODDR_OEN selects the clock source for the Video Output Interface n (VDCEn) 0: VODDR output disabled VDCE0 source clock: CKSC_IVDCE0VOS_CTL output VDCE1 source clock: CKSC_IVDCE1VOS_CTL output 1: VODDR output enabled VDCE0 source clock: VODDR output VODDR_PIXCLK0 VDCE1 source clock: VODDR output VODDR_PIXCLK1 About clock generation from VODDR refer to Section 37.13, Video Output DDR format converter (VODDR) (D1M1A only). **Note:** When this bit is set to 1, SRGB1_OEN and OLDI0_OEN must be set to 0.
3	ROEN	RSDS output enabled/disable (D1M2(H) only) 0: RSDS output disabled (LVTTL output enabled) 1: RSDS output enabled (LVTTL output disabled)
2, 1	RPHSL[1:0]	RSDS output clock phase shift selection (D1M2(H) only) 00_B: 90° 01_B: 180° 10_B: 270° 11_B: 360°

Table 37.80 RSDSCFG register contents

Bit Position	Bit Name	Function
0	RBSW	RSDS bit swap selection (D1M2(H) only) 0: even bit first 1: odd bit first

37.10.1.3 SPEAUPDEN — Sprite Engine update timing control register

Access: This register can be read/written in 32-bit units.

Address: FFC0 6048$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UPDEN[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.81 SPEAUPDEN register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When written, write the initial value.
15 to 0	UPDEN[15:0]	Each of these bits enable the update of the Sprite Engine registers by the respective VE signal. 0: update by VE signal disabled 1: update by VE signal enabled Following the assignment of the bits to the VE sources • VDCE0 VE sources UPDEN0: VE of Image Synthesizer 00 UPDEN1: VE of Image Synthesizer 01 UPDEN2: VE of Image Synthesizer 02 UPDEN3: VE of Image Synthesizer 03 UPDEN4: VE of Output Image Generator UPDEN5: VE after alpha blending of Image Synthesizer 02 and Image Synthesizer 03 • VDCE1 VE sources (D1M2(H) and D1M1A only) UPDEN8: VE of Image Synthesizer 10 UPDEN9: VE of Image Synthesizer 11 UPDEN10: VE of Image Synthesizer 12 UPDEN11: VE of Image Synthesizer 13 UPDEN13: VE after alpha blending of Image Synthesizer 12 and Image Synthesizer 13 UPDEN[6,7,12,14,15]: Reserved

37.11 Video Output Warping Engine (VOWE) Ring Buffer

The Video Output Warping Engine (VOWE) writes rendered data to the memory. The Output Image Renderer (OIR) reads this data and forwards it to the Video Output module.

Since both, writing to the memory by the Video Output Warping Engine and reading the data by the Output Image Renderer is performed on the same timebase, which is determined by the frame geometry, framerate and pixel clock, it is in most cases not necessary to reserve memory for a complete output frame, since the OIR reads the VOWE data already before a complete frame has been stored.

A Ring Buffer in the memory that holds enough VOWE data for the OIR is sufficient to ensure continuous video data output.

The VOWE Ring Buffer holds data for an integer number of lines. Thus its size is calculated by

Ring Buffer size = (number of lines) x (line stride)

The Ring Buffer size must be a power of 2, hence the number of lines and the line stride have to be a power of 2 as well.

The Ring Buffer location in the memory is defined by its base address and mask.

The Ring Buffer base address RB_BASE_ADDR is defined by VOWEMAC.RBBADR[31:10].
It must be aligned to 1 KB boundaries, thus the lower 9 bit of the base address have to be all 0.

The mask is defined by VOWEMMC.RBMASK[31:0] = RB_SIZE - 1.
RB_SIZE denotes the Ring Buffer size.

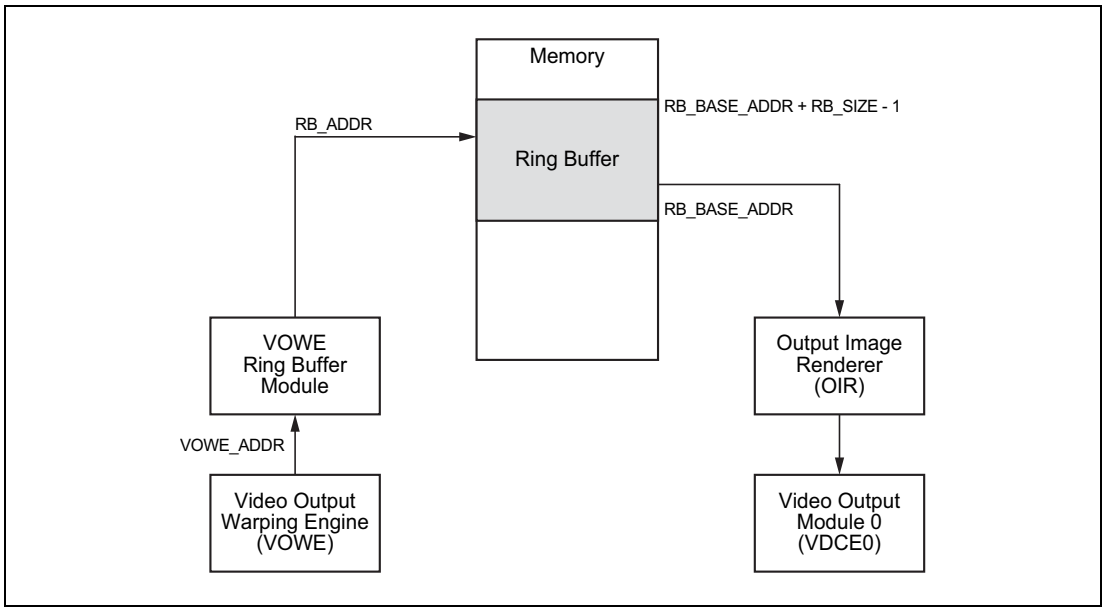


Figure 37.26 VOWE Ring Buffer module

The Ring Buffer address RB_ADDR, that is the address to write data the buffer in the memory, is calculated by

RB_ADDR = RB_BASE_ADDR + (VOWE_ADDR mod RB_SIZE)

VOWE_ADDR denotes the write address, issued by the VOWE.
“mod” denotes the modulo operator.

Example:

- horizontal size: 2048 32-bit pixel -> line stride = 2048 x 4 byte = 8 KB
- Ring Buffer shall hold 16 lines:
  -> RB_SIZE = 16 x 8 KB = 128 KB = 0002 0000_H
  -> VOWEMMC.RBMASK[31:0] = RB_SIZE - 1 = 0001 $FFFF_H$
- Ring Buffer base address RB_BASE_ADDR = VOWEMAC.RBBADR[31:0] = 4001 0000_H

If VOWE_ADDR = 1234 5670_H:

RB_ADDR = 4001 0000_H + (1234 5678_H mod 0002 0000_H)
= 4001 0000_H + 0000 5678_H = 4001 5678_H

Output Image Renderer loop addressing

The Output Image Renderer features a loop addressing function, that is to be used when reading the data from the Ring Buffer. Refer to Section 38.7, Output Image Generator for details.

37.11.1 VOWE Ring Buffer control registers

Table 37.82 List of the VOWE Ring Buffer control registers

Module Name	Register Name	Symbol	Address
SELB	VOWE Ring Buffer address offset register	VOWEMAC	FFC0 6040_H
SELB	VOWE Ring Buffer mask register	VOWEMMC	FFC0 6044_H

NOTES

1. In the header files the names of the above registers are defined in the following format:
   <ModuleName> + <Symbol>.
   <ModuleName> and <Symbol> are defined in the above table.
2. Memory access right after register access has possibility to cause the racing between two accesses via different routes.
   Therefore, the last accessed register value must be confirmed at least before memory access. Refer to Section 3.4.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation for details.

37.11.2 Register details

37.11.2.1 VOWEMAC — VOWE Ring Buffer base address register

Access: This register can be read/written in 32-bit units.

Address: FFC0 6040_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RBBADR[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBBADR[15:10]						0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R

Table 37.83 VOWEMAC register contents

Bit Position	Bit Name	Function
31 to 10	RBBADR[31:10]	Video Output Warping Engine Ring Buffer base address It must be aligned to 1 KB boundaries, thus the lower 10 bits of the base address are always 0.
9 to 0	Reserved	When written, write the initial value.

37.11.2.2 VOWEMMC — VOWE Ring Buffer mask register

Access: This register can be read/written in 32-bit units.

Address: FFC0 6044_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RBMASK[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBMASK[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.84 VOWEMMC register contents

Bit Position	Bit Name	Function
31 to 0	RBMASK[31:0]	Video Output Warping Engine Ring Buffer size RBMASK[31:0] = Ring Buffer size - 1

37.12 Open LDI Interface (D1M1A only)

37.12.1 Overview of Open LDI Interface

37.12.1.1 Units

This microcontroller has the following number of units of the Open LDI Interface.

Table 37.85 Number of Units

Product Name	D1L1	D1L2(H)	D1M1(H) D1M1-V2	D1M1A	D1M2(H)
Number of Units	–	–	–	1	–
Name	–	–	–	OLDIn (n = 0)	–

37.12.1.2 Indices

Following indices are used in this section:

Table 37.86 Indices

Index	Meaning
n	The individual Open LDI Interface units are generically indicated by the index “n”.

37.12.1.3 Register addresses

All OLDIn register addresses are given as address offsets from the individual base addresses <OLDIn_base>.

These base addresses are listed in the following table.

Table 37.87 Register base addressed <OLDIn_base>

OLDIn unit	<OLDIn_base> address
OLDI0	F200 2000H

37.12.1.4 Clock supply

All Open LDI Interfaces provide following clock inputs.

Table 37.88 Clock supply

OLDIn unit	OLDIn clock	Connected to
OLDI0	APB bus clock PCLK	Clock Controller C_ISO_PCLK
	Operation clock OLDICLK	Clock Controller CKSC_IVOEXS_CTL

37.12.1.5 Interrupts

The Open LDI Interface units do not generate any interrupt requests.

37.12.1.6 Reset sources

The Open LDI and their registers are initialized by the following reset signal:

Table 37.89 Reset sources

OLDIn unit	Reset signal
OLDI0	Reset Controller SYSRES

37.12.1.7 External Input/Output Signals

The external input/output signals of Open LDI are listed below.

Table 37.90 Video output signals connection

OLDIn signal	Function	Connected to *1
OLDI0_CH0_CLKP	OLDI0 differential clock (pos)	Video output mode selection OLDI0_CH0_CLKP
OLDI0_CH0_CLKN	OLDI0 differential clock (neg)	Video output mode selection OLDI0_CH0_CLKN
OLDI0_CH1_P	OLDI0 differential Ch1 data (pos)	Video output mode selection OLDI0_CH1_P
OLDI0_CH1_N	OLDI0 differential Ch1 data (neg)	Video output mode selection OLDI0_CH1_N
OLDI0_CH2_P	OLDI0 differential Ch2 data (pos)	Video output mode selection OLDI0_CH2_P
OLDI0_CH2_N	OLDI0 differential Ch2 data (neg)	Video output mode selection OLDI0_CH2_N
OLDI0_CH3_P	OLDI0 differential Ch3 data (pos)	Video output mode selection OLDI0_CH3_P
OLDI0_CH3_N	OLDI0 differential Ch3 data (neg)	Video output mode selection OLDI0_CH3_N
OLDI0_CH4_P	OLDI0 differential Ch4 data (pos)	Video output mode selection OLDI0_CH4_P
OLDI0_CH4_N	OLDI0 differential Ch4 data (neg)	Video output mode selection OLDI0_CH4_N

Note 1. The video output signals can be selected from several kinds of mode. Refer to Section 37.9.3.1, D1M1A video output mode selection.

37.12.2 Open LDI Overview

The Open LDI module converts an RGB signal output by the Video Output Controller E Unit module to the Open LDI format and outputs those signals.

The Open LDI interface supports 8 output data formats with the conversion formats selected by register settings. The output control signals can also be selected freely.

37.12.2.1 Features

This module has the following features.

- Output pins:
  Five differential output pairs (4 data and 1 clock) that conform to the TIA/EIA-644 standard.
- Operating frequency: maximum operation clock (OLDICLK) frequency 240 MHz
- Maximum dot clock frequency of 34 MHz
- Supports eight output data formats

37.12.2.2 Block Diagram

The figure below shows the block diagram of the Open LDI module.

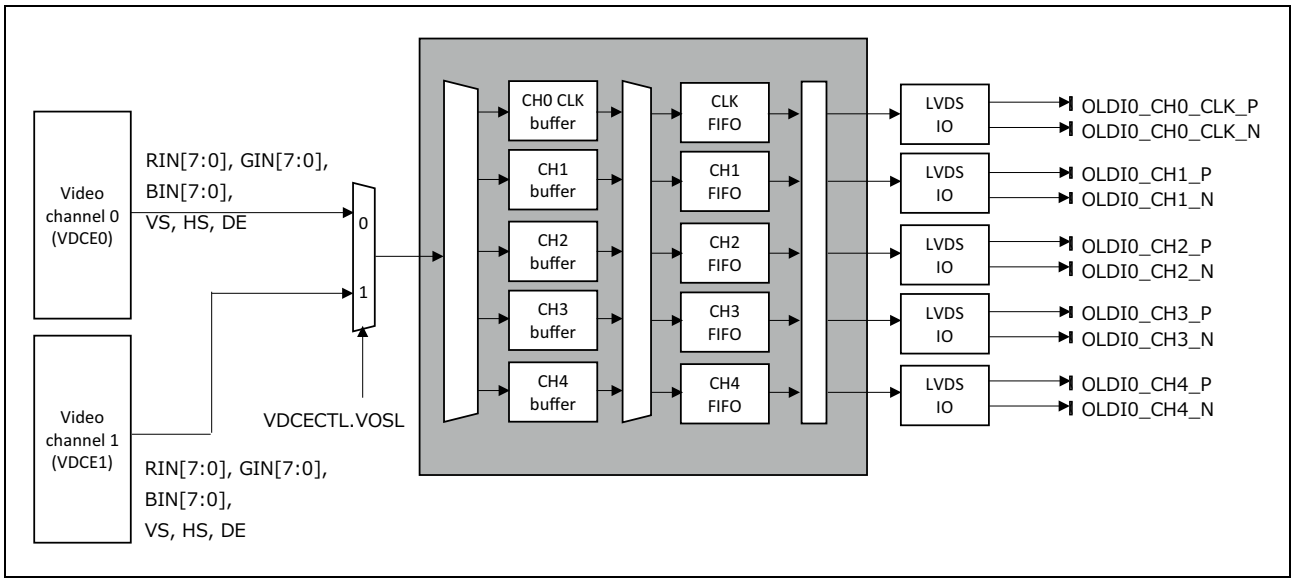


Figure 37.27 Open LDI Block Diagra

Table 37.91 Open LDI selections

OLDIn unit	VDCECTL.VOSL	Function
OLDI0	0	Video channel 0 (VDCE0)
	1	Video channel 1 (VDCE1)

37.12.2.3 Output signals

Table 37.92 Output signals

Pin Name	I/O	Connected to
OLDI0_CH0_CLKP	Output	OLDI0 differential clock (pos)
OLDI0_CH0_CLKN	Output	OLDI0 differential clock (neg)
OLDI0_CH1_P	Output	OLDI0 differential Ch1 data (pos)
OLDI0_CH1_N	Output	OLDI0 differential Ch1 data (neg)
OLDI0_CH2_P	Output	OLDI0 differential Ch2 data (pos)
OLDI0_CH2_N	Output	OLDI0 differential Ch2 data (neg)
OLDI0_CH3_P	Output	OLDI0 differential Ch3 data (pos)
OLDI0_CH3_N	Output	OLDI0 differential Ch3 data (neg)
OLDI0_CH4_P	Output	OLDI0 differential Ch4 data (pos)
OLDI0_CH4_N	Output	OLDI0 differential Ch4 data (neg)

The state of the Open LDI pin is determined by the OLDInCR register.

37.12.3 Register Description

The table below lists the register configuration.

Register Access Size

All registers can be accessed in 32-bit units.

Correct device operation is not guaranteed if any access size other than 32-bit is used to access these registers.

<OLDIn_base>

The base address <OLDIn_base> of the OLDIn is defined in section 37.12.1.3, Register addresses.

Table 37.93 Register Configuration

Register Name	Symbol	Address
OLDI control register 0	OLDInCR0	<OLDIn_base> + 00_H
OLDI control register 1	OLDInCR1	<OLDIn_base> + 04_H
OLDI CTR control register	OLDInCTRCR	<OLDIn_base> + $0C_H$
OLDI control register	OLDInCHCR	<OLDIn_base> + 10_H
OLDI skew control register	OLDInSKEWCTR	<OLDIn_base> + 70_H

37.12.3.1 OLDInCR0 – Open LDI control register 0

This register controls Open LDI function.

Access: This register can be read/written in 32-bit units.

Address: <OLDIn_base> + 00_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	LVMD[3:0]				–	–	–	–	–	–	LVEN	LVRES
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Table 37.94 OLDInCR0 register contents

Bit position	Bit name	Function
31 to 12	–	These bits are always read as 0. When written, write the initial value.
11 to 8	LVMD[3:0]	OpenLDI Mode Selects the Open LDI module output data format (see Figure 37.28, Output Data Format). 0000: MODE0 0001: MODE1 0010: MODE2 0011: MODE3 0100: MODE4 0101: MODE5 0110: MODE6 0111: MODE7 All other values: Setting prohibited
7 to 2	–	These bits are always read as 0. When written, write the initial value.
1	LVEN	Open LDI Enable Bit Controls the Open LDI operation. 0: Stopped 1: Normal operation Once Open LDI operation is started, do not change this bit to 0.
0	LVRES	Open LDI Reset Bit Controls the Open LDI output. 0: Output off 1: Output on Once Open LDI operation is started, do not change this bit to 0.

37.12.3.2 OLDInCR1 – Open LDI control register 1

This register controls Open LDI function.

Access: This register can be read/written in 32-bit units.

Address: <OLDIn_base> + 04_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	CH4STBY[1:0]		CH3STBY[1:0]		CH2STBY[1:0]		CH1STBY[1:0]		CLKSTBY[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.95 OLDInCR1 register contents

Bit position	Bit name	Function
31 to 10	–	These bits are always read as 0. When written, write the initial value.
9 to 8	CH4STBY[1:0]	CH4 Control Controls the CH4 pin. 00: Standby mode 01: Output off mode 11: Operating mode Other settings are prohibited. Once Open LDI operation is started, do not change this bit0 to 0.
7 to 6	CH3STBY[1:0]	CH3 Control Controls the CH3 pin. 00: Standby mode 01: Output off mode 11: Operating mode Other settings are prohibited. Once Open LDI operation is started, do not change this bit0 to 0.
5 to 4	CH2STBY[1:0]	CH2 Control Controls the CH2 pin. 00: Standby mode 01: Output off mode 11: Operating mode Other settings are prohibited. Once Open LDI operation is started, do not change this bit0 to 0.
3 to 2	CH1STBY[1:0]	CH1 Control Controls the CH1 pin. 00: Standby mode 01: Output off mode 11: Operating mode Other settings are prohibited. Once Open LDI operation is started, do not change this bit0 to 0.
1 to 0	CLKSTBY[1:0]	CLK Control Controls the CLK pin. 00: Standby mode 01: Output off mode 11: Operating mode Other settings are prohibited. Once Open LDI operation is started, do not change this bit0 to 0.

37.12.3.3 OLDInCTRCR – Open LDI CTR control register

This register controls Open LDI function.

Access: This register can be read/written in 32-bit units.

Address: <OLDIn_base> + $0C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	CTR2SEL[2:0]			–	CTR1SEL[2:0]			–	CTR0SEL[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 37.96 OLDInCTRCR register contents

Bit position	Bit name	Function
31 to 11	–	These bits are always read as 0. When written, write the initial value.
10 to 8	CTR2SEL[2:0]	Ctrl2 Select Selects data to be output to Ctrl2 (see section 37.12.4.3, CH Selection). 000: DE 011: HSYNC 100: VSYNC Other settings are prohibited.
7	–	These bits are always read as 0. When written, write the initial value.
6 to 4	CTR1SEL[2:0]	Ctrl1 Select Selects data to be output to Ctrl1 (see section 37.12.4.3, CH Selection). 000: VSYNC 001: DE 100: HSYNC Other settings are prohibited.
3	–	These bits are always read as 0. When written, write the initial value.
2 to 0	CTR0SEL[2:0]	Ctrl0 Select Selects data to be output to Ctrl0 (see section 37.12.4.3, CH Selection). 000: HSYNC 001: VSYNC 010: DE Other settings are prohibited.

37.12.3.4 OLDInCHCR – Open LDI CH control register

This register controls Open LDI function.

Access: This register can be read/written in 32-bit units.

Address: <OLDIn_base> + 10_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	CH4SEL[1:0]		–	–	CH3SEL[1:0]		–	–	CH2SEL[1:0]		–	–	CH1SEL[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Table 37.97 OLDInCHCR register contents

Bit position	Bit name	Function
31 to 14	–	These bits are always read as 0. When written, write the initial value.
13 to 12	CH4SEL[1:0]	CH4 Select Selects data to be output to CH4 (see section 37.12.4.3, CH Selection). 00: CH4 01: CH1 10: CH2 11: CH3
11 to 10	–	These bits are always read as 0. When written, write the initial value.
9 to 8	CH3SEL[1:0]	CH2 Select Selects data to be output to CH3 (see section 37.12.4.3, CH Selection). 00: CH3 01: CH4 10: CH1 11: CH2
7 to 6	–	These bits are always read as 0. When written, write the initial value.
5 to 4	CH2SEL[1:0]	CH2 Select Selects data to be output to CH2 (see section 37.12.4.3, CH Selection). 00: CH2 01: CH3 10: CH4 11: CH1
3 to 2	–	These bits are always read as 0. When written, write the initial value.
1 to 0	CH1SEL[1:0]	CH1 Select Selects data to be output to CH1 (see section 37.12.4.3, CH Selection). 00: CH1 01: CH2 10: CH3 11: CH4

37.12.3.5 OLDInSKEWCTR – Open LDI skew control register

This register controls Open LDI function.

Access: This register can be read/written in 32-bit units.

Address: <OLDIn_base> + 70_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	SKEWC[2:0]			SKEW4[2:0]			SKEW3[2:0]			SKEW2[2:0]			SKEW1[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.98 OLDInSKEWCTR register contents

Bit position	Bit name	Function
31 to 15	–	These bits are always read as 0. When written, write the initial value.
14 to 12	SKEWC[2:0]	Setting Clock Skew. 011 : +150 ps 010 : +100 ps 001 : +50 ps x00 : 0 ps 101 : - 50 ps 110 : - 100 ps 111 : - 150 ps
11 to 9	SKEW2[2:0]	Setting CH4 Skew.
8 to 6	SKEW3[2:0]	Setting CH3 Skew.
5 to 3	SKEW2[2:0]	Setting CH2 Skew.
2 to 0	SKEW1[2:0]	Setting CH1 Skew.

37.12.4 Operation

The Open LDI module converts the RGB signals output from a Video Data Controller E (VDCE) Unit to the Open LDI format and outputs those signals. Since the output data format can be selected by register settings, this module can convert to any of the Open LDI formats. The output data format is determined by (1) the mode selection, (2) the Ctrl signal selection, and (3) the CH selection.

The registers concerned with the output data format must be set before the Open LDI module is started and must not be changed during module operation.

37.12.4.1 Mode Selection

The mode is selected by the LVMD[3:0] bits in the Open LDI control register 0. **Figure 37.28** shows the output data format.

Here, R0 to R7, G0 to G7, and B0 to B7 are the RGB signals and Ctrl0 to Ctrl3 are control signals (such as HSYNC and VSYNC). The Ctrl signals can be set by the CTR control register. CH1, CH2, CH3, and CH4 are buffers that hold data temporarily. The default is for the CH1, CH2, CH3, and CH4 data to be output directly without change to OLDIn_CH1_P/N, OLDIn_CH2_P/N, OLDI0_CH3_P/N, and OLDIn_CH4_P/N. The CH assignment can be switched with the CH control register settings.

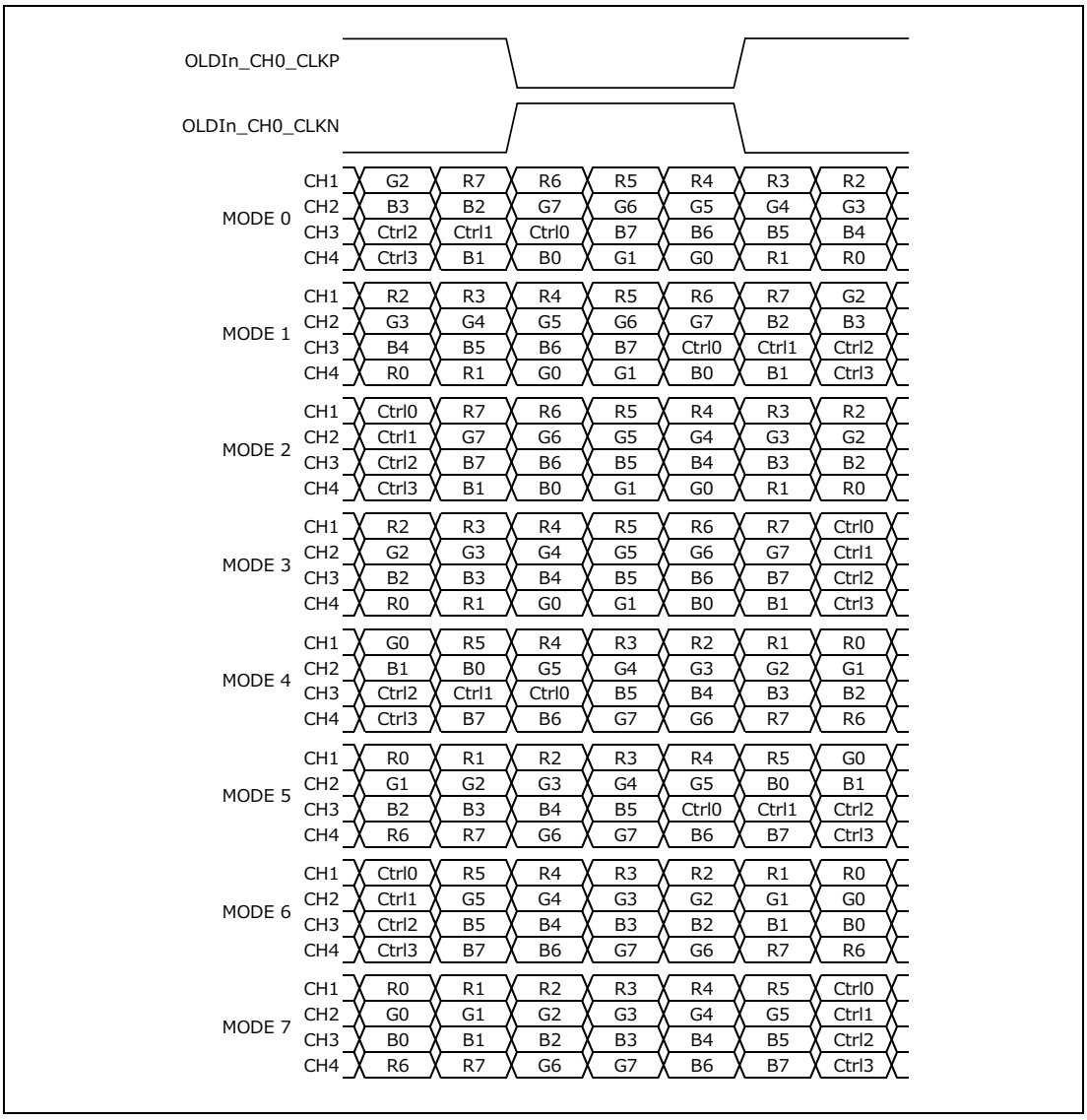


Figure 37.28 Output Data Format

37.12.4.2 Ctrl Signal Selection

The two stage settings shown below are required to select the Ctrl signals.

(1) Open LDI Port Selection

The Open LDI ports used to output the Ctrl signals are selected with the CTR control register (OLDInCTRCR). This results in the signals input to the Open LDI ports being output as Ctrl signals.

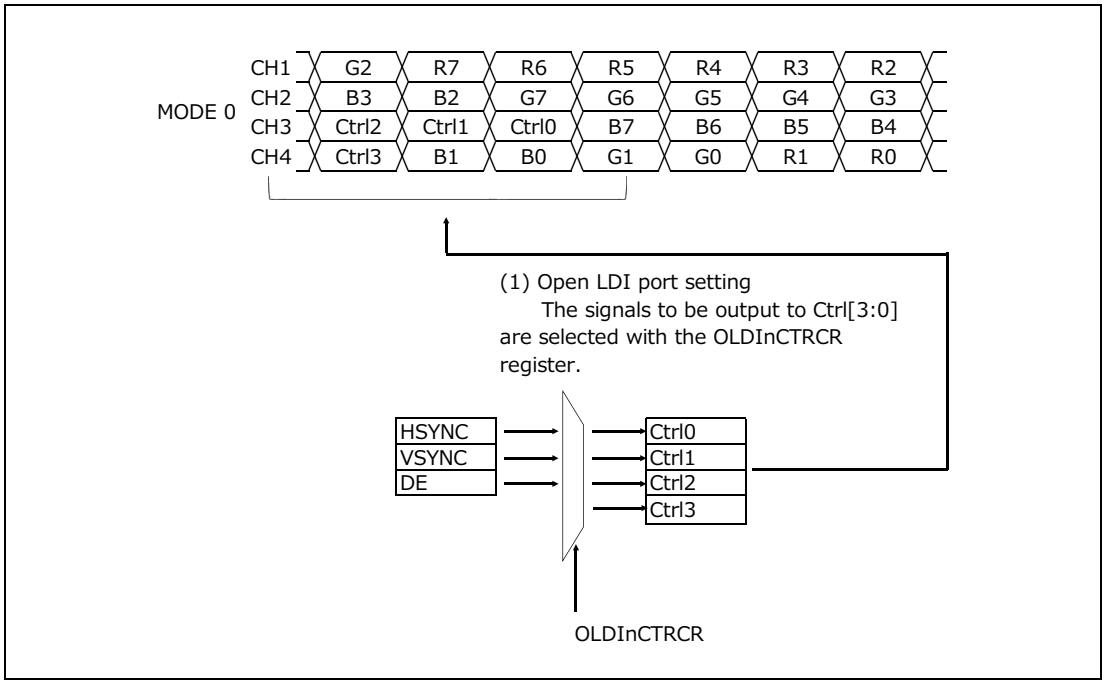


Figure 37.29 Ctrl Signal Selection

Set the desired Ctrl signal characteristics with the VDCE registers.

For example, to reverse the polarity of the HSYNC signal, the polarity must be set with VDCE register settings. (The Open LDI module only converts the signals output from VDCE to the Open LDI format.)

An example of Ctrl signal settings is shown below.

1. Ctrl0 = HSYNC, Ctrl1 = VSYNC, and Ctrl2 = DE

Set the VDCE registers so that HSYNC, VSYNC, and DE are output.

Then set the CTR control register (OLDInCTRCR) so that CTR0SEL[2:0] = 000, CTR1SEL[2:0] = 000, and CTR2SEL[2:0] = 000.

37.12.4.3 CH Selection

The Open LDI module stores the RGB signal data in CH1, CH2, CH3, and CH4 according to the mode selection and Ctrl signal selection registers.

The CH1, CH2, CH3, and CH4 data is then stored in the CH1, CH2, CH3, and CH4 FIFOs according to the CH selection register setting. The data stored in the CH1, CH2, CH3, and CH4 FIFOs is output from the external pins after passing through the LVDS buffers.

37.12.5 Usage Notes

The mode is selected by the LVMD[3:0] bits in the Open LDI control register 0. Figure 37.28, Output Data Format shows the output data format.

Here, R0 to R7, G0 to G7, and B0 to B7 are the RGB signals and Ctrl0 to Ctrl3 are control signals (such as HSYNC and VSYNC). The Ctrl signals can be set by the CTR control register. CH1, CH2, CH3, and CH4 are buffers that hold data temporarily. The default is for the CH1, CH2, CH3, and CH4 data to be output directly without change to OLDIn_CH1_P/N, OLDIn_CH2_P/N, OLDIn_CH3_P/N, and OLDIn_CH4_P/N. The CH assignment can be switched with the CH control register settings.

Sample Configuration flow of Open LDI

(1) Initialization

(1) Initialize Open LDI ports to port mode and input mode:
- PMCn_m = 0
- PMn_m = 1

(2) Set OLDInCR0 = 0:
- If OpenLDI was previously active, disable the output and stop the operation.

(3) Set RSDSCFG:
- VODDR output disable. (RSDSCFG.VODDR_OEN = 0)
- Open LDI output enable. (RSDSCFG.OLDI_OEN = 1)

(4) Set OLDInCR1:
- Select output off mode. (CLKSTBY/CH1STBY/CH2STBY/CH3STBY/CH4STBY[1:0] = 01)

(5) Set OLDInCTRCR:
- Configure the placement of synchronisation signals in data stream.

(6) Set OLDInCHCR:
- Select the data channel to be output on each of the data output pin pairs.

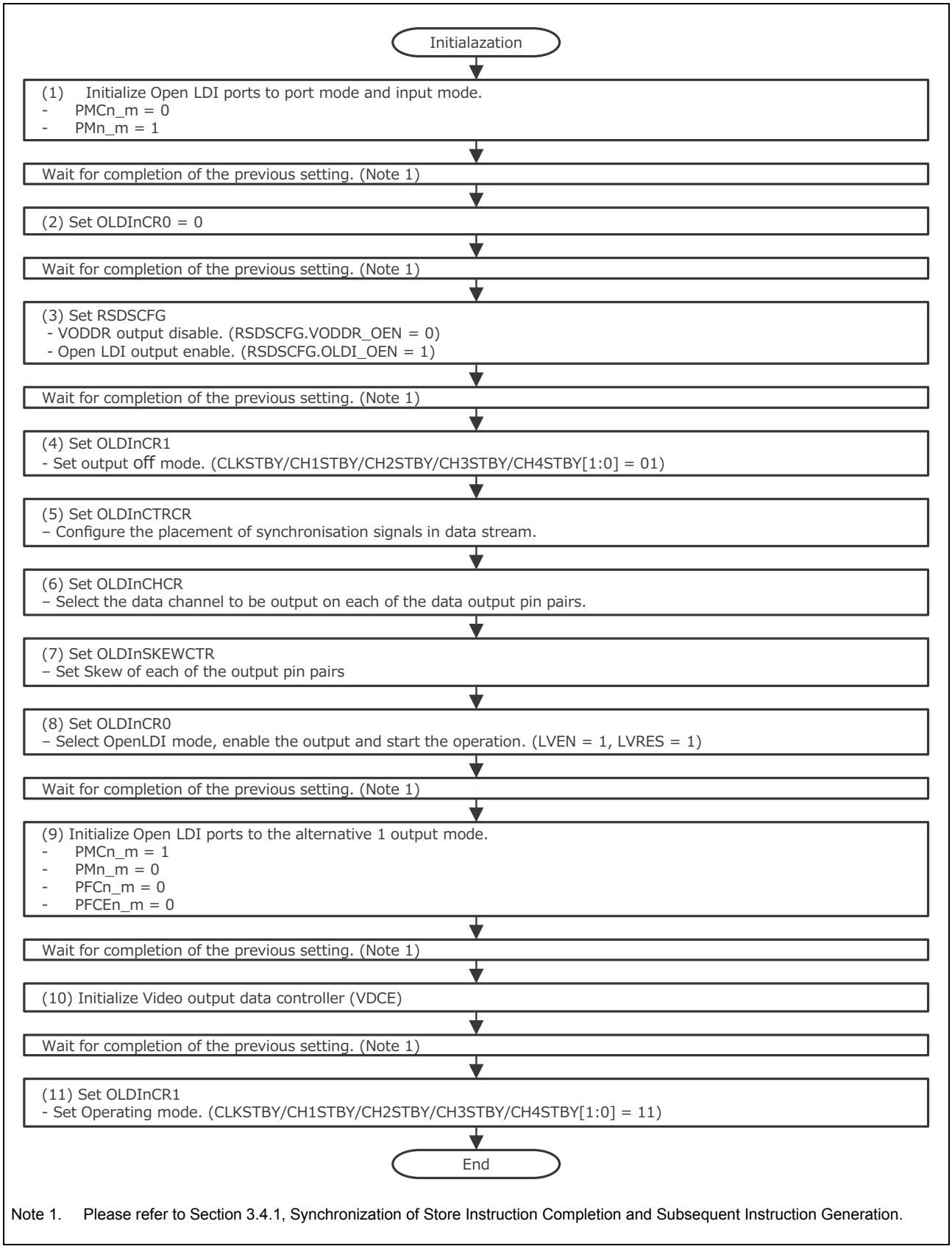
(7) Set OLDInSKEWCTR:
- Set Skew of each of the output pin pairs

(8) Set OLDInCR0:
- Select OpenLDI mode, enable the output and start the operation. (LVEN = 1, LVRES = 1)

(9) Initialize Open LDI ports to the alternative 1 output mode :
- PMCn_m = 1
- PMn_m = 0
- PFCn_m = 0
- PFCEn_m = 0

(10) Initialize Video data controller (VDCE)

(11) Set OLDI0CR1 :
- Select operating mode. (CLKSTBY/CH1STBY/CH2STBY/CH3STBY/CH4STBY[1:0] = 11)



Note 1. Please refer to Section 3.4.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation.

Figure 37.30 Open LDI initialization flow

(2) Shutdown

(12) Set OLDInCR1 :

– Select output off mode.
(CLKSTBY/CH1STBY/CH2STBY/CH3STBY/CH4STBY[1:0] = 01)

(13) De-initialize Open LDI ports to port mode :

– PMn_m = 1, PMCn_m = 0 or Pn_m = 0, PMn_m = 0, PMCn_m = 0 (Note 1)

NOTE

Ports get set to GPIO input (Hi-Z) with PMn_m = 1, PMCn_m = 0 or to GPIO output Low with Pn_m = 0, PMn_m = 0, PMCn_m = 0.

(3) 3. Recovery

(14) Re-initialize Open LDI ports to the alternative 1 output mode :

– PMCn_m = 1

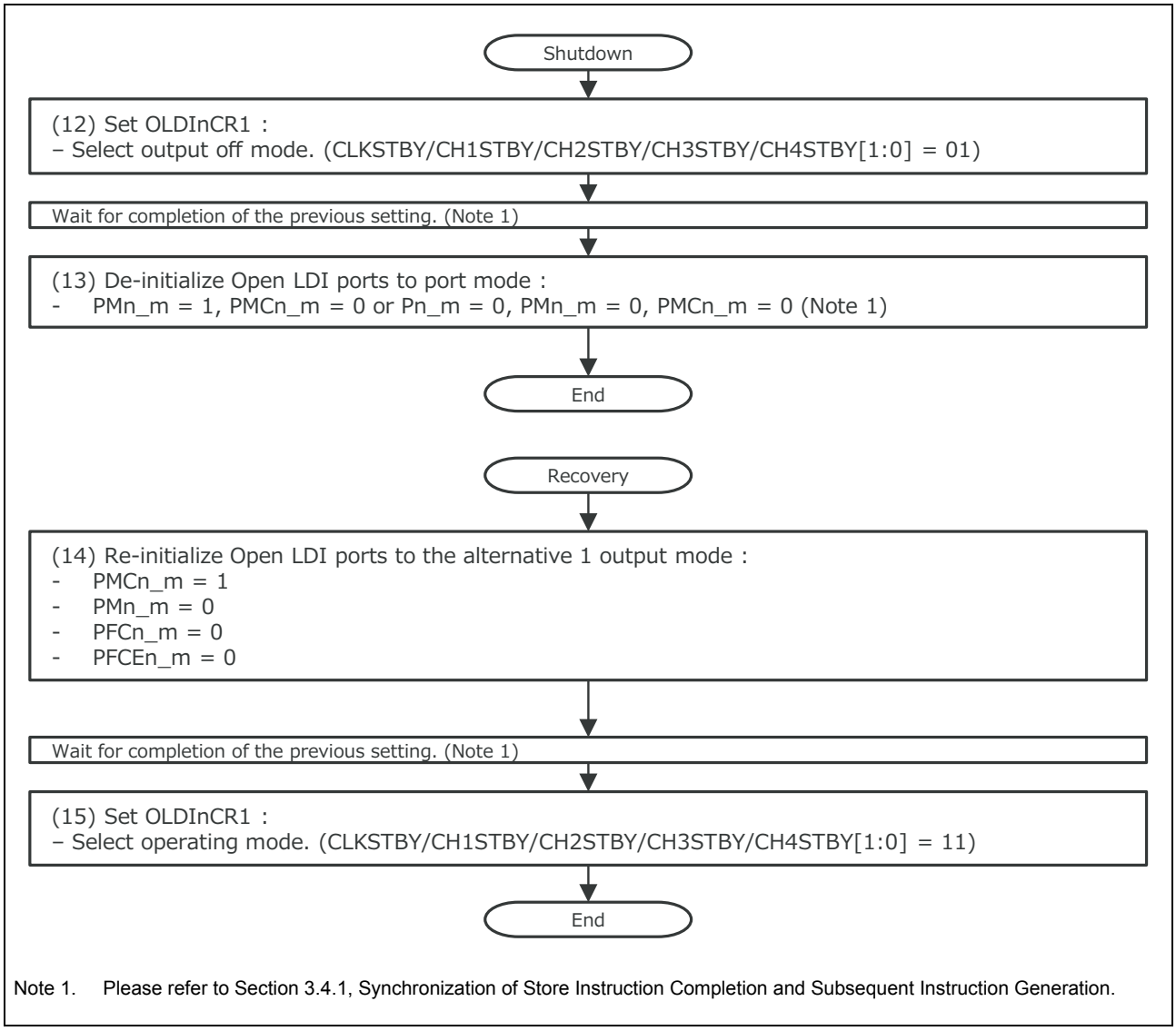
– PMn_m = 0

– PFCn_m = 0

– PFCEn_m = 0

(15) Set OLDInCR1:

– Select operating mode.
(CLKSTBY/CH1STBY/CH2STBY/CH3STBY/CH4STBY[1:0] = 11)



Note 1. Please refer to Section 3.4.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation.

Figure 37.31 Open LDI shutdown and recovery flow

NOTE

When OpenLDI is used (RSDSCFG.VODDR_OEN is set to 0 and RSDSCFG.OLDI_OEN is set to 1), P45_0 to P45_9 can not be used for other functions than Open LDI.

37.12.6 Limitation

1. Once Open LDI operation is started (OLDInCR0.LVRES = 1 and OLDInCR0.LVEN = 1), do not modify the Open LDI registers/bits as below.
   - OLDInCR0.LVRES = 0 (Stopped)
   - OLDInCR0.LVEN = 0 (Output off)
   - OLDInCR1.CLKSTBY/CH1STBY/CH2STBY/CH3STBY/CH4STBY[1:0] = 00 (Standby mode)
2. Do not disable C_ISO_VDCE0CLK or C_ISO_VDCE1CLK, while Open LDI operation clock (OLDICLK) is running.
   - CKSC_IVDCE0VOS_CTL.VDCE0VOSCSID[2:0] = 000B to disable C_ISO_VDCE0CLK (When using Open LDI Interface with video channel 0)
   - CKSC_IVDCE1VOS_CTL.VDCE1VOSCSID[2:0] = 000B to disable C_ISO_VDCE1CLK (When using Open LDI Interface with video channel 1)

When stopping the operation clock (OLDICLK), it is necessary to stop OLDICLK and IVOEXS0_OUTCLK at the same timing by using CKSC_IDOTCLK0S_CTL (When using Open LDI Interface with video channel 0) or CKSC_IDOTCLK1S_CTL (When using Open LDI Interface with video channel 1).

37.13 Video Output DDR format converter (VODDR) (D1M1A only)

37.13.1 Overview of VODDR

37.13.1.1 Units

This microcontroller has the following number of units of the VODDR.

Table 37.99 Number of Units

Product Name	D1L1	D1L2(H)	D1M1(H) D1M1-V2	D1M1A	D1M2(H)
Units	–	–	–	1	–
Names	–	–	–	VODDR0	–

37.13.1.2 Register addresses

The VODDR register addresses are given as address offsets from the individual base addresses <VODDR_base>.

These base addresses are listed in the following table.

Table 37.100 Register Base Address

Base Address Name	Base Address
<VODDR0_base>	F200 1000_H

37.13.1.3 Clock supply

All VODDR Interfaces provide following clock inputs.

Table 37.101 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
VODDR0	APB bus clock PCLK	Clock Controller C_ISO_PCLK
	Operation clock PLLCLK	Clock Controller C_ISO_VODDR_SYSCLK

37.13.1.4 Reset sources

The VODDR and their registers are initialized by the following reset signal:

Table 37.102 Reset sources

Unit Name	Reset Source
VODDR0	• Reset Controller SYSRES

37.13.1.5 External Input/Output Signals

The external input/output signals of VODDR are listed below.

Table 37.103 I/O signals connections

Unit Signal Name	Outline	Alternative Port Pin Signal*1
VODDR0:		
VODDR_LCDOUT[23:0]	Video output data	Video output mode selection VODDR_LCDOUT[23:0]
VODDR_TCON0(VS)	Video output synchronization signal (VSYNC)	Video output mode selection VODDR_TCON0(VS)
VODDR_TCON2(HS)	Video output synchronization signal (HSYNC)	Video output mode selection VODDR_TCON2(HS)
VODDR_TCON3(DE)	Video output synchronization signal (DE)	Video output mode selection VODDR_TCON3(DE)
VODDR_OUT0_CLK	Video output Ch0 pixel clock	Video output mode selection VODDR_OUT0_CLK
VODDR_OUT1_CLK	Video output Ch1 pixel clock	Video output mode selection VODDR_OUT1_CLK

Note 1. The video output signals can be selected from several kinds of format. Refer to Section 37.9.3, D1M1A Video output configuration.

37.13.2 Function Overview

The VODDR module converts two video output channels (VDCE0, VDCE1) to a signal set of one channel video output (LCDOUT[23:0], VSYNC, HSYNC, DE). Two video output pixel clocks are generated by dividing the PLLCLK.

This module has the following features.

- Video output format conversion
- Pixel clock generation
  - Maximum PLLCLK: max. 480MHz
  - Maximum pixel clock: max. 30 MHz. (VO0_PCLK, VO1_PCLK)
- Timing adjustment function of each channel
  - Data output timing adjustment
  - Clock phase adjustment

37.13.3 Function Description

The following figure provides a functional block diagram of the VODDR module.

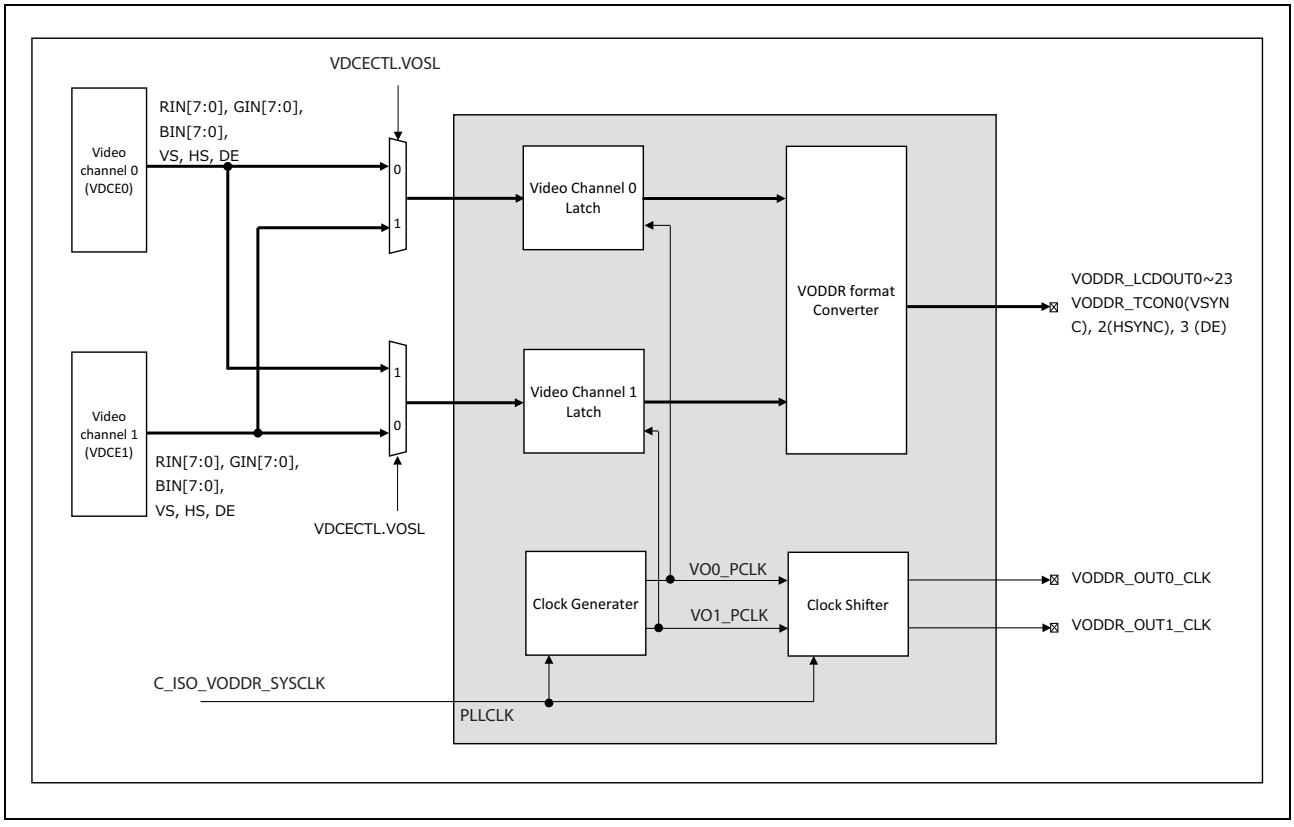


Figure 37.32 VODDR block diagram

37.13.3.1 Video output format conversion

The following figure shows examples of video output format conversion in VODDR module.

VODDR module mix two video output data streams to single video output signal set (LCDOUT23_0, TCON0:VS, TCON2:HS, TCON3:DE).

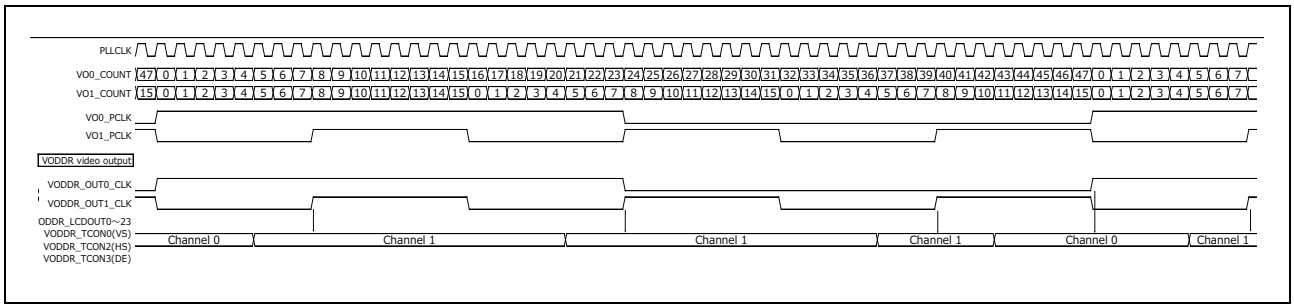


Figure 37.33 VODDR format conversion

37.13.3.2 Pixel clock generation

The following figure shows examples of pixel clock generation in VODDR module.

The frequency of the generated channel 0 pixel clock (VO0_PCLK) is defined by CLK_DIV0[7:0] and the frequency of PLLCLK as $f_{VO0\_PCLK} = f_{PLLCLK} / CLK\_DIV0[7:0]$. Also, the frequency of the generated channel 1 pixel clock (VO1_PCLK) is defined by CLK_DIV1[7:0] and the frequency of PLLCLK as $f_{VO1\_PCLK} = f_{PLLCLK} / CLK\_DIV1[7:0]$.

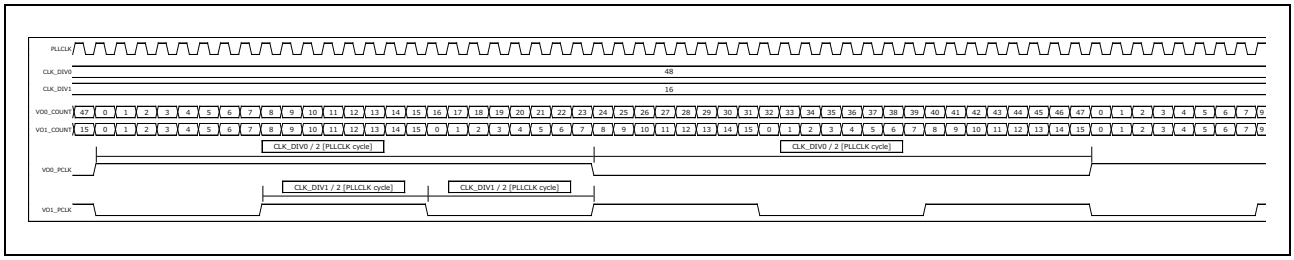


Figure 37.34 VODDR pixel clock generation

CLK_DIV0[7:0] and CLK_DIV1[7:0] must satisfy the followings.

1. The frequency of VO1_PCLK is larger than or equal to the frequency of VO0_PCLK
   - $f_{VO1\_PCLK} \geq f_{VO0\_PCLK}$
2. The frequency of VO0_PCLK and VO1_PCLK is less than or equal to maximum pixel clock frequency.
   - $CLK\_DIV0[7:0] \geq f_{PLLCLK} / f_{max.\ pixel\ clock}$: $30\ MHz \geq f_{VO0\_PCLK}$
   - $CLK\_DIV1[7:0] \geq f_{PLLCLK} / f_{max.\ pixel\ clock}$: $30\ MHz \geq f_{VO1\_PCLK}$
3. The values of CLK_DIV0[7:0] and CLK_DIV1[7:0] is even value.
   - CLK_DIV0[7:0] % 2 = 0
   - CLK_DIV1[7:0] % 2 = 0
4. VO1_PCLK frequency is an integer multiple of the VO0_PCLK frequency.
   - $f_{VO0\_PCLK} : f_{VO1\_PCLK} = 1 : 1, 1 : 2, 1 : 3, \ldots$

37.13.3.3 Timing adjustment function

(1) Data output timing adjustment

The following figure shows examples of video data output timing.

The video data output timing is defined by OUT_TIM0[2:0] and OUT_TIM1[2:0].

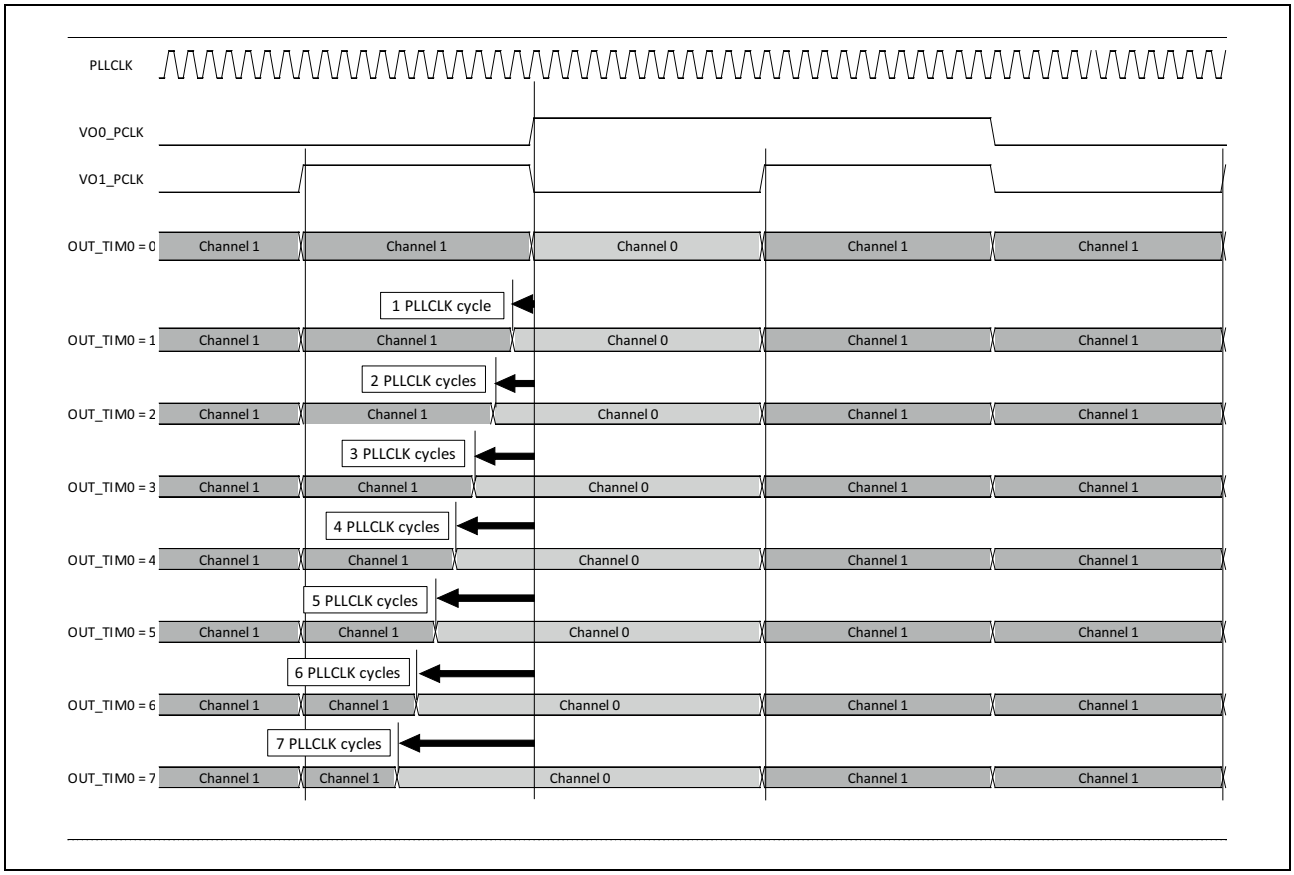


Figure 37.35 VODDR data output timing adjustment

(2) Clock phase adjustment

The following figure shows examples of pixel clock phase adjustment in VODDR module.

The VODDR_OUT0_CLK is shifted to CONFIG0[2:0] cycles behind. Also, the VODDR_OUT1_CLK is shifted to CONFIG1[2:0] cycles behind. The data output timing adjustment is performed the rise edge of VO0_PCLK or VO1_PCLK as a base point. Thus, the clock phase adjustment does not affect video data output timing.

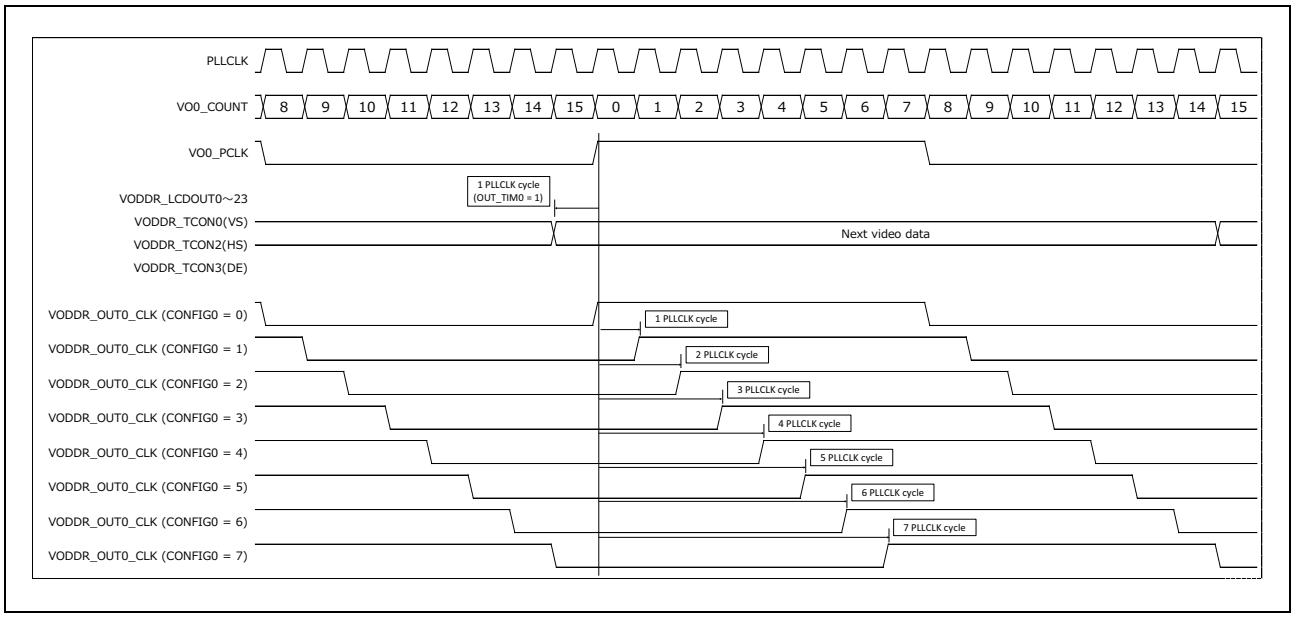


Figure 37.36 VODDR clock phase adjustment

37.13.4 Register Description

The table below lists the register configuration.

Register Access Size

All registers can be accessed in 32-bit units.

Correct device operation is not guaranteed if any access size other than 32-bit is used to access these registers.

<VODDR_base>

The base address <VODDR_base> of the VODDR is defined in Section 37.13.1.2, Register addresses.

Table 37.104 Register Configuration

Module Name	Register name	Shortcut	Address
VODDR0	System control register	VODDR0SYSCNT	<VODDR_base> + 00_H
	Clock divider control register	VODDR0CLKDIV	<VODDR_base> + 04_H
	Timing control 1 register	VODDR0TIMCNT1	<VODDR_base> + $0C_H$

37.13.4.1 VODDR0SYSCNT – VODDR System control register

This register controls VODDR function.

Access: This register can be read/written in 32-bit units.

Address: <VODDR_base> + 00_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VO1_ENABLE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLK_START
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 37.105 VODDR0SYSCNT register contents

Bit Position	Bit Name	Function
31 to 17	-	These bits are always read as 0. When written, write the initial value.
16	VO1_ENABLE	Video channel 1 enable When this bit is 1 (enable), video channel 1 is output. 0: Disable Video channel 1. (Only video channel 0 is output) 1: Enable Video channel 1. (Both of video channel 0 and video channel 1 is output)
15 to 1	-	These bits are always read as 0. When written, write the initial value.
0	CLK_START	Clock generation enable 0: Disable clock generation 1: Enable clock generation Before setting this bit to 1, it must be set to following registers. VODDR0CLKDIV, VODDR0TIMCNT1

37.13.4.2 VODDR0CLKDIV – VODDR Clock divider control register

This register controls VODDR function.

Access: This register can be read/written in 32-bit units.

Address: <VODDR_base> + 04_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CLK_DIV1[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CLK_DIV0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 37.106 VODDR0CLKDIV register contents

Bit Position	Bit Name	Function
31 to 24	-	These bits are always read as 0. When written, write the initial value.
23 to 16	CLK_DIV1[7:0]	Clock divider setting for video channel 1 This register must be set even value. 0: Setting prohibited. 1: Setting prohibited. 2: $f_{(VO1\_PCLK)} = f_{(PLLCLK)} / 2$ 3: Setting prohibited. 4: $f_{(VO1\_PCLK)} = f_{(PLLCLK)} / 4$ ... 254: $f_{(VO1\_PCLK)} = f_{(PLLCLK)} / 254$ 255: Setting prohibited.
15 to 8	-	These bits are always read as 0. When written, write the initial value.
7 to 0	CLK_DIV0[7:0]	Clock divider setting for video channel 0 This register must be set even value. 0: Setting prohibited. 1: Setting prohibited. 2: $f_{(VO0\_PCLK)} = f_{(PLLCLK)} / 2$ 3: Setting prohibited. 4: $f_{(VO0\_PCLK)} = f_{(PLLCLK)} / 4$ ... 254: $f_{(VO0\_PCLK)} = f_{(PLLCLK)} / 254$ 255: Setting prohibited.

NOTES

1. CLK_DIV0[7:0] and CLK_DIV1[7:0] must satisfy the following condition:
   $f_{(VO0\_PCLK)} \leq f_{(VO1\_PCLK)}$
2. While VODDR0SYSCNT.CLK_START = 1, the setting of this register is prohibited.

37.13.4.3 VODDR0TIMCNT1 – VODDR timing control 1 register

This register controls VODDR function.

Access: This register can be read/written in 32-bit units.

Address: <VODDR_base> + $0C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	OUT_TIM1[2:0]			—	—	—	—	—	OUT_TIM0[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	CONFIG1[2:0]			—	—	—	—	—	CONFIG0[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 37.107 VODDR0TIMCNT1 register contents

Bit Position	Bit Name	Function
31 to 27	-	These bits are always read as 0. When written, write the initial value.
26 to 24	OUT_TIM1[2:0]	Setting data output timing adjustment for video channel 1 Video data of channel 1 are output OUT_TIM1[2:0] cycles prior to rising edge of VO1_PCLK. 0: 0 PLLCLK cycle 1: 1 PLLCLK cycles … 7: 7 PLLCLK cycles
23 to 19	-	These bits are always read as 0. When written, write the initial value.
18 to 16	OUT_TIM0[2:0]	Setting data output timing adjustment for video channel 0 Video data of channel 0 are output OUT_TIM0[2:0] cycles prior to rising edge of VO0_PCLK. 0: 0 PLLCLK cycle 1: 1 PLLCLK cycles … 7: 7 PLLCLK cycles
15 to 11	-	These bits are always read as 0. When written, write the initial value.
10 to 8	CONFIG1[2:0]	Setting clock phase adjustment for video channel 1 The VODDR_OUT1_CLK is output shifted to CONFIG1[2:0] cycles behind. 0: 0 PLLCLK cycle 1: 1 PLLCLK cycle 2: 2 PLLCLK cycles … 7: 7 PLLCLK cycles
7 to 3	-	These bits are always read as 0. When written, write the initial value.
2 to 0	CONFIG0[2:0]	Setting clock phase adjustment for video channel 0 The VODDR_OUT0_CLK is output shifted to CONFIG0[2:0] cycles behind. 0: 0 PLLCLK cycle 1: 1 PLLCLK cycle 2: 2 PLLCLK cycles … 7: 7 PLLCLK cycles

NOTE

While VODDR0SYSCNT.CLK_START = 1, the setting of this register is prohibited.

37.13.5 Operation

(1) Start sequence for VODDR.

The following sequence describes the start sequence for VODDR.

1. Select VODDR format by RSDSCFG.VODDR_OEN = 1B
2. Set C_ISO_VODDR_SYSCLK by configuration of CKSC_IDOTCLK0S_CTL
3. Set up following registers:
   - VODDR0CLKDIV.CLK_DIV0[7:0]
   - VODDR0CLKDIV.CLK_DIV1[7:0]
   - VODDR0TIMCNT1.CONFIG0[2:0]
   - VODDR0TIMCNT1.CONFIG1[2:0]
   - VODDR0TIMCNT1.OUT_TIM0[2:0]
   - VODDR0TIMCNT1.OUT_TIM1[2:0]
4. Enable clock generation by VODDR0SYSCNT = 0001 0001H
5. Setting Video Data Controller E (VDCE)

(2) Re-start sequence for VODDR

The following sequence describes the re-start sequence for VODDR.

1. Disable clock generation by VODDR0SYSCNT = 0000 0000H
2. Set up following registers:
   - VODDR0CLKDIV.CLK_DIV0[7:0]
   - VODDR0CLKDIV.CLK_DIV1[7:0]
   - VODDR0TIMCNT1.CONFIG0[2:0]
   - VODDR0TIMCNT1.CONFIG1[2:0]
   - VODDR0TIMCNT1.OUT_TIM0[2:0]
   - VODDR0TIMCNT1.OUT_TIM1[2:0]
3. Enable clock generation by VODDR0SYSCNT = 0001 0001_H

37.13.5.1 Setting Example for WVGA + WQVGA

Table 37.108 and **Table 37.109** are setting example for using WVGA (800 x 480 @ 60Hz) + WQVGA (480 x 272 @ 60Hz) displays.

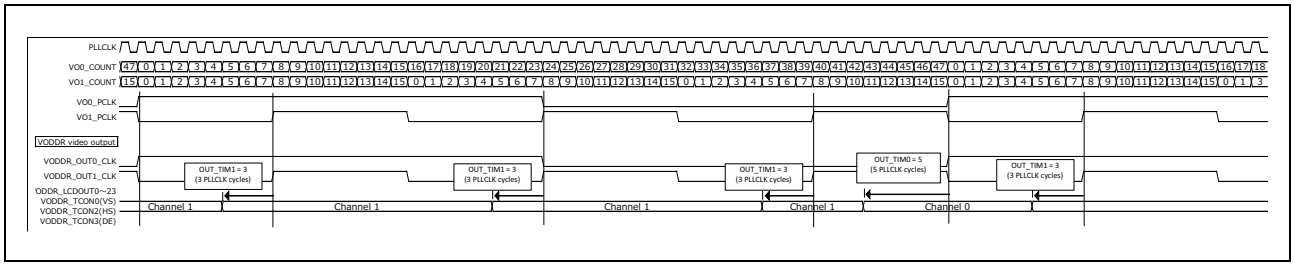


Figure 37.37 VODDR WVGA + WQVGA (60Hz) Video Output

Table 37.108 Video parameters for WVGA + WQVGA (60 Hz) display

Item		Settings
PLLCLK	PLLCLK frequency	480 MHz
Channel 0	Video image size	WQVGA
	Frame rate	60 Hz
	Vertical valid period	272 lines
	Horizontal valid period	480 pixels
	Pixel clock frequency	10 MHz
Channel 1	Video image size	WVGA
	Frame rate	60 Hz
	Vertical valid period	480 lines
	Horizontal valid period	800 pixels
	Pixel clock frequency	30 MHz

Table 37.109 VODDR Register Setting Example for WVGA + WQVGA (60 Hz) display

Register Name	Bit Name	Settings	Remarks
VODDR0CLKDIV	CLK_DIV0[7:0]	48	$f_{(VO0\_PCLK)} = f_{(PLLCLK)} / 48 = 10$ MHz
	CLK_DIV1[7:0]	16	$f_{(VO1\_PCLK)} = f_{(PLLCLK)} / 16 = 30$ MHz
VODDR0TIMCNT1	CONFIG0[2:0]	0	VODDR_OUT0_CLK is shifted 0 cycles for VO0_PCLK
	COMFIG1[2:0]	0	VODDR_OUT1_CLK is shifted 0 cycles for VO1_PCLK
	OUT_TIM0[2:0]	5	Video data of channel 0 are output before 5 cycles prior to rising edge of VO0_PCLK
	OUT_TIM1[2:0]	3	Video data of channel 1 are output before 3 cycles prior to rising edge of VO1_PCLK

Table 37.110 and **Table 37.111** are setting example for using WVGA (800 x 480 @ 60Hz) + WQVGA (480 x 272 @ 50Hz) displays.

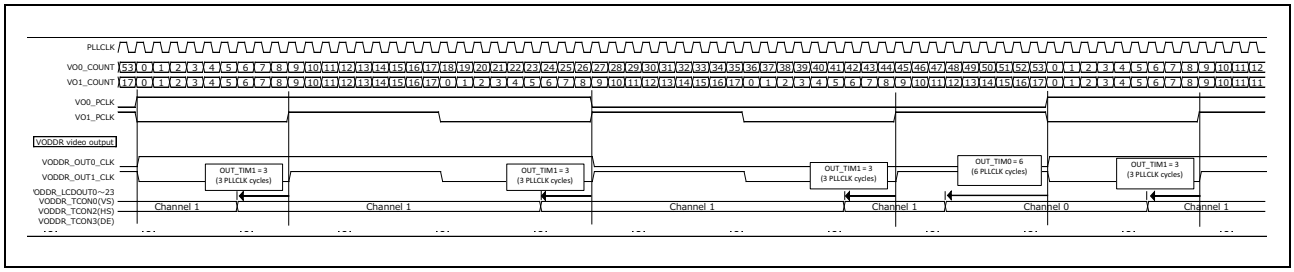


Figure 37.38 VODDR WVGA + WQVGA (50 Hz) Video Output

Table 37.110 Video parameters for WVGA + WQVGA (50 Hz) display

Item		Settings
PLLCLK	PLLCLK frequency	480MHz
Channel 0	Video image size	WQVGA
	Frame rate	50 Hz
	Vertical valid period	272 lines
	Horizontal valid period	480 pixels
	Pixel clock frequency	8.9 MHz
Channel 1	Video image size	WVGA
	Frame rate	60 Hz
	Vertical valid period	480 lines
	Horizontal valid period	800 pixels
	Pixel clock frequency	26.7 MHz

Table 37.111 VODDR Register Setting Example for WVGA + WQVGA (50Hz) display

Register Name	Bit Name	Settings	Remarks
VODDR0CLKDIV	CLK_DIV0[7:0]	54	$f_{(VO0\_PCLK)} = f_{(PLLCLK)} / 54 = 8.9$ MHz
	CLK_DIV1[7:0]	18	$f_{(VO1\_PCLK)} = f_{(PLLCLK)} / 18 = 26.7$ MHz
VODDR0TIMCNT1	CONFIG0[2:0]	0	VODDR_OUT0_CLK is shifted 0 cycles for VO0_PCLK
	COMFIG1[2:0]	0	VODDR_OUT1_CLK is shifted 0 cycles for VO1_PCLK
	OUT_TIM0[2:0]	6	Video data of channel 0 are output before 6 cycles prior to rising edge of VO0_PCLK
	OUT_TIM1[2:0]	3	Video data of channel 1 are output before 3 cycles prior to rising edge of VO1_PCLK

37.13.5.2 Setting Example for WQVGA + WQVGA

Table 37.112 and **Table 37.113** are setting example for using WQVGA (480 x 272 @ 60Hz) + WQVGA (480 x 272 @ 60Hz) displays.

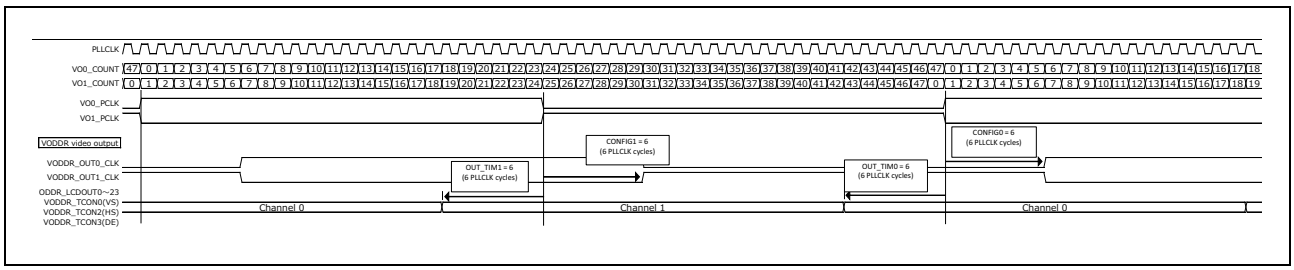


Figure 37.39 VODDR WQVGA + WQVGA Video Output

Table 37.112 Video parameters for WQVGA + WQVGA display

Item		Settings
PLLCLK	PLLCLK frequency	480 MHz
Channel 0	Video image size	WQVGA
	Frame rate	60 Hz
	Vertical valid period	272 lines
	Horizontal valid period	480 pixels
	Pixel clock frequency	10 MHz
Channel 1	Video image size	WQVGA
	Frame rate	60 Hz
	Vertical valid period	272 lines
	Horizontal valid period	480 pixels
	Pixel clock frequency	10 MHz

Table 37.113 VODDR Register Setting Example for WQVGA + WQVGA display

Register Name	Bit Name	Settings	Remarks
VODDR0CLKDIV	CLK_DIV0[7:0]	48	$f_{(VO0\_PCLK)} = f_{(PLLCLK)} / 48 = 10$ MHz
	CLK_DIV1[7:0]	48	$f_{(VO1\_PCLK)} = f_{(PLLCLK)} / 48 = 10$ MHz
VODDR0TIMCNT1	CONFIG0[2:0]	6	VODDR_OUT0_CLK is shifted 6 cycles for VO0_PCLK
	CONFIG1[2:0]	6	VODDR_OUT1_CLK is shifted 6 cycles for VO1_PCLK
	OUT_TIM0[2:0]	6	Video data of channel 0 are output before 6 cycles prior to rising edge of VO0_PCLK
	OUT_TIM1[2:0]	6	Video data of channel 1 are output before 6 cycles prior to rising edge of VO1_PCLK

Section 38 Video Data Controller E (VDCE)

This section contains a generic description of the Video Data Controller E.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

38.1 Overview of the RH850/D1L/D1M Video Data Controllers E

38.1.1 Units

This microcontroller has the following number of units of the Video Data Controller E.

Table 38.1 Units

Video Data Controller E (VDCE)	D1L1	D1L2(H)	D1M1(H) D1M1-V2	D1M1A D1M2(H)
Units	–	1	1	2
Names	–	VDCE0	VDCE0	VDCE0, VDCE1

Unit index n

Throughout this section, the individual units of the Video Data Controllers E are identified by the index “n” (n = 0, 1).

38.1.2 Register addresses

All Video Data Controller E register addresses are given as address offsets from the individual base addresses <VDCEn_base>.
The <VDCEn_base> addresses of each VDCEn are listed in the following table:

Table 38.2 Register base addresses <VDCEn_base>

VDCEn unit	<VDCEn_base> address
VDCE0	F200 4000_H
VDCE1	F200 6000_H

38.1.3 Clock supply

All Video Data Controllers E provide following clock inputs.

Table 38.3 Clock supply

VDCEn unit	VDCEn clock	Connected to
VDCE0	PCLK	Video channels clock generator C_ISO_XCCLK/2
	Cross-connect clock	Clock Controller C_ISO_XCCLK
	Video output pixel clock	Video channels clock generator C_ISO_VDCE0CLK
	Video input pixel clock	Video channels clock generator C_ISO_VI0PIXCLK
VDCE1	PCLK	Video channels clock generator C_ISO_XCCLK/2
	Cross-connect clock	Clock Controller C_ISO_XCCLK
	Video output pixel clock	Video channels clock generator C_ISO_VDCE1CLK
	Video input pixel clock	Video channels clock generator C_ISO_VI1PIXCLK

NOTE

For a detailed description of the video channels clock generator refer to Section 37.5, Video channels clock generation.

38.1.4 Interrupts

The Video Data Controllers E can generate the following interrupt requests:

Table 38.4 VDCEn interrupt requests (1/2)

VDCEn signals	Function	Connected to
VDCE0:		
S0_VSYNCERR	Missing Vsync signal for scaler 0	Interrupt Controller INTVDCE0ERR logical OR combination of all VDCE0 error interrupts
IV1_VBUFERR	Frame buffer write overflow signal for scaler 0	
IV3_VBUFERR	Frame buffer read underflow signal for graphics 0	
IV5_VBUFERR	Frame buffer read underflow signal for graphics 2	
IV6_VBUFERR	Frame buffer read underflow signal for graphics 3	
S1_VSYNCERR	Missing Vsync signal for scaler 1	
IV4_VBUFERR	Frame buffer read underflow signal for graphics 1	
V8_VBUFERR	Frame buffer read underflow signal for graphics (OIR)	
	VBLANK detection at Graphics 3	Interrupt Controller INTVDCE0GR3VBLANK
S0_VI_VSYNC	VSYNC input at Scaler 0 interrupt	Interrupt Controller INTVDCE0S0VIVSYNC
S0_LO_VSYNC	VSYNC output at Scaler 0 interrupt	Interrupt Controller INTVDCE0S0LOVSYNC
GR3_VLINE	Line detection of designated line at Graphics3 interrupt	Interrupt Controller INTVDCE0GR3VLINE

Table 38.4 VDCEn interrupt requests (2/2)

VDCEn signals	Function	Connected to
S0_VFIELD	End of field for record function at Scaler 0 interrupt	Interrupt Controller INTVDCE0S0VFIELD
S1_LO_VSYNC	VSYNC output at Scaler 1 interrupt	Interrupt Controller INTVDCE0S1LOVSYNC
OIR_VI_VSYNC	VSYNC input at Output Image Render interrupt	Interrupt Controller INTVDCE0OIRVIVSYNC
OIR_LO_VSYNC	VSYNC output at Output Image Render interrupt	Interrupt Controller INTVDCE0OIRLOVSYNC
OIR_VLINE	Line detection of designated line to panel output at Output Image Render interrupt	Interrupt Controller INTVDCE0OIRVLINE
VDCE1:		
S0_VSYNCERR	Missing Vsync signal for scaler 0	Interrupt Controller INTVDCE1ERR logical OR combination of all VDCE1 error interrupts
IV1_VBUFERR	Frame buffer write overflow signal for scaler 0	
IV3_VBUFERR	Frame buffer read underflow signal for graphics 0	
IV5_VBUFERR	Frame buffer read underflow signal for graphics 2	
IV6_VBUFERR	Frame buffer read underflow signal for graphics 3	
S1_VSYNCERR	Missing Vsync signal for scaler 1	
IV4_VBUFERR	Frame buffer read underflow signal for graphics 1	
	VBLANK detection at Graphics 3	Interrupt Controller INTVDCE1GR3VBLANK
S0_VI_VSYNC	VSYNC input at Scaler 0 interrupt	Interrupt Controller INTVDCE1S0VIVSYNC
S0_LO_VSYNC	VSYNC output at Scaler 0 interrupt	Interrupt Controller INTVDCE1S0LOVSYNC
GR3_VLINE	Line detection of designated line at Graphics3 interrupt	Interrupt Controller INTVDCE1GR3VLINE
S0_VFIELD	End of field for record function at Scaler 0 interrupt	Interrupt Controller INTVDCE1S0VFIELD
S1_LO_VSYNC	VSYNC output at Scaler 1 interrupt	Interrupt Controller INTVDCE1S1LOVSYNC

38.1.5 Reset sources

The Video Data Controllers E and their registers are initialized by the following reset signal:

Table 38.5 Reset sources

VDCEn unit	Reset signal
VDCE0	• Reset Controller SYSRES • Reset Controller VDCE0RES • reset upon wake-up from DEEPSTOP mode
VDCE1	• Reset Controller SYSRES • Reset Controller VDCE1RES • reset upon wake-up from DEEPSTOP mode

CAUTION

By default the VDCEnRES resets are active.
Thus before accessing a VDCEn module this reset must be released via the MRSTC register of the Reset Controller. Refer to Section 9, Reset Controller for details.

38.1.6 I/O signals

The following table shows the I/O signals of the Video Data Controllers E.

Table 38.6 I/O signals connections

VDCEn signal	Function	Connected to
VDCE0		
DV_DATA[23:0]	Video Input I/F 0 *1	Video input selection DV_DATA[23:0]
DV_CLK		Video input selection DVCLK
DV_HSYNC		Video input selection DV_HSYNC
DV_VSYNC		Video input selection DV_VSYNC
RIN[7:0]	Video Output I/F 0 data *2	Video output selection RIN[7:0]
GIN[7:0]		Video output selection GIN[7:0]
BIN[7:0]		Video output selection BIN[7:0]
LCD_CLK	Video Output I/F 0 clock *2	Video output selection LCD_CLK
LCD_TCON0	Video Output I/F 0 synchronization signals	Port VDCE0_VO_TCON0 *2
LCD_TCON1		Port VDCE0_VO_TCON1*3
LCD_TCON2		Port VDCE0_VO_TCON2 *2
LCD_TCON3		Port VDCE0_VO_TCON3 *2
LCD_TCON4		Port VDCE0_VO_TCON4*3
LCD_TCON5		Port VDCE0_VO_TCON5*3
LCD_TCON6		Port VDCE0_VO_TCON6*3
VDCE1		
DV_DATA[23:0]	Video Input I/F 1 *1	Video input selection DV_DATA[23:0]
DV_CLK		Video input selection DVCLK
DV_HSYNC		Video input selection DV_HSYNC
DV_VSYNC		Video input selection DV_VSYNC
RIN[7:0]	Video Output I/F 1 data *2	Video output selection RIN[7:0]
GIN[7:0]		Video output selection GIN[7:0]
BIN[7:0]		Video output selection BIN[7:0]
LCD_CLK	Video Output I/F 1 clock *2	Video output selection LCD_CLK
LCD_TCON0	Video Output I/F 1 synchronization signals	Port VDCE1_VO_TCON0 *2
LCD_TCON1		Port VDCE1_VO_TCON1
LCD_TCON2		Port VDCE1_VO_TCON2 *2
LCD_TCON3		Port VDCE1_VO_TCON3 *2
LCD_TCON4		Port VDCE1_VO_TCON4
LCD_TCON5		Port VDCE1_VO_TCON5
LCD_TCON6		Port VDCE1_VO_TCON6

Note 1. The video input signals can be selected from various ports. Refer to Section 37.7, Video Input selection for details about the video input signal selections.

Note 2. The output of the video output signals to the ports can be output to the display in various modes. Refer to Section 37.9, Video Output selection and RSDS control for details about the video output modes.

Note 3. For availability of these signals, refer to the Section 2.3.2, List of Alternative Function Pins.

38.1.7 Bus master IDs

The Video Data Controller master interfaces are connected to the XC2 cross-connect system, which forms an own master ID domain.

The following table summarizes the Video Data Controller master interfaces IDs, their connection to an XC2 layer and the Sprite Engine slave interface.

Since all Video Data Controller masters access the XC0 and XC1 cross-connects through the Sprite Engine master interfaces, the master ID of the respective Sprite Engine master interface appears on the XC0/XC1 cross-connects.

For detailed information about the cross-connection systems and the usage of the master IDs refer to Section 14.2, Cross-connect systems.

Table 38.7 Video channels master IDs

			XC2 cross-connect		Sprite Engine	
Master I/F		**Access**	**Master ID**	**XC2 layer**	**XC2 slave**	**XC0/XC1 master ID**
Video channel 0	Video Input 0	W	0	XC2_0	RLE Units	MSTID8
	Image Synthesizer 00	R	0	XC2_0	RLE Units	MSTID8
	Image Synthesizer 01	R	4	XC2_1	Sprite Unit 0	MSTID9
	Image Synthesizer 02	R	8	XC2_2	Sprite Unit 1	MSTID10
	Image Synthesizer 03	R	12	XC2_3	Sprite Unit 2	MSTID11
	Output Image Generator	R	10	XC2_1	Sprite Unit 0	MSTID9
	Video Output Warping Engine	W	6	XC2_2	Sprite Unit 1	MSTID10
		R				
Video channel 1	Video Input 1	W	13	XC2_3	Sprite Unit 2	MSTID11
	Image Synthesizer 10	R	1	XC2_0	RLE Units	MSTID8
	Image Synthesizer 11	R	13	XC2_3	Sprite Unit 2	MSTID11
	Image Synthesizer 12	R	9	XC2_2	Sprite Unit 1	MSTID10
	Image Synthesizer 13	R	5	XC2_1	Sprite Unit 0	MSTID9

38.2 Overview

38.2.1 Features

The Video Display Controller consists of the following seven blocks. For the image synthesis, two planes of video image + two graphics planes, one plane of video image + three graphics planes, or four graphics planes can be selected.

1. Input controller: Input video image selection, sync signal adjustment, and brightness adjustment, gain adjustment, and YCbCr ↔ GBR conversion using a color matrix
2. Scaler: Scale up and scale down
3. Image quality improver: YCbCr ↔ GBR conversion using a color matrix
4. Image synthesizer: Synthesis of two planes of video image + two graphics planes, one plane of video image + three graphics planes, or four graphics planes
5. Output image generator: Writing and reading of image data to and from the frame buffer after the image synthesis
6. Output controller: Brightness/contrast adjustment, gamma correction, dither processing, output format conversion, control signal output for TFT-LCD panel
7. System controller: Interrupt control, panel clock control, CLUT table select signal status flag output

The functions of Video Display Controller are listed in the following table.

Table 38.8 Features of Video Display Controller (1/2)

Item	Function
Input video image specification*3	• 8-bit input conforming to ITU-R BT.656 standard (27 MHz, interlace signal) • 8-bit input conforming to ITU-R BT.656 extended standard (27 MHz, progressive signal)*1 • 8-bit input conforming to ITU-R BT.601 extended standard (27 MHz, interlace signal)*1 • 16-bit input conforming to ITU-R BT.601 extended standard (13.5 MHz, interlace signal)*1 • Digital pin input: YCbCr422, YCbCr444, RGB888, RGB666, and RGB565 video image • Digital pin input size: Maximum input video image size to be set*2: 1024 pixels × 1024 lines (horizontal × vertical) • Examples of input video image size: XGA (1024 × 768), SVGA (800 × 600), WVGA (800 × 480), VGA (640 × 480), WQVGA (480 × 240), QVGA in landscape (320 × 240), QVGA in portrait (240 × 320)
Video image quality adjustment function	Contrast adjustment, brightness adjustment
Video image scaling processing*3	Vertical: ×1/8 to ×8, linear/hold interpolation Horizontal: ×1/8 to ×8, linear/hold interpolation IP conversion can be performed by adjusting the initial phase.
Video image rotation function	• Horizontal mirroring in the YCbCr422/YCbCr444/RGB565/RGB888 format

Table 38.8 Features of Video Display Controller (2/2)

Item	Function
Graphics	• Number of graphic planes: Four planes (graphics 0, graphics 1, graphics 2, and graphics 3) • Supported pixel formats: – RGB565 progressive format (α: none, R: 5 bits, G: 6 bits, B: 5 bits; 16 bits in total) – RGB888 progressive format (α: none, R: 8 bits, G: 8 bits, B: 8 bits; 24 bits in total) – α RGB1555 progressive format (α: 1 bit, R: 5 bits, G: 5 bits, B: 5 bits; 16 bits in total) – α RGB4444 progressive format (α: 4 bits, R: 4 bits, G: 4 bits, B: 4 bits; 16 bits in total) – α RGB8888 progressive format (α: 8 bits, R: 8 bits, G: 8 bits, B: 8 bits; 32 bits in total) – RGBα5551 progressive format (R: 5 bits, G: 5 bits, B: 5 bits, α: 1 bit; 16 bits in total) – RGBα8888 progressive format (R: 8 bits, G: 8 bits, B: 8 bits, α: 8 bits; 32 bits in total) – CLUT8 progressive format (CLUT: 8 bits) – CLUT4 progressive format (CLUT: 4 bits) – CLUT1 progressive format (CLUT: 1 bits) – YCbCr422 progressive format (Y: 8 bits, Cb/Cr: 8 bits; 16 bits in total) (only for graphics 0 and 1) – YCbCr444 progressive format (Y: 8 bits, Cb: 8 bits, Cr: 8 bits; 24 bits in total) (only for graphics 0 and 1)
Graphics function	• Alpha blending in rectangular area: Mixes images according to transparency rate α in the specified area. • Chroma-key: Mixes images using the specified RGB color and CLUT value according to transparency rate α. • Alpha blending in one pixel units: Mixes images according to transparency rate α when the target graphics image is in the αRGB1555, αRGB4444, αRGB8888, RGBα5551, RGBα8888, or CLUT8/4/1 format. Alpha blending rectangular and one pixel unit at the same time. Constant rectangular alpha value is multiplied with pixel alpha value.
	For each dot, the priority among the α values of the above functions is as follows: Alpha blending in rectangular area > Chroma-key > Alpha blending in one pixel units
	Note: Alpha blending and Chroma-Key features are not supported for Graphic Layer 0 (GR0).
Output video image size	Maximum output video image size to be set*: 1280 pixels × 1024 lines (horizontal × vertical) Note: * Depends on the AC characteristics of the display panel. Examples of output video image size: • WXGA(1280 x 768), XGA (1024 × 768) • SVGA (800 × 600), WVGA (800 × 480), • VGA (640 × 480), WQVGA (480 × 240), • QVGA size in landscape (320 × 240) • QVGA size in portrait (240 × 320)
Output video image format	• RGB888 progressive video output (24-bit parallel output) • RGB666 progressive video output (18-bit parallel output) • RGB565 progressive video output (16-bit parallel output) • Serial RGB progressive output (8-bit serial output) (D1M1A and D1M1-V2 only)
Panel output adjustment	Panel brightness/contrast adjustment, RGB gamma correction*3, dither processing, output format conversion
Sync signal output	Control signal output for the TFT-LCD panel
Interrupt output	• Vsync signal for video image input/output • Line interrupt output (can be output on a desired line.) • Erroneous Vsync cycle detection signal for video input • Field write completion signal • Overflow/underflow signal for the internal buffer

Note 1. The ITU-R BT.656 and BT.601 standard does not include the description regarding the progressive signal. The ITU-R BT.601 standard does not include the description regarding the connection interface.

Note 2. Depends on the AC characteristics of the connected device.

Note 3. Not available for D1L2(H) devices.

38.2.2 Block Diagram

Following figures show the entire block diagram of this module. This LSI incorporates two video display controllers (channels 0 and 1), to each of which the video signal is input from the corresponding video decoder. Each controller can also fetch output from the input control block of the other controller. For details, see the description of each block.

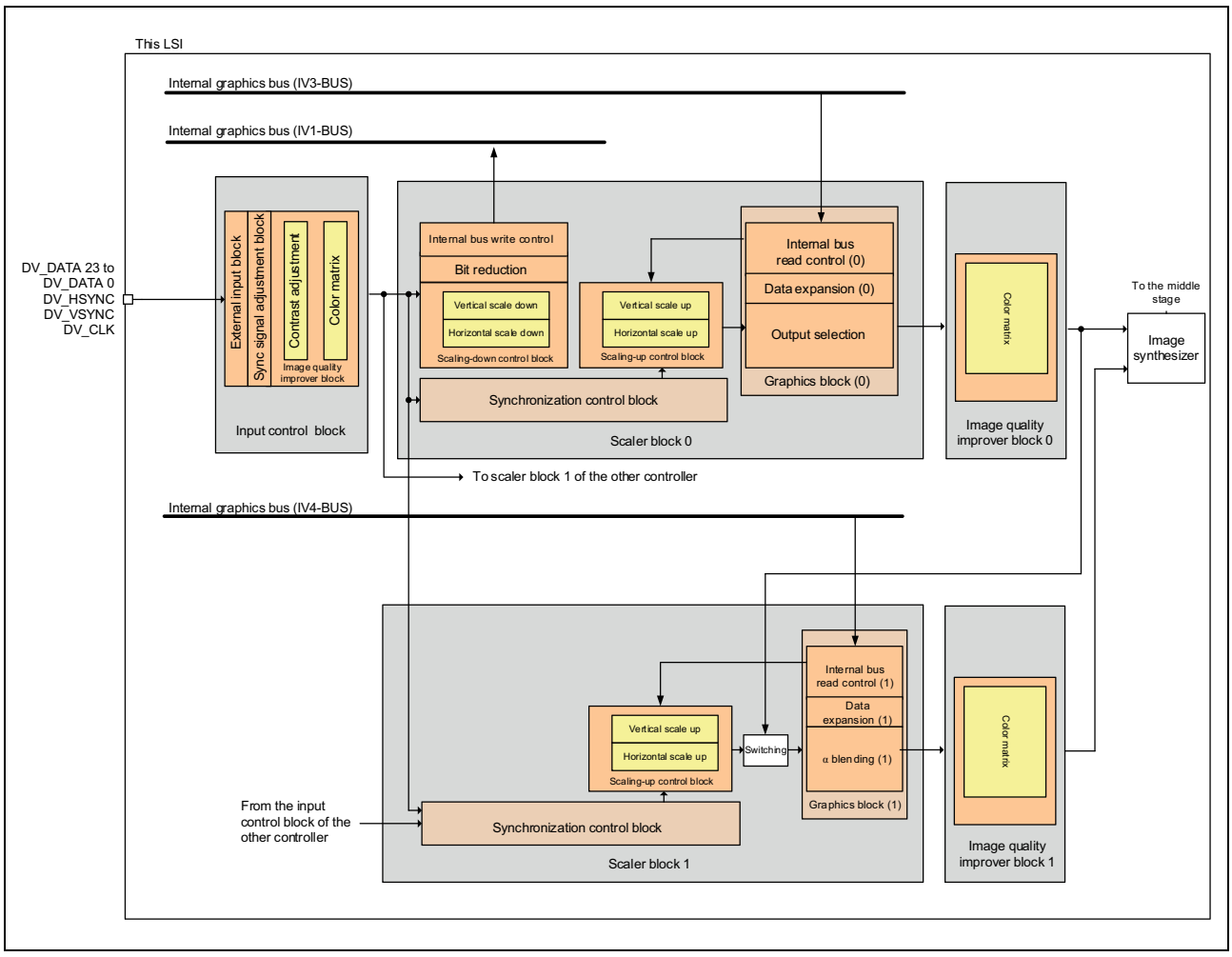
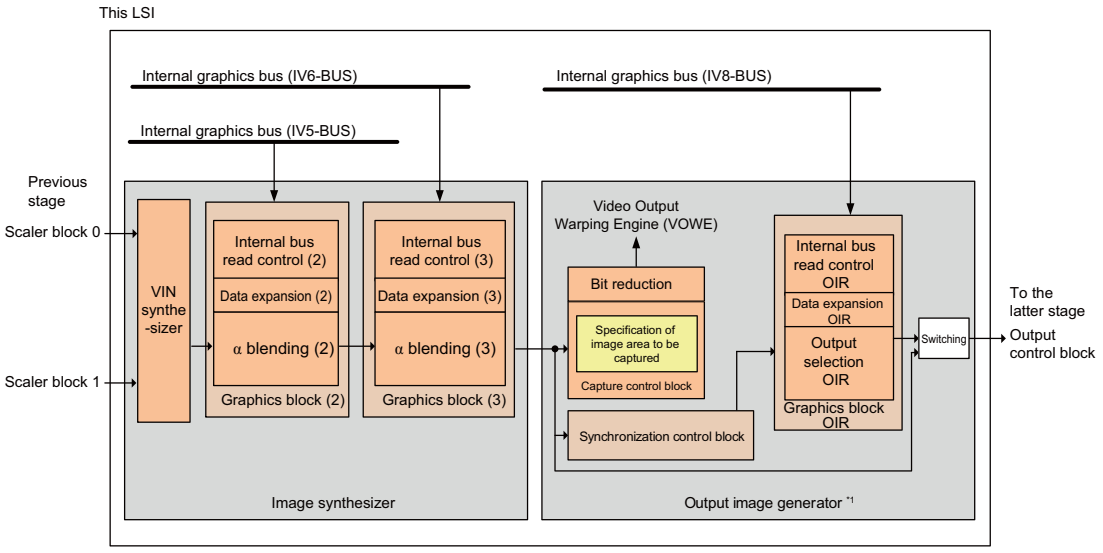


Figure 38.1 Video Display Controller Former Stage Block Diagram



Note 1. The output image generator is bypassed in the D1L2(H) devices.

Figure 38.2 Video Display Controller Middle Stage Block Diagram

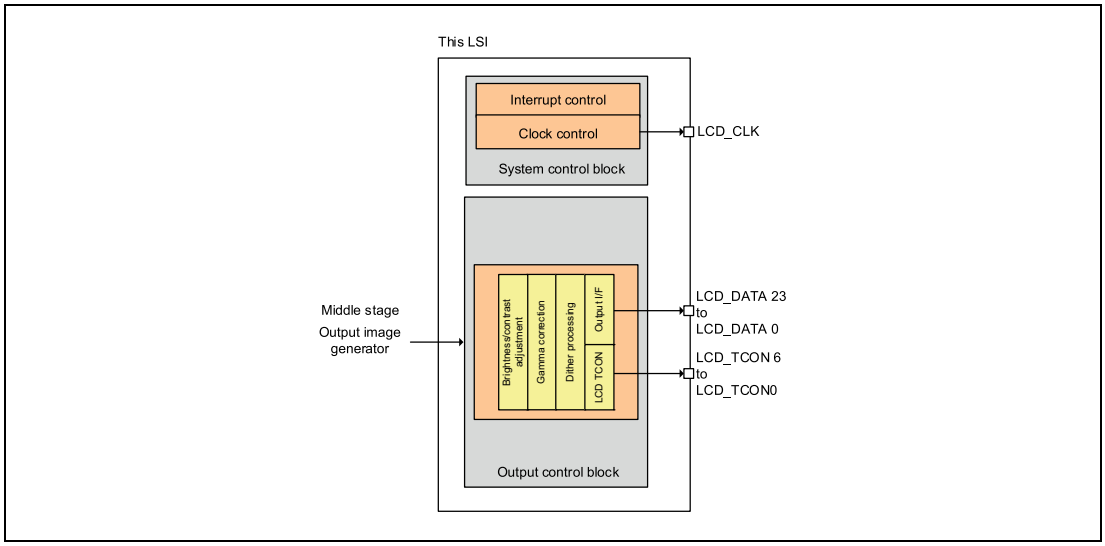


Figure 38.3 Video Display Controller Latter Stage Block Diagram

38.2.3 Input/Output Pins

The following tables show the pin configuration.

Table 38.9 Input/Output Pins (Channel 0)

Symbol	I/O	Pin Name	Function
DV0_CLK	Input	External input clock 0	External input 0 clock pin
DV0_VSYNC	Input	External input Vsync 0	External input 0 Vsync signal pin
DV0_HSYNC	Input	External input Hsync 0	External input 0 Hsync signal pin
DV0_DATA 17 to DV0_DATA 0	Input	External input video image data 0	External input 0 video image data pin
LCD0_CLK	Output	Panel clock 0	Panel output 0 clock pin
LCD0_DATA 23 to LCD0_DATA 0	Output	Video image data 0 for panel	Panel output 0 video image data pin
LCD0_TCON 6 to LCD0_TCON 0	Output	Control signal 0 for panel	Panel output 0 timing control pin

Table 38.10 Input/Output Pins (Channel 1)

Symbol	I/O	Pin Name	Function
DV1_CLK	Input	External input clock 1	External input 1 clock pin
DV1_VSYNC	Input	External input Vsync 1	External input 1 Vsync signal pin
DV1_HSYNC	Input	External input Hsync 1	External input 1 Hsync signal pin
DV1_DATA 23 to DV1_DATA 0	Input	External input video image data 1	External input 1 video image data pin
LCD1_CLK	Output	Panel clock 1	Panel output 1 clock pin
LCD1_DATA 23 to LCD1_DATA 0	Output	Video image data 1 for panel	Panel output 1 video image data pin
LCD1_TCON 6 to LCD1_TCON 0	Output	Control signal 1 for panel	Panel output 1 timing control pin

38.2.4 Clocks

There are two clocks to be mainly used by the Video Display Controller: the video image clock and pixel clock.

The video image clock is used while the video image is processed in the input controller, passed to the scale-down control block in the scaler, and then written to the buffer (internal bus write control).

The pixel clock is used in graphics read-out processing by the scaler (internal bus read controller) through output controller processing. For parallel RGB output, the frequencies of the pixel clock and panel clock (LCD_CLK) are the same.

When the serial RGB output is selected in the output controller, the pixel clock frequency is 1/3 or 1/4 of the panel clock (LCD_CLK) frequency.

38.2.5 Hsync and Vsync Signals

Hsync and Vsync signals to be used in the logic stage following the scale-up control block of the scaler are generated by the synchronization control block of the scaler. Since the Hsync and Vsync signals are used as the reference signals for the LCD TCON, which generates various LCD panel driving timings, they are also the reference signals for the control signals (LCD_TCON6 to LCD_TCON0 pins) passed to the LCD panel.

The output Hsync signal always operates at a free-running frequency, and the horizontal period is set with SC_SCL0_FRC4.SC_RES_FH[10:0]. On the other hand, the output Vsync signal is selected from the external input (digital pin input) and free-running Vsync signals with SC_SCL0_FRC3.SC_RES_VS_SEL of the scaler.

38.2.5.1 External Input Vsync

(1) Operation Outline

In this mode, the output Vsync signal is generated according to an external input Vsync signal. When displaying video image input from a digital pin on the panel, always use this mode. However, the output Hsync signal is free running even in this mode. The following figure shows the timing of external input Vsync signal.

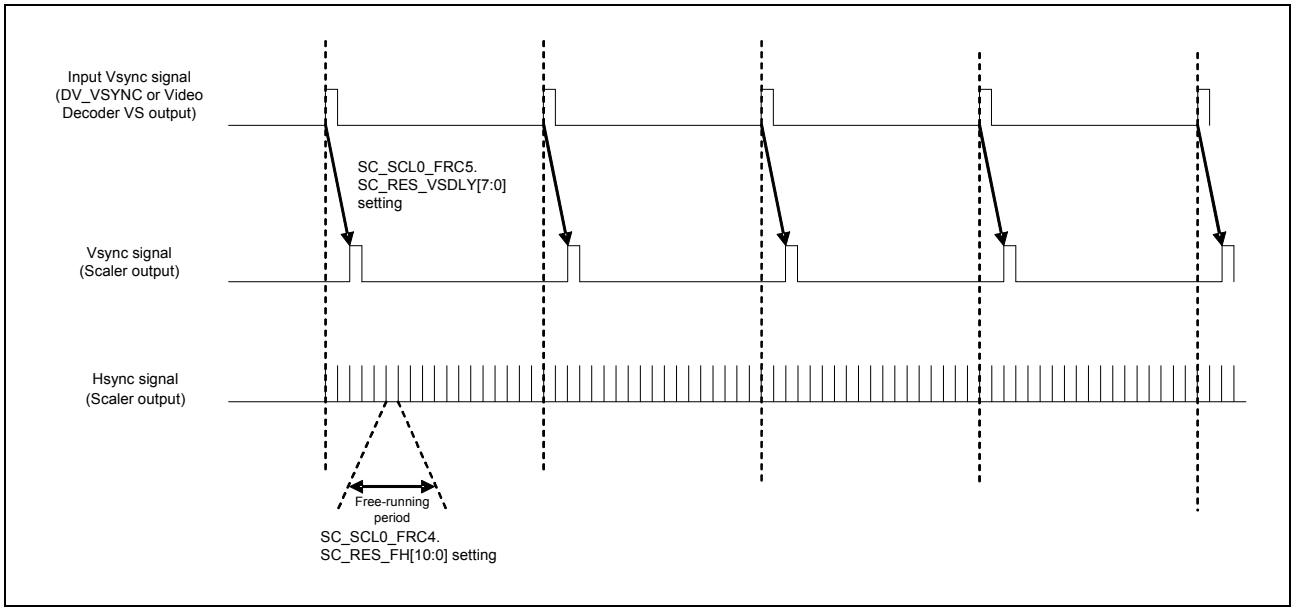


Figure 38.4 External Input Vsync Timing

(2) Notes

When Vsync is externally input, generation of the output Vsync signal is based on the external Vsync signal. That is, the output Vsync signal follows the input Vsync signal, so if an unstable Vsync signal is input, the output Vsync signal will also be unstable.

Since the output Hsync signal is generated according to the frequency generated by a free-running clock and the Vsync signal is generated from the video input as a base, the signals will not be in synchronization. This module adjusts the timing between these signals by adjusting the output Vsync signal so that it stays in time with the output Hsync signal. Therefore, even if the input Vsync signal is stable, the timing of the output Vsync signal may be increased or decreased by up to one line to stay in synchronization with the timing of the output Hsync signal.

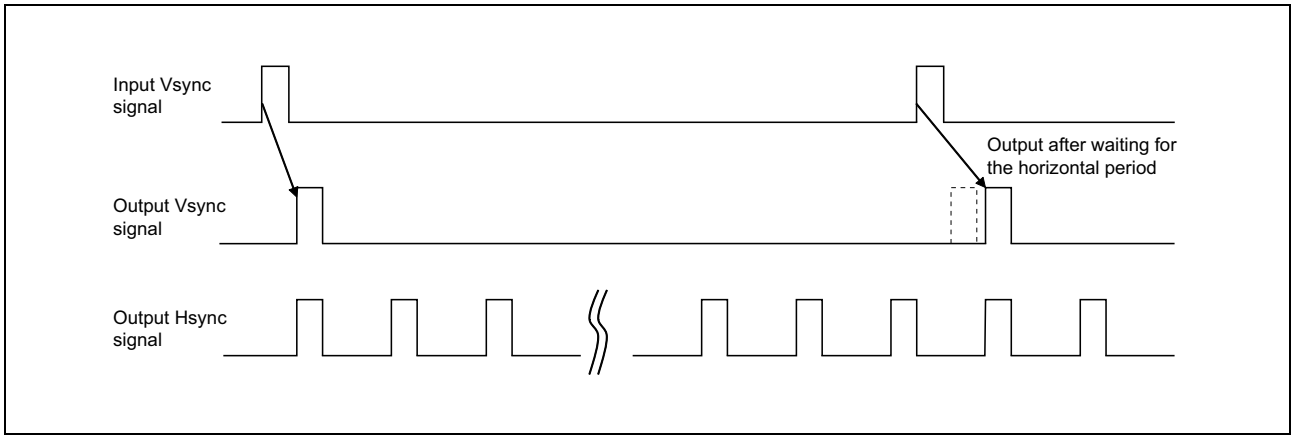


Figure 38.5 Detailed Timing Chart for Generation of the Output Vsync Signal

38.2.5.2 Free-Running Vsync

In this mode, the Vsync signal is generated according to the pixel clock (free running). The vertical period is selected with SC_SCL0_FRC4.SC_RES_FV[10:0]. The output Hsync signal is also free running. The following figure shows the timing.

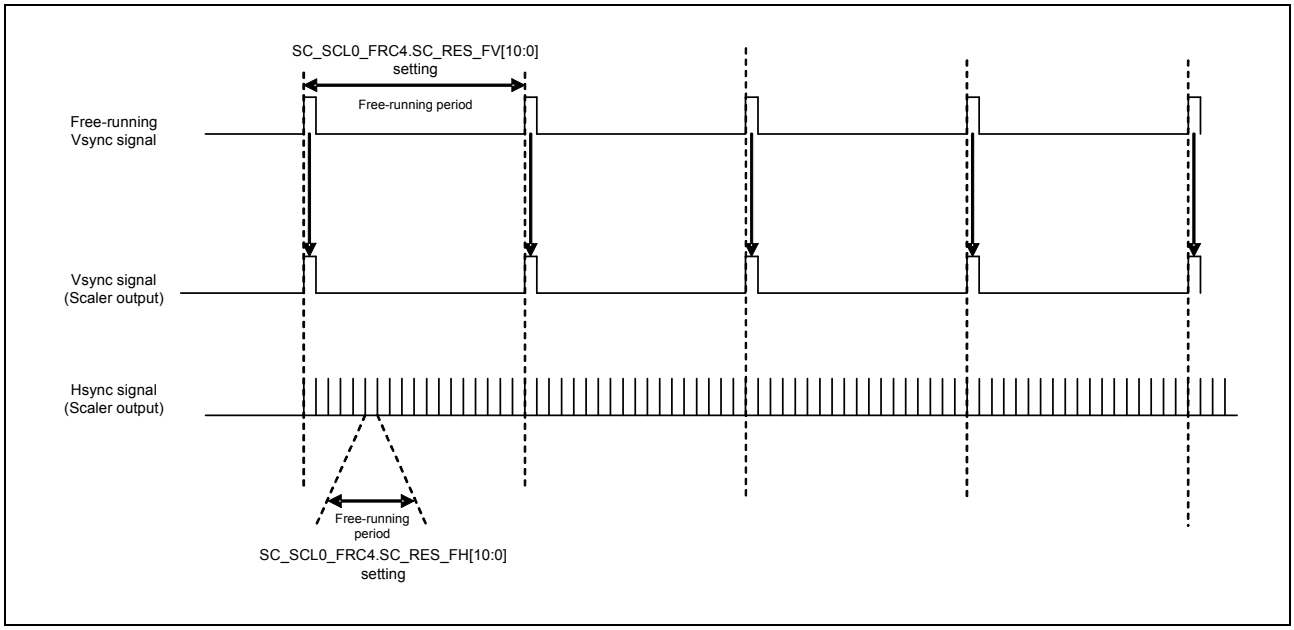


Figure 38.6 Free Running Vsync Timing

38.2.5.3 Sync Signal Selection

The following table shows examples of sync signal selection.

Table 38.11 Sync Signal Selection

External Video Input	Graphics	Vsync Signal Selected
Not displayed	Not displayed	Not used
Not displayed	Displayed	Free-running Vsync
Displayed	Not displayed	External input Vsync
Displayed	Displayed	External input Vsync

38.2.5.4 Usage Note on Changing Vsync Signal Selections

When the Vsync signal selection is changed, the output Vsync signal is discontinuous, resulting in disordered panel display. In this case, perform the mute processing according to the panel specification as necessary and change the Vsync signal selection.

38.3 Input Controller

NOTE

The Input Controller is not available for the D1L2(H) devices.

38.3.1 Input Controller Functions

38.3.1.1 Overview of Functions

The input controller receives signals supplied via the external input pins, and subjects the signals to synchronization adjustment, horizontal noise reduction, contrast correction (dynamic range compression), and brightness adjustment, gain adjustment, and YCbCr ↔ GBR conversion using a color matrix.

The functional block diagram of the input controller is shown below.

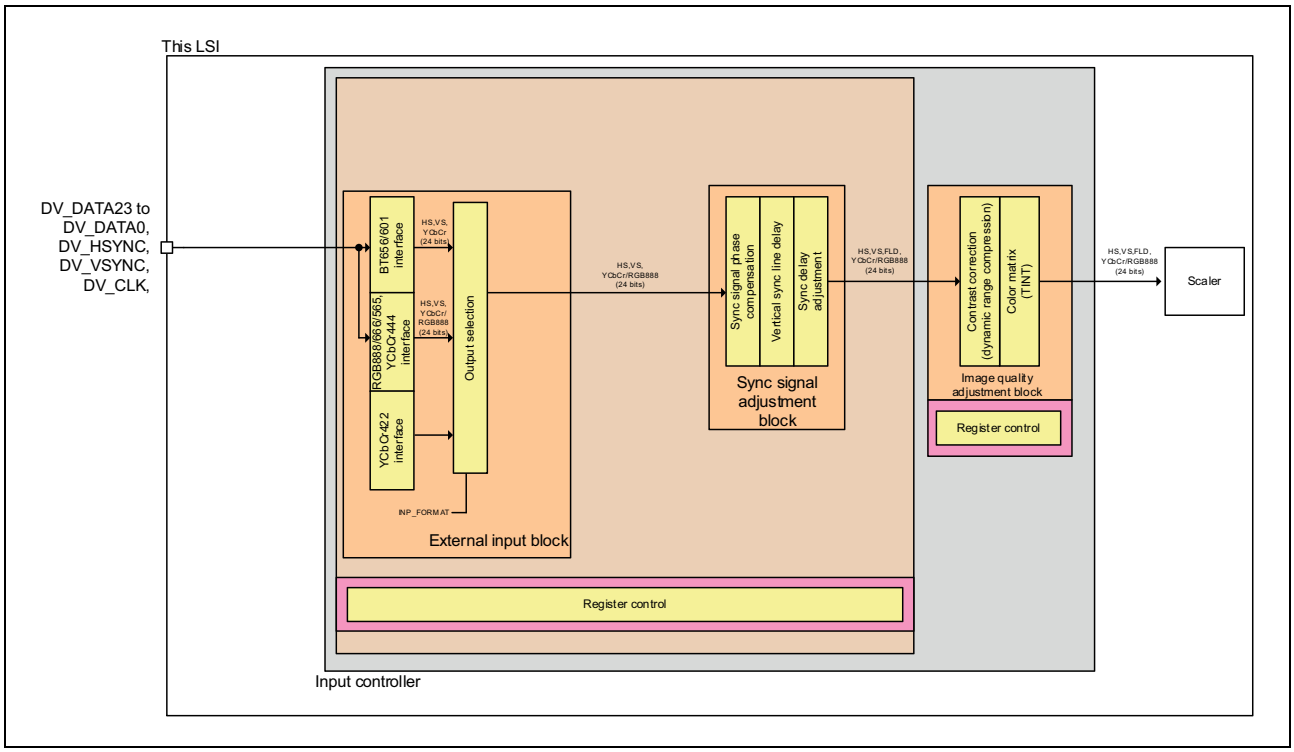


Figure 38.7 Functional Block Diagram of Input Controller

38.3.1.2 Updating Registers of External Signal Input Block and Sync Signal Adjustment Block

The control registers of the external input block and sync signal adjustment block are updated by setting the relevant update control bit to 1.

For the control registers other than the IMGCNT_DRC_REG register of the image quality adjustment block, the update timing is controlled using the Vsync signal.

After 1 is set to the bits in the update control register, the contents of the relevant registers are actually modified at the rising edge of the Vsync signal, when the update control register is automatically cleared to 0.

Table 38.12 Register Update Control

Register Name	Bit Name	Initial Value	Description
INP_UPDATE	INP_EXT_UPDATE	0	External Input Block Register Update 0: Registers are not updated. 1: Registers are updated.
INP_UPDATE	INP_IMG_UPDATE	0	Sync Signal Adjustment Block Register Update 0: Registers are not updated. 1: Registers are updated.
IMGCNT_UPDATE	IMGCNT_VEN	0	Image Quality Adjustment Block Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync signal.

38.3.1.3 Controlling External Input Video Signals

(1) The external input video image signals in the YCbCr444, RGB888, RGB666, RGB565, BT656 (extended), BT601 (extended), YCbCr422 (BT601 extended to 16 bits) formats can be handled.

(2) The BT656 signals can be used for the 525-line and 59.94-Hz (27.0-MHz) or the 625-line and 50.00-Hz (27.0-MHz) interlace signals and for the 525-line and 59.94-Hz (54.0-MHz) or the 625-line and 50.00-Hz (54.0-MHz) BT656-extended progressive signals.

(3) The BT601 signals can be used for the 8-bit data line 525-line and 59.94 Hz (27.0-MHz) or the 625-line and 50.00-Hz (27.0-MHz) interlace signals and for the 525-line and 59.94 Hz (54.0-MHz) or the 625-line and 50.00-Hz (54.0-MHz) progressive signals.

(4) The YCbCr422 signals can be used for the 16-bit data line 525-line and 59.94-Hz (13.5-MHz) or the 625-line and 50.00-Hz (13.5-MHz) BT601-extended interlace signals.

(5) The above signals can be selected by the INP_FORMAT[2:0] bits. Bit endian change and B/R signal swap are controlled by setting the INP_ENDIAN_ON and INP_SWAP_ON bits.

Table 38.13 External Input Video Signal Control

Register Name	Bit Name	Initial Value	Description
INP_SEL_CNT	INP_FORMAT[2:0]	000	External Input Format Select 0: YCbCr444, RGB888 1: RGB666 2: RGB565 3: BT656 4: BT601 5: YCbCr422 6, 7: Setting prohibited
INP_EXT_SYNC_CNT	INP_ENDIAN_ON	0	External Input Bit Endian Change On/Off Control 0: Off 1: On
INP_EXT_SYNC_CNT	INP_SWAP_ON	0	External Input B/R Signal Swap On/Off Control 0: Off 1: On

38.3.1.4 Selecting Clock Edge for External Input Signals

The clock edge for receiving the video image signals, Vsync signals, and Hsync signals is individually selected with the INP_PXD_EDGE, INP_VS_EDGE, INP_HS_EDGE bits.

Table 38.14 External Input Clock Edge Selection

Register Name	Bit Name	Initial Value	Description
INP_SEL_CNT	INP_PXD_EDGE	0	Clock Edge Select for Capturing External Input Video Image Signals DV_DATA23 to DV_DATA0 0: Rising edge 1: Falling edge
INP_SEL_CNT	INP_VS_EDGE	0	Clock Edge Select for Capturing External Input Vsync Signal DV_VSYNC 0: Rising edge 1: Falling edge
INP_SEL_CNT	INP_HS_EDGE	0	Clock Edge Select for Capturing External Input Hsync Signal DV_HSYNC 0: Rising edge 1: Falling edge

The figure below shows the typical input timing of external input signals.

The input signals can be received at the rising edge of the clock signal DV_CLK when the INP_PXD_EDGE, INP_VS_EDGE, and INP_HS_EDGE bits are 0.

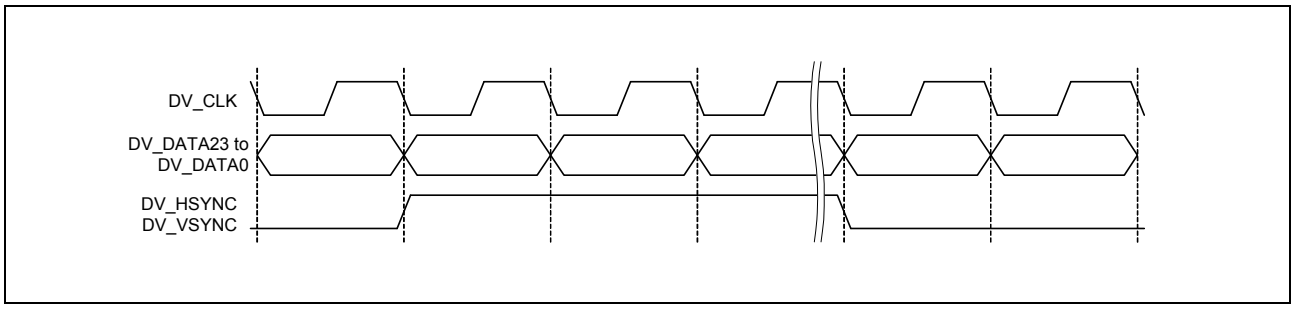


Figure 38.8 Typical Input Timing of external Input Signals (Clock Phase)

38.3.1.5 External Input Sync Signal Inversion Control

Inversion of polarity of the Vsync and Hsync signals can be controlled by the INP_VS_INV and INP_HS_INV bits.

Table 38.15 Sync Signal Inversion Control

Register Name	Bit Name	Initial Value	Description
INP_EXT_ SYNC_CNT	INP_VS_INV	0	External Input Vsync Signal DV_VSYNC Inversion Control 0: Not inverted (positive polarity) 1: Inverted (negative polarity)
INP_EXT_ SYNC_CNT	INP_HS_INV	0	External Input Hsync Signal DV_HSYNC Inversion Control 0: Not inverted (positive polarity) 1: Inverted (negative polarity)

38.3.1.6 Bit Allocation of External Input Video Image Signals

Allocation of the external input video image signal pins DV_DATA to the signals in each format is described below.

(1) YCbCr444/RGB888 Input

When the external input is of YCbCr444/RGB888 format, the video image signal pins DV_DATA are allocated to the internal signals Y/GOUT, Cb/BOUT, Cr/ROUT, as shown in **Table 38.16**.

Table 38.16 Bit Allocation of DV_DATA Pin Inputs when the External Input is of YCbCr444/RGB888

INP_FORMAT[2:0]	**0**	**0**	**0**	**0**
INP_ENDIAN_ON	**0**	**0**	**1**	**1**
INP_SWAP_ON	**0**	**1**	**0**	**1**
DV_DATA23	Cr/ROUT[7]	Cb/BOUT[7]	Cr/ROUT[0]	Cb/BOUT[0]
DV_DATA22	Cr/ROUT[6]	Cb/BOUT[6]	Cr/ROUT[1]	Cb/BOUT[1]
DV_DATA21	Cr/ROUT[5]	Cb/BOUT[5]	Cr/ROUT[2]	Cb/BOUT[2]
DV_DATA20	Cr/ROUT[4]	Cb/BOUT[4]	Cr/ROUT[3]	Cb/BOUT[3]
DV_DATA19	Cr/ROUT[3]	Cb/BOUT[3]	Cr/ROUT[4]	Cb/BOUT[4]
DV_DATA18	Cr/ROUT[2]	Cb/BOUT[2]	Cr/ROUT[5]	Cb/BOUT[5]
DV_DATA17	Cr/ROUT[1]	Cb/BOUT[1]	Cr/ROUT[6]	Cb/BOUT[6]
DV_DATA16	Cr/ROUT[0]	Cb/BOUT[0]	Cr/ROUT[7]	Cb/BOUT[7]
DV_DATA15	Y/GOUT[7]	Y/GOUT[7]	Y/GOUT[0]	Y/GOUT[0]
DV_DATA14	Y/GOUT[6]	Y/GOUT[6]	Y/GOUT[1]	Y/GOUT[1]
DV_DATA13	Y/GOUT[5]	Y/GOUT[5]	Y/GOUT[2]	Y/GOUT[2]
DV_DATA12	Y/GOUT[4]	Y/GOUT[4]	Y/GOUT[3]	Y/GOUT[3]
DV_DATA11	Y/GOUT[3]	Y/GOUT[3]	Y/GOUT[4]	Y/GOUT[4]
DV_DATA10	Y/GOUT[2]	Y/GOUT[2]	Y/GOUT[5]	Y/GOUT[5]
DV_DATA9	Y/GOUT[1]	Y/GOUT[1]	Y/GOUT[6]	Y/GOUT[6]
DV_DATA8	Y/GOUT[0]	Y/GOUT[0]	Y/GOUT[7]	Y/GOUT[7]
DV_DATA7	Cb/BOUT[7]	Cr/ROUT[7]	Cb/BOUT[0]	Cr/ROUT[0]
DV_DATA6	Cb/BOUT[6]	Cr/ROUT[6]	Cb/BOUT[1]	Cr/ROUT[1]
DV_DATA5	Cb/BOUT[5]	Cr/ROUT[5]	Cb/BOUT[2]	Cr/ROUT[2]
DV_DATA4	Cb/BOUT[4]	Cr/ROUT[4]	Cb/BOUT[3]	Cr/ROUT[3]
DV_DATA3	Cb/BOUT[3]	Cr/ROUT[3]	Cb/BOUT[4]	Cr/ROUT[4]
DV_DATA2	Cb/BOUT[2]	Cr/ROUT[2]	Cb/BOUT[5]	Cr/ROUT[5]
DV_DATA1	Cb/BOUT[1]	Cr/ROUT[1]	Cb/BOUT[6]	Cr/ROUT[6]
DV_DATA0	Cb/BOUT[0]	Cr/ROUT[0]	Cb/BOUT[7]	Cr/ROUT[7]

(2) RGB666 Input

When the external input is of RGB666 format, the video image signal pins DV_DATA are allocated to the internal signals GOUT, BOUT, ROUT as shown in **Table 38.17**.

The internal signals GOUT, BOUT, ROUT to which the video image signal pins DV_DATA are allocated are output as a 24-bit video image from the RGB666 interface with the following formulae.

G[7:0] = GOUT[7:2] × 255 ÷ 63

B[7:0] = BOUT[7:2] × 255 ÷ 63

R[7:0] = ROUT[7:2] × 255 ÷ 63

Table 38.17 Bit Allocation of DV_DATA Pin Inputs When the External Input is of RGB666

INP_FORMAT[2:0]	**1**	**1**	**1**	**1**
INP_ENDIAN_ON	**0**	**0**	**1**	**1**
INP_SWAP_ON	**0**	**1**	**0**	**1**
DV_DATA17	ROUT[7]	BOUT[7]	ROUT[2]	BOUT[2]
DV_DATA16	ROUT[6]	BOUT[6]	ROUT[3]	BOUT[3]
DV_DATA15	ROUT[5]	BOUT[5]	ROUT[4]	BOUT[4]
DV_DATA14	ROUT[4]	BOUT[4]	ROUT[5]	BOUT[5]
DV_DATA13	ROUT[3]	BOUT[3]	ROUT[6]	BOUT[6]
DV_DATA12	ROUT[2]	BOUT[2]	ROUT[7]	BOUT[7]
DV_DATA11	GOUT[7]	GOUT[7]	GOUT[2]	GOUT[2]
DV_DATA10	GOUT[6]	GOUT[6]	GOUT[3]	GOUT[3]
DV_DATA9	GOUT[5]	GOUT[5]	GOUT[4]	GOUT[4]
DV_DATA8	GOUT[4]	GOUT[4]	GOUT[5]	GOUT[5]
DV_DATA7	GOUT[3]	GOUT[3]	GOUT[6]	GOUT[6]
DV_DATA6	GOUT[2]	GOUT[2]	GOUT[7]	GOUT[7]
DV_DATA5	BOUT[7]	ROUT[7]	BOUT[2]	ROUT[2]
DV_DATA4	BOUT[6]	ROUT[6]	BOUT[3]	ROUT[3]
DV_DATA3	BOUT[5]	ROUT[5]	BOUT[4]	ROUT[4]
DV_DATA2	BOUT[4]	ROUT[4]	BOUT[5]	ROUT[5]
DV_DATA1	BOUT[3]	ROUT[3]	BOUT[6]	ROUT[6]
DV_DATA0	BOUT[2]	ROUT[2]	BOUT[7]	ROUT[7]

(3) RGB565 Input

When the external input is of RGB565 format, the video image signal pins DV_DATA are allocated to the internal signals GOUT, BOUT, ROUT as shown in **Table 38.18**.

The internal signals GOUT, BOUT, ROUT to which the video image signal pins DV_DATA are allocated are output as a 24-bit video image from the RGB565 interface with the following formulae.

G[7:0] = GOUT[7:2] × 255 ÷ 63

B[7:0] = BOUT[7:3] × 255 ÷ 31

R[7:0] = ROUT[7:3] × 255 ÷ 31

Table 38.18 Bit Allocation of DV_DATA Pin Inputs When the External Input is of RGB565

INP_FORMAT[2:0]	**2**	**2**	**2**	**2**
INP_ENDIAN_ON	**0**	**0**	**1**	**1**
INP_SWAP_ON	**0**	**1**	**0**	**1**
DV_DATA15	ROUT[7]	BOUT[7]	ROUT[3]	BOUT[3]
DV_DATA14	ROUT[6]	BOUT[6]	ROUT[4]	BOUT[4]
DV_DATA13	ROUT[5]	BOUT[5]	ROUT[5]	BOUT[5]
DV_DATA12	ROUT[4]	BOUT[4]	ROUT[6]	BOUT[6]
DV_DATA11	ROUT[3]	BOUT[3]	ROUT[7]	BOUT[7]
DV_DATA10	GOUT[7]	GOUT[7]	GOUT[2]	GOUT[2]
DV_DATA9	GOUT[6]	GOUT[6]	GOUT[3]	GOUT[3]
DV_DATA8	GOUT[5]	GOUT[5]	GOUT[4]	GOUT[4]
DV_DATA7	GOUT[4]	GOUT[4]	GOUT[5]	GOUT[5]
DV_DATA6	GOUT[3]	GOUT[3]	GOUT[6]	GOUT[6]
DV_DATA5	GOUT[2]	GOUT[2]	GOUT[7]	GOUT[7]
DV_DATA4	BOUT[7]	ROUT[7]	BOUT[3]	ROUT[3]
DV_DATA3	BOUT[6]	ROUT[6]	BOUT[4]	ROUT[4]
DV_DATA2	BOUT[5]	ROUT[5]	BOUT[5]	ROUT[5]
DV_DATA1	BOUT[4]	ROUT[4]	BOUT[6]	ROUT[6]
DV_DATA0	BOUT[3]	ROUT[3]	BOUT[7]	ROUT[7]

(4) BT656/BT601 Input

When the external input is of BT656 or BT601 format, the video image signal pins DV_DATA are allocated to the internal signal BTOUT, as shown in **Table 38.19**.

The internal signal BTOUT to which the video image signal pins DV_DATA are allocated is expanded to the YCbCr signal.

For expansion to the YCbCr signal, see Section 38.3.1.11, BT656/BT601/YCbCr422 Format Setting.

Table 38.19 Bit Allocation of DV_DATA Pin Inputs When the External Input is of BT656 or BT601

INP_FORMAT[2:0]	3 to 4	3 to 4
INP_ENDIAN_ON	0	1
INP_SWAP_ON	0	0
DV_DATA7	BTOUT[7]	BTOUT[0]
DV_DATA6	BTOUT[6]	BTOUT[1]
DV_DATA5	BTOUT[5]	BTOUT[2]
DV_DATA4	BTOUT[4]	BTOUT[3]
DV_DATA3	BTOUT[3]	BTOUT[4]
DV_DATA2	BTOUT[2]	BTOUT[5]
DV_DATA1	BTOUT[1]	BTOUT[6]
DV_DATA0	BTOUT[0]	BTOUT[7]

(5) YCbCr422 Input

When the external input is of YCbCr422 format, the video image signal pins DV_DATA are allocated to the internal signals Y and Cb/Cr, as shown in **Table 38.20**.

Table 38.20 Bit Allocation of DV_DATA Pin Inputs When the External Input is of YCbCr422

INP_FORMAT[2:0]	5	5	5	5
INP_ENDIAN_ON	0	0	1	1
INP_SWAP_ON	0	1	0	1
DV_DATA15	Y[7]	Cb/Cr[7]	Y[0]	Cb/Cr[0]
DV_DATA14	Y[6]	Cb/Cr[6]	Y[1]	Cb/Cr[1]
DV_DATA13	Y[5]	Cb/Cr[5]	Y[2]	Cb/Cr[2]
DV_DATA12	Y[4]	Cb/Cr[4]	Y[3]	Cb/Cr[3]
DV_DATA11	Y[3]	Cb/Cr[3]	Y[4]	Cb/Cr[4]
DV_DATA10	Y[2]	Cb/Cr[2]	Y[5]	Cb/Cr[5]
DV_DATA9	Y[1]	Cb/Cr[1]	Y[6]	Cb/Cr[6]
DV_DATA8	Y[0]	Cb/Cr[0]	Y[7]	Cb/Cr[7]
DV_DATA7	Cb/Cr[7]	Y[7]	Cb/Cr[0]	Y[0]
DV_DATA6	Cb/Cr[6]	Y[6]	Cb/Cr[1]	Y[1]
DV_DATA5	Cb/Cr[5]	Y[5]	Cb/Cr[2]	Y[2]
DV_DATA4	Cb/Cr[4]	Y[4]	Cb/Cr[3]	Y[3]
DV_DATA3	Cb/Cr[3]	Y[3]	Cb/Cr[4]	Y[4]
DV_DATA2	Cb/Cr[2]	Y[2]	Cb/Cr[5]	Y[5]
DV_DATA1	Cb/Cr[1]	Y[1]	Cb/Cr[6]	Y[6]
DV_DATA0	Cb/Cr[0]	Y[0]	Cb/Cr[7]	Y[7]

38.3.1.7 Typical Signal Timing of BT601 Format

Figure 38.9 and **Figure 38.10** show the horizontal timings and **Figure 38.11** and **Figure 38.12** show the vertical timings of the BT601 format.

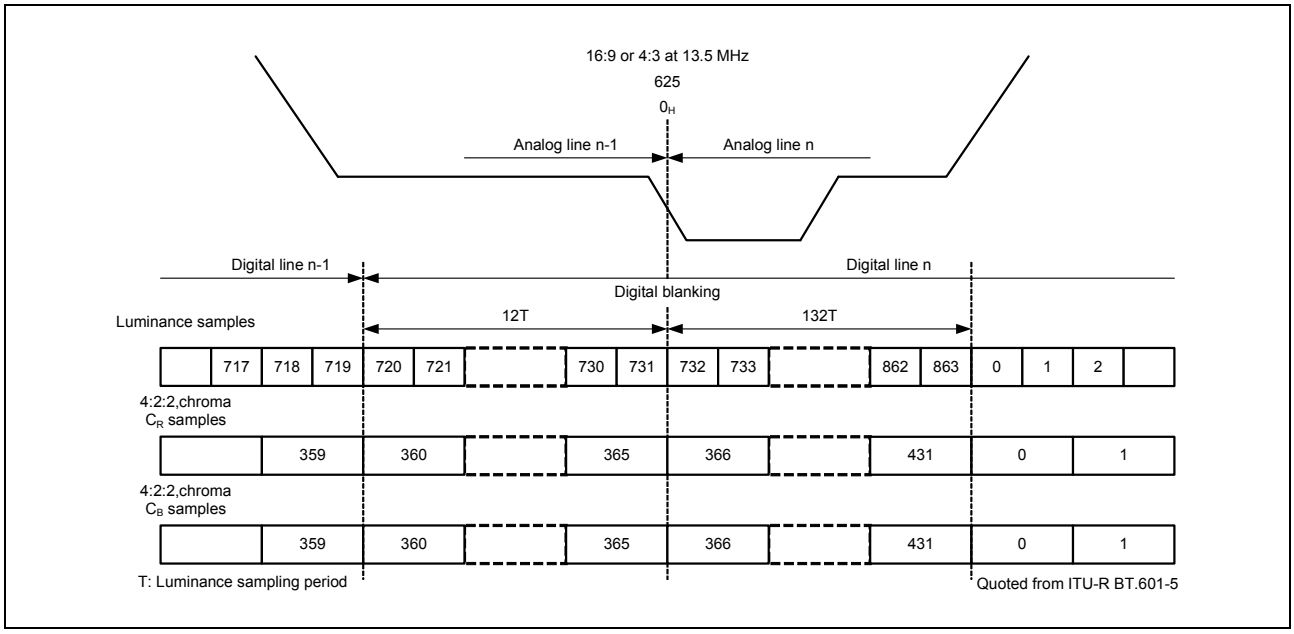


Figure 38.9 BT601 Horizontal Timing (625 Lines/50.00 Hz)

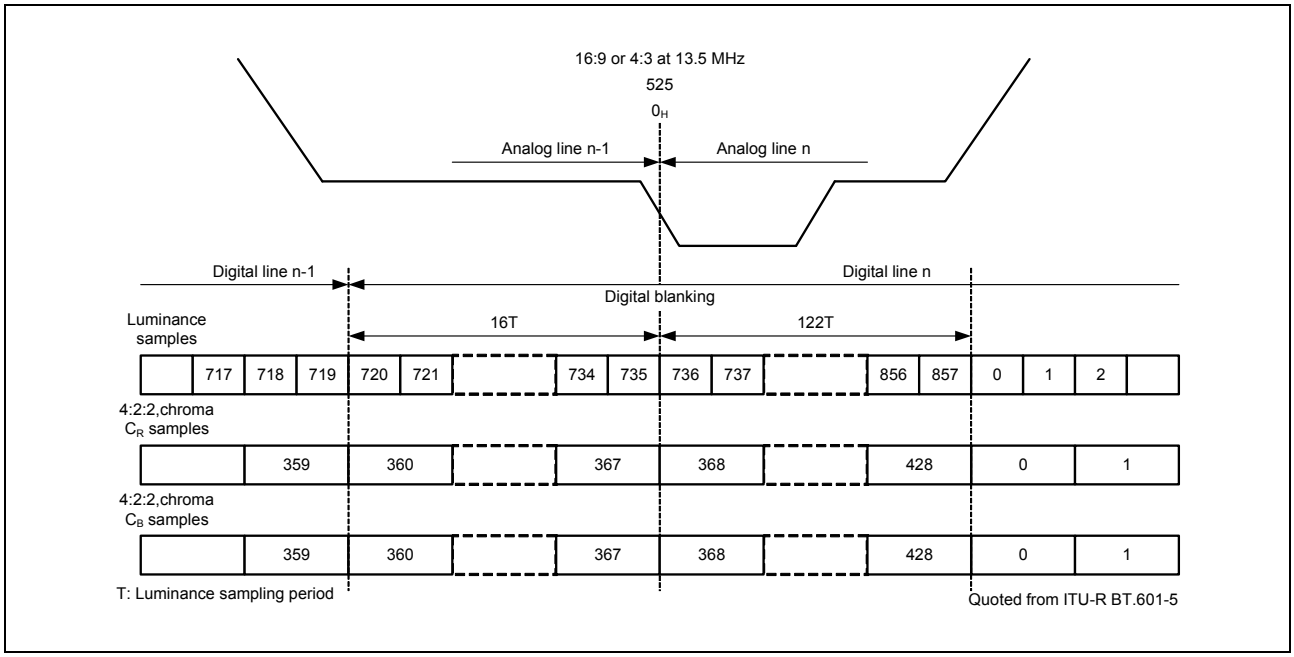


Figure 38.10 BT601 Horizontal Timing (525 Lines/59.94 Hz)

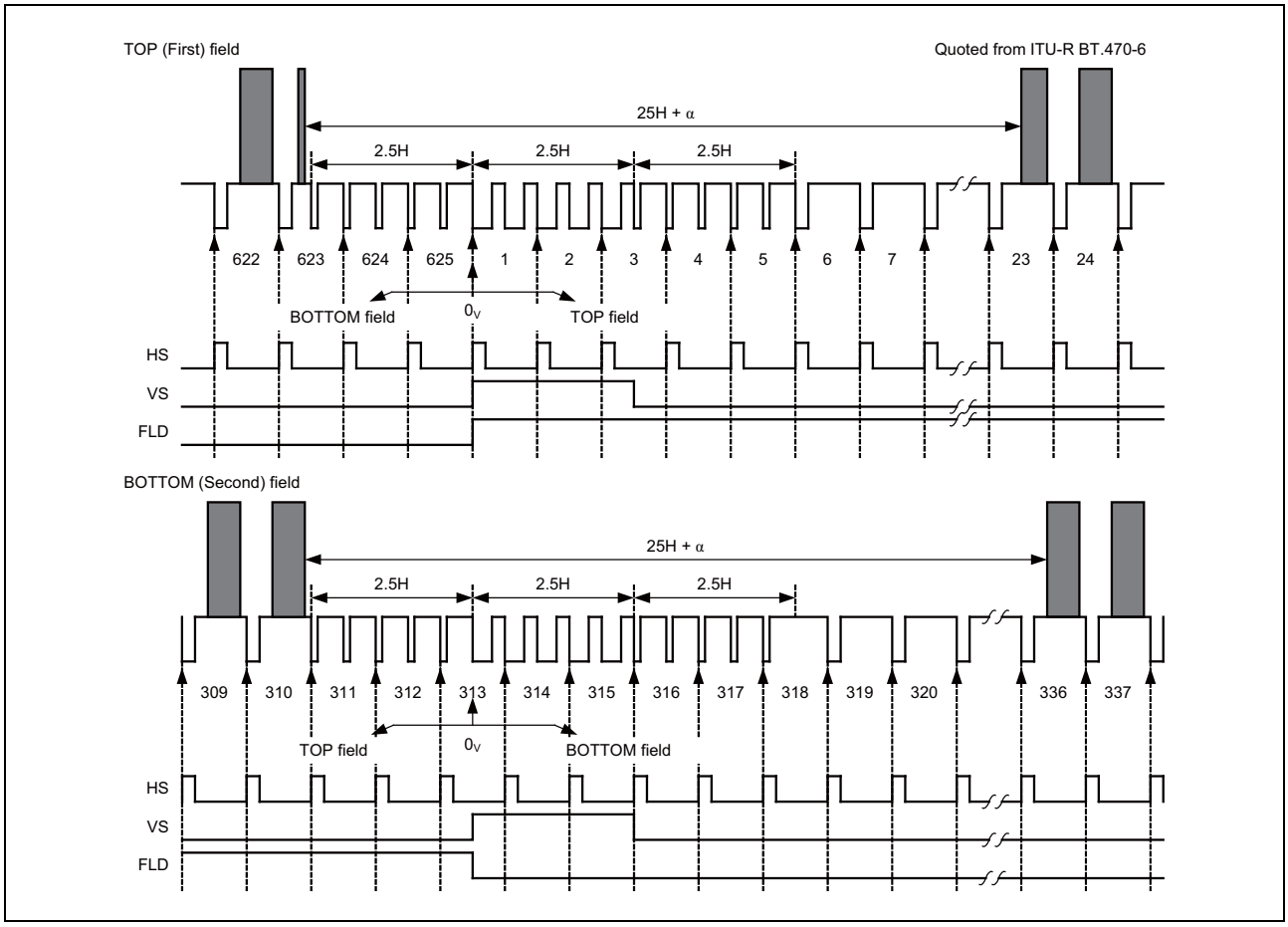


Figure 38.11 BT601 Vertical Timing (625 Lines/50.00 Hz)

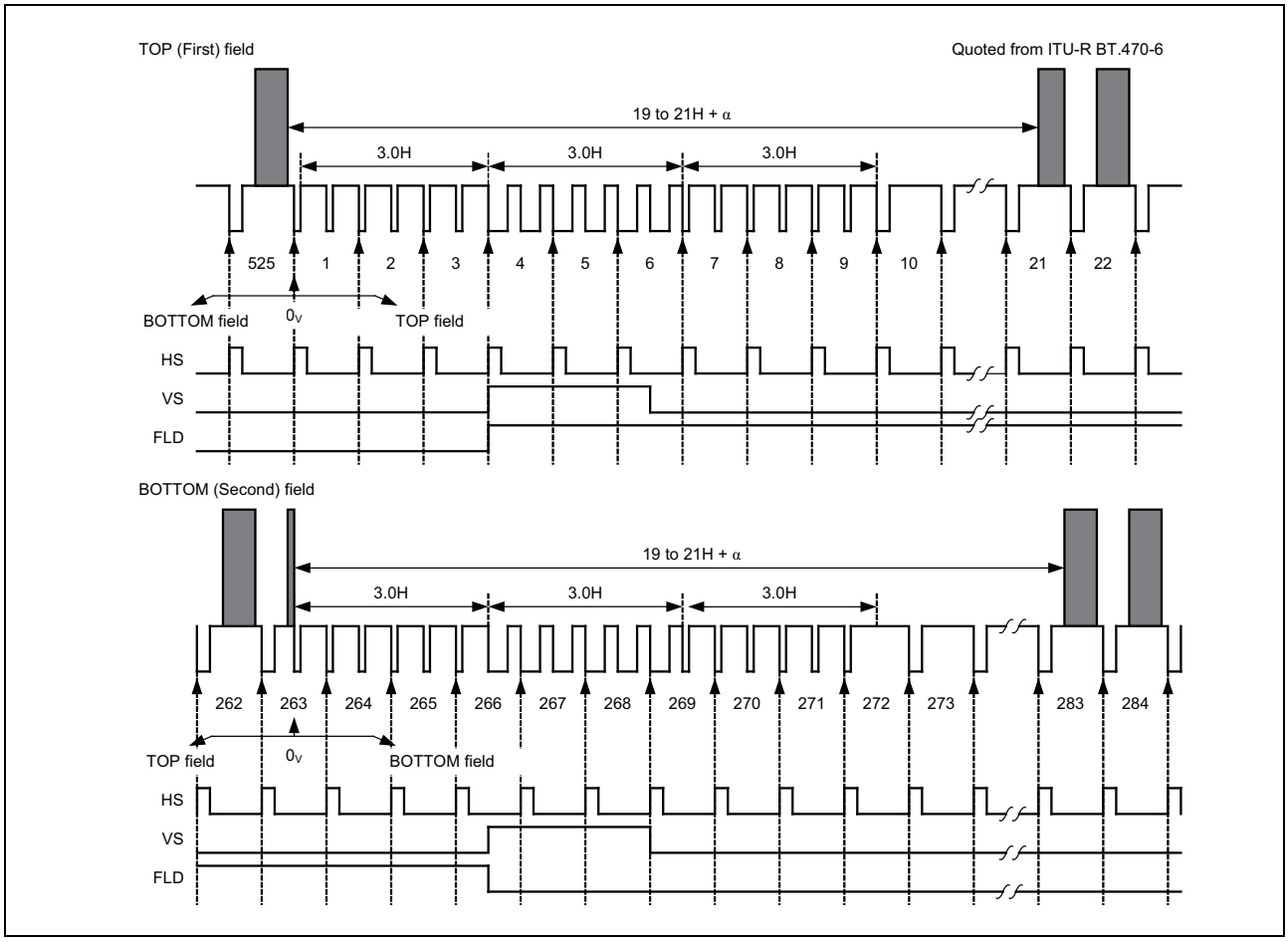


Figure 38.12 BT601 Vertical Timing (525 Lines/59.94 Hz)

38.3.1.8 Typical Signal Timing of BT656 Format

Figure 38.13 and **Figure 38.14** show the horizontal timings of the BT656 format.

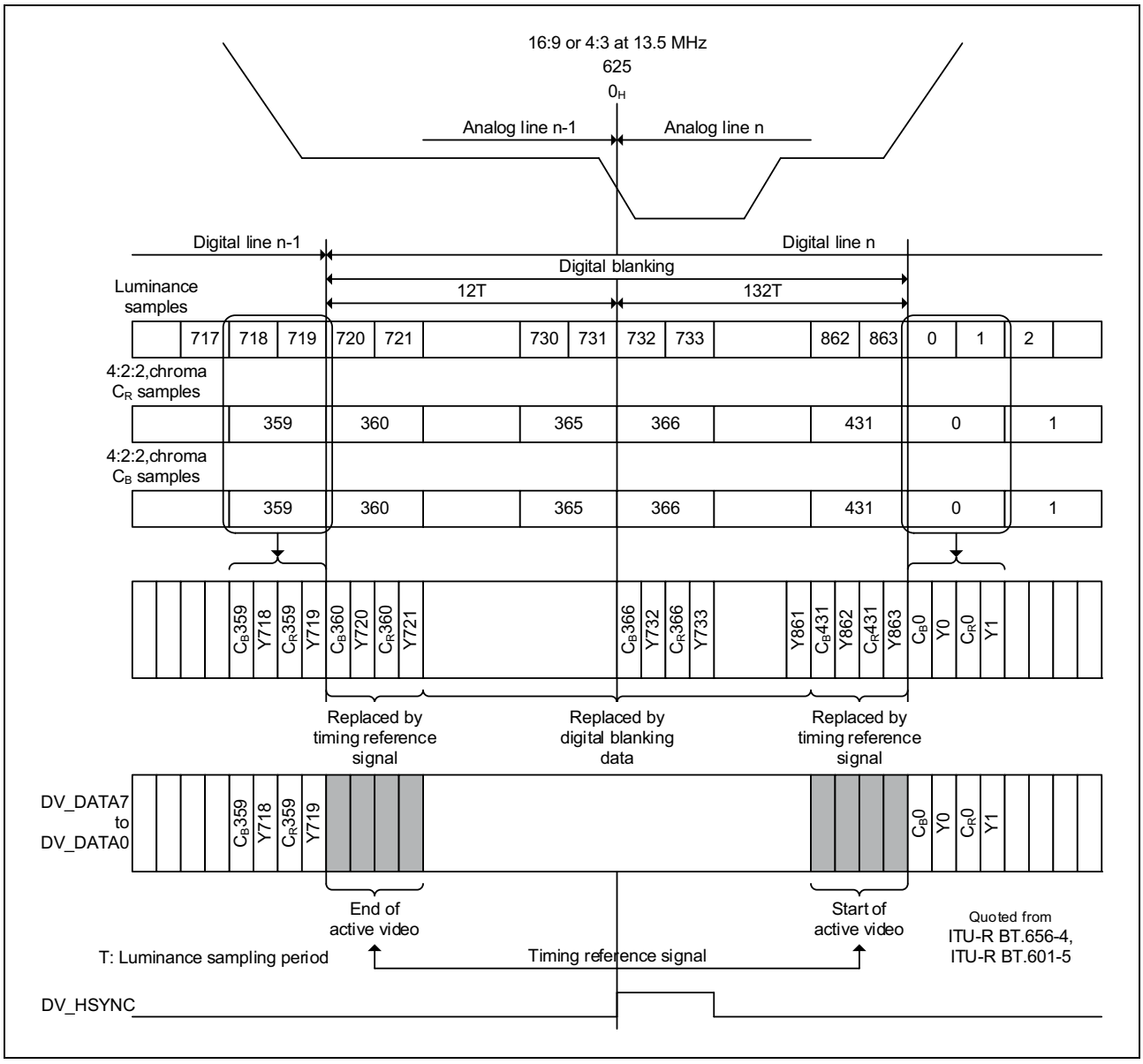


Figure 38.13 BT656 Horizontal Timing (625 Lines/50.00 Hz)

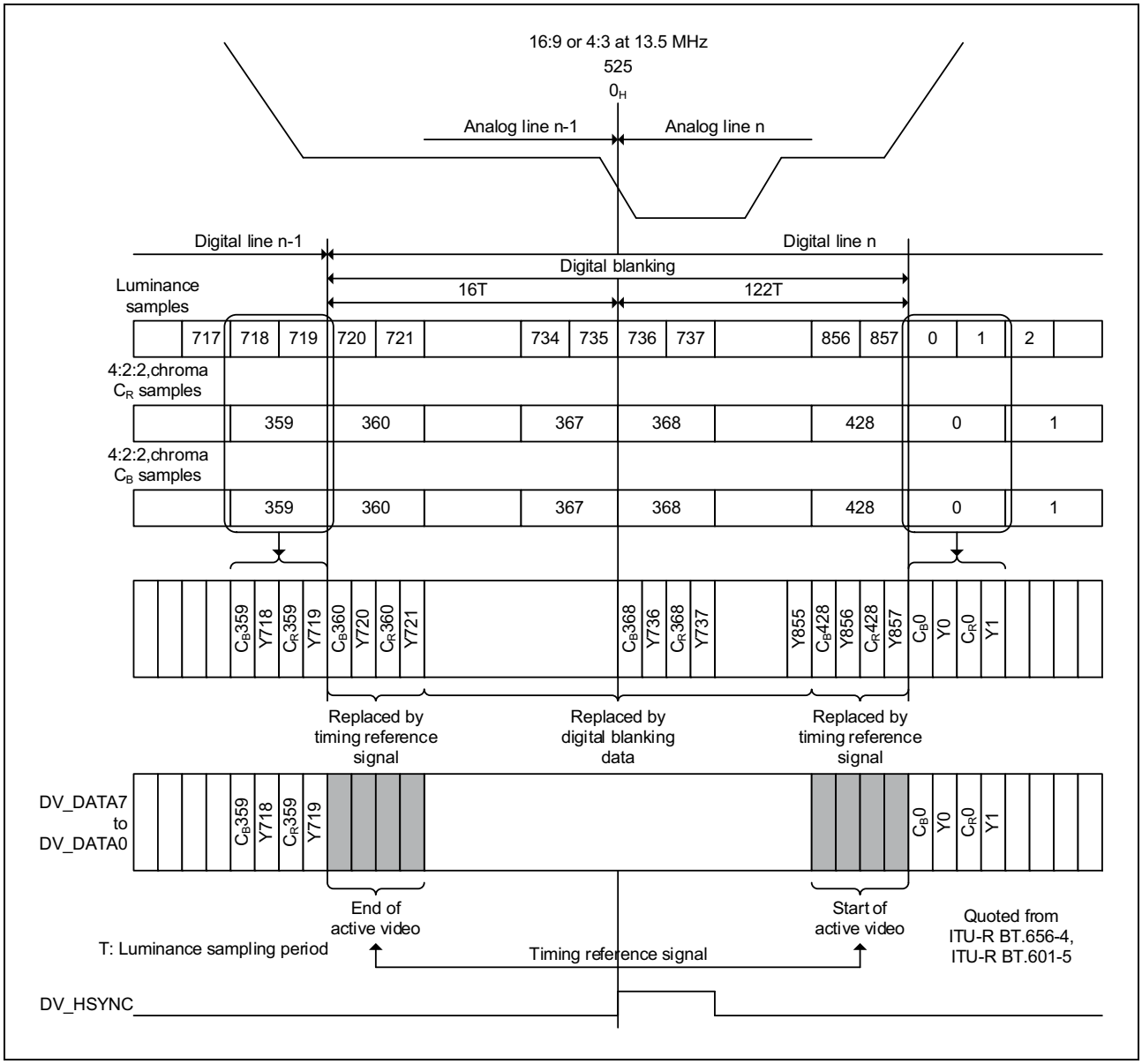


Figure 38.14 BT656 Horizontal Timing (525 Lines/59.94 Hz)

- Timing Example of 525-Line Interface Input in BT656 Format

The following figures show examples of the timing of vertical/horizontal synchronization signal extracted from 525-line interlaced input in the BT656 format.

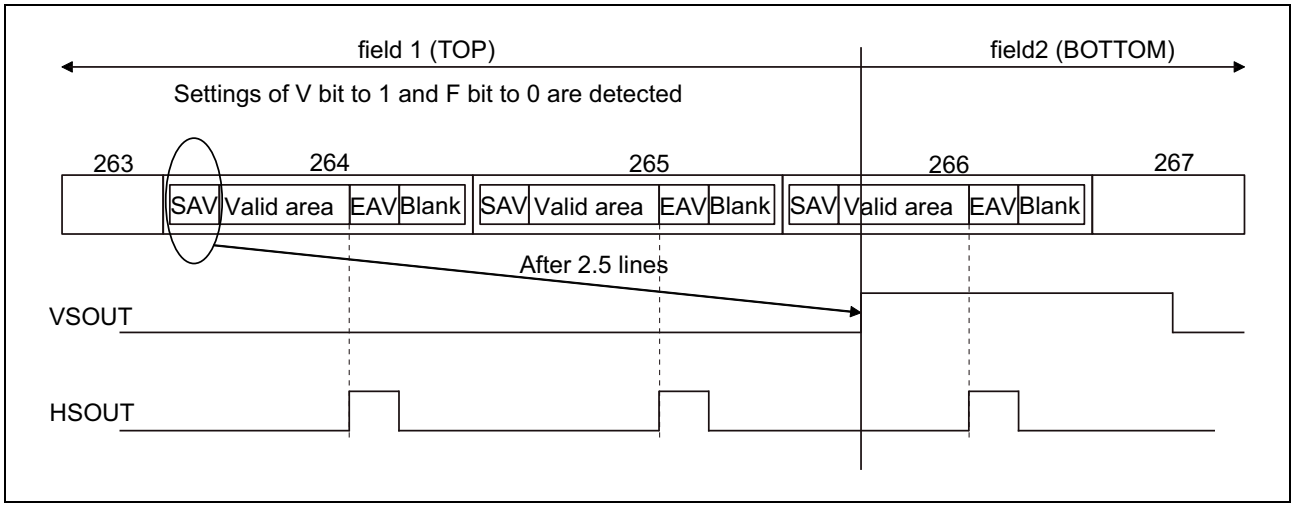


Figure 38.15 Timing of Vertical/Horizontal Synchronization Signal from 525-Line Interlaced Input in BT656 Format (Top to Bottom)

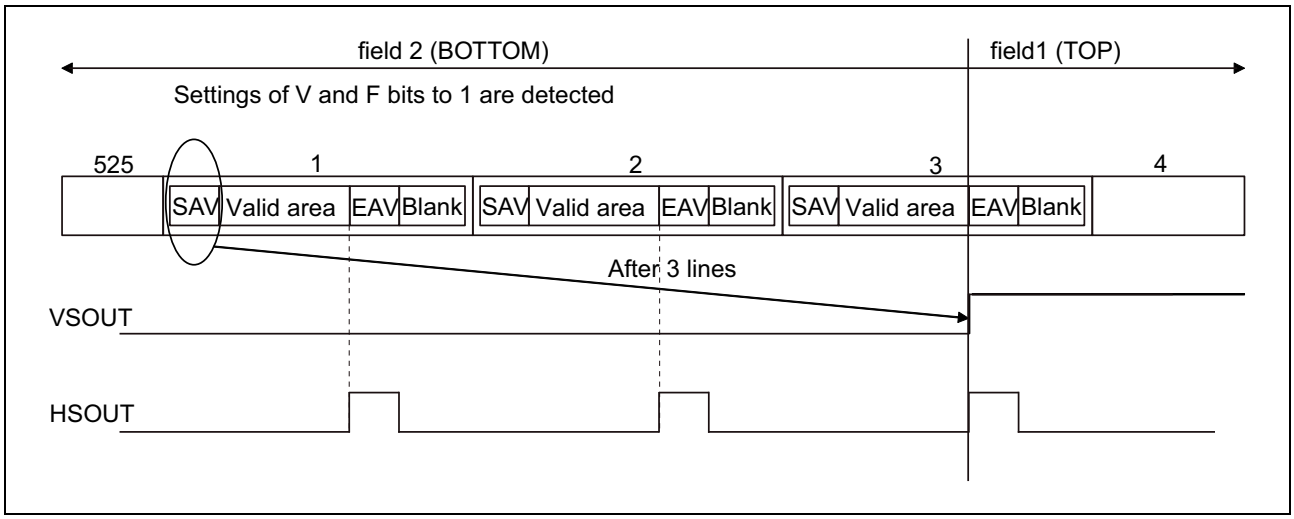


Figure 38.16 Timing of Vertical/Horizontal Synchronization Signal from 525-Line Interlaced Input in BT656 Format (Bottom to Top)

38.3.1.9 SAV/EAV Code in BT656 Format

Table 38.21 shows the timing of inserting the SAV/EAV code in the BT656 format. Bit information is shown in **Table 38.22** and **Table 38.23**. This module does not refer to the parity bits P3 to P0 shown in **Table 38.23**.

Table 38.21 SAV/EAV Code Insertion Timing (Line)

		625	525
V-digital field blanking			
Field 1	Start (V = 1)	Line 624	Line 1
	Finish (V = 0)	Line 23	Line 20
Field 2	Start (V = 1)	Line 311	Line 264
	Finish (V = 0)	Line 336	Line 283
V-digital field blanking			
Field 1	F = 0	Line 1	Line 4
Field 2	F = 1	Line 313	Line 266

Table 38.22 SAV/EAV Code Bit Information (1)

Data Bit Number	1st Word (FF)	2nd Word (00)	3rd Word (00)	4th Word (XY)
7 (MSB)	1	0	0	1
6	1	0	0	F
5	1	0	0	V
4	1	0	0	H
3	1	0	0	P3
2	1	0	0	P2
1	1	0	0	P1
0	1	0	0	P0

[Legend]

F = 0 during field 1

F = 1 during field 2

V = 0 elsewhere

V = 1 during field blanking

H = 0 is SAV

H = 1 is EAV

Table 38.23 SAV/EAV Code Bit Information (2)

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

Table 38.24 Table 38.24 and **Table 38.25** show the SAV/EAV code tables.

Table 38.24 SAV/EAV Code in BT656 Format (625 Lines/50.00 Hz)

		One Horizontal Period																	
		EAV				H blank	SAV				Valid area								
		1	2	3	4		285	286	287	288	289	290	291	292	...	1725	1726	1727	1728
Field1 (top)	1	FF	00	00	B6		FF	00	00	AB	Digital Blanking Data								
	:	FF	00	00	B6		FF	00	00	AB									
	22	FF	00	00	B6		FF	00	00	AB									
	23	FF	00	00	9D		FF	00	00	80	Cb0	Y0	Cr0	Y1	...	Cb359	Y718	Cr359	Y719
	:	FF	00	00	9D		FF	00	00	80	:	Valid pixel data area							:
	:	FF	00	00	9D		FF	00	00	80	:								:
	:	FF	00	00	9D		FF	00	00	80	:								:
	:	FF	00	00	9D		FF	00	00	80	:								:
	:	FF	00	00	9D		FF	00	00	80	:								:
	310	FF	00	00	9D		FF	00	00	80	Cb0	Y0	Cr0	Y1	...	Cb359	Y718	Cr359	Y719
	311	FF	00	00	B6		FF	00	00	AB	Digital Blanking Data								
	312	FF	00	00	B6		FF	00	00	AB									
Field2 (bottom)	313	FF	00	00	F1		FF	00	00	EC	Digital Blanking Data								
	:	FF	00	00	F1		FF	00	00	EC									
	335	FF	00	00	F1		FF	00	00	EC									
	336	FF	00	00	DA		FF	00	00	C7	Cb0	Y0	Cr0	Y1	...	Cb359	Y718	Cr359	Y719
	:	FF	00	00	DA		FF	00	00	C7	:	Valid pixel data area							:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	623	FF	00	00	DA		FF	00	00	C7	Cb0	Y0	Cr0	Y1	...	Cb359	Y718	Cr359	Y719
	624	FF	00	00	F1		FF	00	00	EC	Digital Blanking Data								
	625	FF	00	00	F1		FF	00	00	EC									

Table 38.25 SAV/EAV Code in BT656 Format (525 Lines/59.94 Hz)

		One Horizontal Period																	
		EAV				H blank	SAV				Valid area								
		1	2	3	4		273	274	275	276	277	278	279	280	...	1713	1714	1715	1716
Field2	1	FF	00	00	F1		FF	00	00	EC	Digital Blanking Data								
	2	FF	00	00	F1		FF	00	00	EC									
	3	FF	00	00	F1		FF	00	00	EC									
Field1 (top)	4	FF	00	00	B6		FF	00	00	AB	Digital Blanking Data								
	:	FF	00	00	B6		FF	00	00	AB									
	19	FF	00	00	B6		FF	00	00	AB									
	20	FF	00	00	9D		FF	00	00	80	Cb0	Y0	Cr0	Y1	...	Cb359	Y718	Cr359	Y719
	:	FF	00	00	9D		FF	00	00	80	:	Valid pixel data area							:
	:	FF	00	00	9D		FF	00	00	80	:								:
	:	FF	00	00	9D		FF	00	00	80	:								:
	:	FF	00	00	9D		FF	00	00	80	:								:
	:	FF	00	00	9D		FF	00	00	80	:								:
	263	FF	00	00	9D		FF	00	00	80	Cb0	Y0	Cr0	Y1	...	Cb359	Y718	Cr359	Y719
	264	FF	00	00	B6		FF	00	00	AB	Digital Blanking Data								
	265	FF	00	00	B6		FF	00	00	AB									
Field2 (bottom)	266	FF	00	00	F1		FF	00	00	EC	Digital Blanking Data								
	:	FF	00	00	F1		FF	00	00	EC									
	282	FF	00	00	F1		FF	00	00	EC									
	283	FF	00	00	DA		FF	00	00	C7	Cb0	Y0	Cr0	Y1	...	Cb359	Y718	Cr359	Y719
	:	FF	00	00	DA		FF	00	00	C7	:	Valid pixel data area							:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	:	FF	00	00	DA		FF	00	00	C7	:								:
	525	FF	00	00	DA		FF	00	00	C7	Cb0	Y0	Cr0	Y1	...	Cb359	Y718	Cr359	Y719

38.3.1.10 BT656 Progressive Format

This product can be connected with devices which output data in the BT656 progressive format. Because the standard for the BT656 format does not include description of output in the progressive format, there is no guarantee that this product is connected with devices which output data in the progressive format. The following description shows how to generate a vertical/horizontal synchronization signal by decoding the SAV/EAV code input via the BT656 interface of this module. Confirm the connection with devises which output data in the BT656 progressive format in accordance with this section.

(1) SAV/EAV Code

The SAV/EAV code consists of four words. When the first word is set to FF and the second and third words are set to 00, timing signals are generated by decoding the value of the fourth word (XY). For bit information, see **Table 38.22** in **Section 38.3.1.9**. This product does not refer to the parity bits (P3 to P0).

(2) Vertical/Horizontal Synchronization Signal

Based on the SAV/EAV code, the vertical/horizontal synchronization signal is generated.

(a) Vertical Synchronization Signal

The vertical synchronization signal is output when the value of the V bit is changed from 0 to 1 in the BT656 format. The timing of the output varies with the setting of INP_EXT_SYNC_CNT.INP_F525_625 setting and the value of the F bit in the BT656 format.

The table below lists the timing.

Table 38.26 Timing of Delay for Output of Vertical Synchronization Signal

INP_EXT_SYNC_CNT. INP_F525_625	F Bit in BT656 Format	Output Timing	Remark
0: 525 lines	0 (Field 1)	2.5 lines after setting of V bit to 1 is detected	525 lines, vertical synchronization signal for field 2
	0 (Field 2)	3 lines after setting of V bit to 1 is detected	525 lines, vertical synchronization signal for field 1
1: 625 lines	0 (Field 1)	2.5 lines after setting of V bit to 1 is detected	625 lines, vertical synchronization signal for field 2
	0 (Field 2)	2 lines after setting of V bit to 1 is detected	625 lines, vertical synchronization signal for field 1

(b) Horizontal Synchronization Signal

Based on the setting of the INP_EXT_SYNC_CNT.INP_H_EDGE_SEL bit, the horizontal synchronization signal is output.

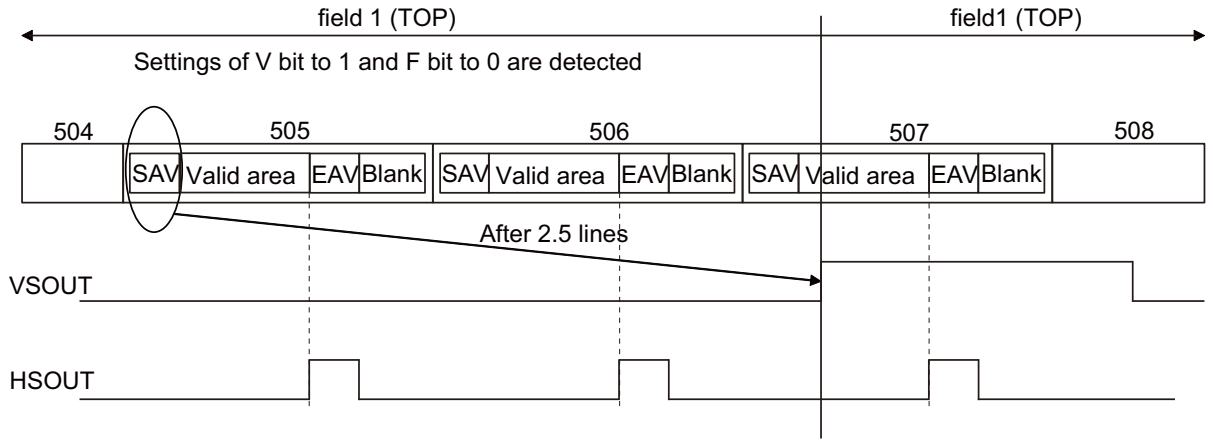
(3) Example of Timing for Progressive Input in BT656 Format

Table 38.27 shows an example of the SAV/EAV code in 525-line progressive input in the BT656 format. **Figure 38.17** shows the vertical/horizontal synchronization signal extracted from 525-line progressive input in the BT656 format. The field is detected as field 1 in this example, because the value of the F bit is set to 0 when that of the V bit is changed from 0 to 1. The field is regarded as the bottom field. The vertical synchronization signal is output 2.5 lines after the detection of the SAV code.

Table 38.27 SAV/EAV Code in BT656 Progressive Format (525 Lines/59.94 Hz)

		One Horizontal Period																
		EAV				H blank	SAV				Valid area							
		1	2	3	4		273	274	275	276	277	278	279	280	...	1713	1714	1715	1716
Field1 (top)	1	FF	00	00	BX		FF	00	00	AX	Digital Blanking Data								
	:	FF	00	00	BX		FF	00	00	AX									
	19	FF	00	00	BX		FF	00	00	AX									
	20	FF	00	00	9X		FF	00	00	8X	Cb0	Y0	Cr0	Y1	...	Cb359	Y718	Cr359	Y719
	:	FF	00	00	9X		FF	00	00	8X	:	Valid pixel data area							:
	:	FF	00	00	9X		FF	00	00	8X	:								:
	:	FF	00	00	9X		FF	00	00	8X	:								:
	:	FF	00	00	9X		FF	00	00	8X	:								:
	:	FF	00	00	9X		FF	00	00	8X	:								:
	504	FF	00	00	9X		FF	00	00	8X	Cb0	Y0	Cr0	Y1	...	Cb359	Y718	Cr359	Y719
	505	FF	00	00	BX		FF	00	00	AX	Digital Blanking Data								
	:	FF	00	00	BX		FF	00	00	AX									
	525	FF	00	00	BX		FF	00	00	AX									

X: Not referred



Note: In the case of BT601/656, because 2 pixel (YCbCr: Yn,Cbn,Crn,Yn+1) is captured at 4 cycle, horizontal size is doubled internally.
Therefore, in this case, horizontal size must be scale down to be half by using down-scaler in order to keep original input size.

The registers related to horizontal down-scaling is the below.
* SC_SCL0_DS1.SC_RES_DS_H_ON
* SC_SCL0_DS4.SC_RES_DS_H_RATIO
* SC_SCL0_DS7.SC_RES_OUT_HW

Figure 38.17 Vertical/Horizontal Synchronization Signal in BT656 Format (525 Lines, Progressive)

38.3.1.11 BT656/BT601/YCbCr422 Format Setting

The BT656 format can be used for the 525-line and 59.94-Hz or the 625-line and 50.00-Hz interlace signal format and progressive signal format (extended).

The BT601 format can be used for the 525-line and 59.94-Hz or the 625-line and 50.00-Hz interlace signal format and progressive signal format.

The YCbCr422 format can be used for the 525-line and 59.94-Hz or the 625-line and 50.00-Hz interlace signal format in the 16-bit extended data-bus format of the BT601 standard.

The Vsync signal timing for the 525-line BT656 format and 625-line BT656 format are different.

The operating mode is set by the INP_F525_625 bit.

Table 38.28 Operating Mode Setting for BT656 Format

Register Name	Bit Name	Initial Value	Description
INP_EXT_SYNC_CNT	INP_F525_625	0	Number of Lines for BT656 Input of External Input System 0: 525 lines 1: 625 lines

When the interlace signals are to be input in BT656/BT601/YCbCr422 format, half of 2fH phase timings of the Vsync signal and the Hsync signal are set with the INP_FH50[9:0] bits.

The INP_FH50[9:0] bits are also used for the vertical synchronous phase adjustment block. Therefore, for bit description, see **Table 38.33**.

When the external input is of BT656 format, the reference point of the Hsync signal is set with the INP_H_EDGE_SEL bit.

Table 38.29 Hsync Signal Reference Selection for BT656 Format

Register Name	Bit Name	Initial Value	Description
INP_EXT_SYNC_CNT	INP_H_EDGE_SEL	0	Hsync Signal Reference Select for BT656 Format of External Input System 0: EAV 1: SAV

When the external input is of BT656/BT601 format, the internal signal BTOUT[7:0], which is input from the DV_DATA pins and allocated, is expanded to the 24-bit YCbCr signal.

Expansion timing with respect to the Hsync signal reference is set with the INP_H_POS[1:0] bits.

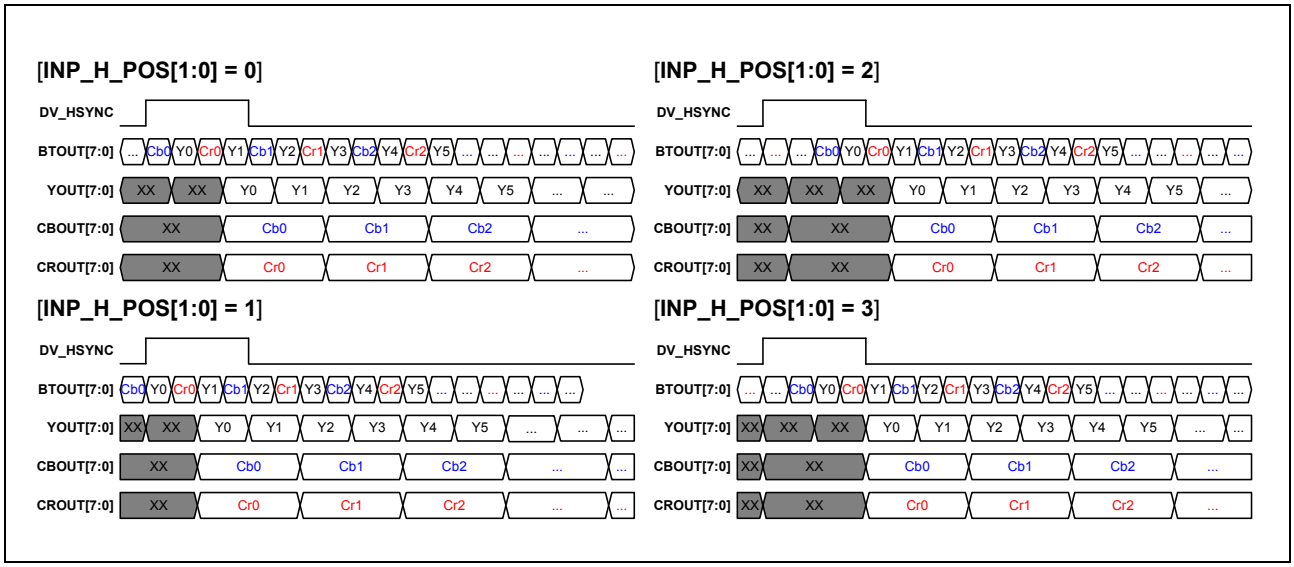


Figure 38.18 YCbCr Data Expansion for BT656/BT601 Input

NOTE

In the case of BT601/656, because 2 pixel (YCbCr:Yn,Cbn,Crn,Yn+1) is captured at 4 cycle, horizontal size is doubled internally.

Therefore, in this case, horizontal size must be scale down to be half by using down-scaler in order to keep original input size.

The registers related to horizontal down-scaling is the below.

- SC_SCL0_DS1.SC_RES_DS_H_ON
- SC_SCL0_DS4.SC_RES_DS_H_RATIO
- SC_SCL0_DS7.SC_RES_OUT_HW

Table 38.30 Data String Start Timing Selection for BT656/BT601 Input

Register Name	Bit Name	Initial Value	Description
INP_EXT_SYNC_CNT	INP_H_POS[1:0]	0	Y/Cb/Y/Cr Data String Start Timing with respect to Hsync Reference 0: Cb/Y/Cr/Y 1: Y/Cr/Y/Cb 2: Cr/Y/Cb/Y 3: Y/Cb/Y/Cr

When the external input is in YCbCr422 format, the input from the DV_DATA pins is allocated to the internal Y[7:0] and CbCr[7:0] signals, and the CbCr[7:0] are expanded to a 16-bit signal.

Expansion timing with respect to the Hsync signal reference is set with the INP_H_POS[1:0] bits.

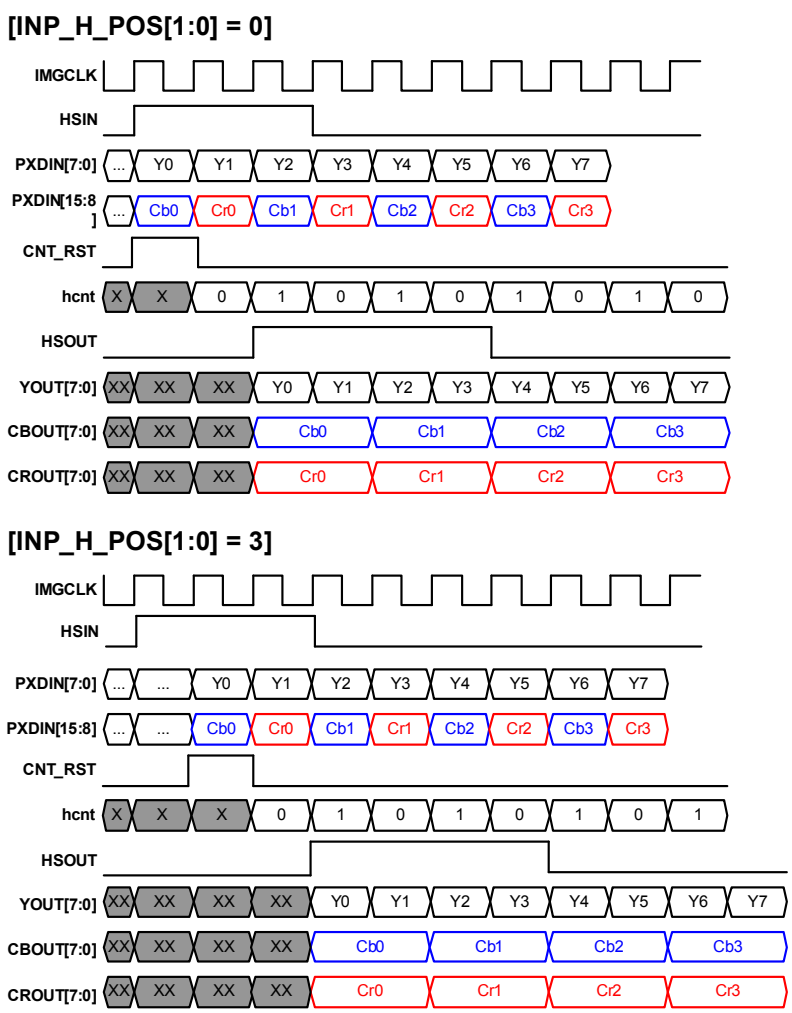


Figure 38.19 YCbCr Data Expansion for YCbCr422 Input

Table 38.31 Data String Start Timing Selection for YCbCr422 Input

Register Name	Bit Name	Initial Value	Description
INP_EXT_SYNC_CNT	INP_H_POS[1:0]	0	Cb/Cr Data String Start Timing with respect to Hsync Reference 0: Cb/Cr 3: Cr/Cb 1, 2: Setting prohibited

38.3.1.12 YCbCr444/RBG888/666/565 Input Timing

The YCbCr444/RGB888/666/565 format can be used for the progressive YCbCr/RGB signal.

The sync signal width (H_SYNC, V_SYNC), sync signal polarity (H_POL, V_POL), valid period start position (H_BP, V_BP), valid period end position (H_FP, V_FP), and valid period video width (H_ACTIVE, V_ACTIVE) are shown in **Table 38.32**.

Table 38.32 YCbCr/RGB Signal Reception Timing

Item	Description
External input clock	Maximum external input clock frequency: up to 48.00 MHz
Vsync signal width (V_SYNC)	Minimum Vsync signal width: 1 CLK
Vsync signal polarity (V_POL)	Positive or negative polarity is selected by the relevant registers.
Vertical valid period start position (V_BP)	From Vsync reference to the head of the video image: 4 lines or more
Vertical valid period video width (V_ACTIVE)	Maximum vertical valid period: 1024 lines
Vertical valid period end position (V_FP)	From the end of the video image to the Vsync reference: 4 lines or more
Hsync signal width (H_SYNC)	Minimum Hsync signal width: 1 CLK
Hsync signal polarity (H_POL)	Positive or negative polarity is selected by the relevant registers.
Horizontal valid period start position (H_BP)	From Hsync reference to the head of the video image: 16 CLK or more
Horizontal valid period video width (H_ACTIVE)	Maximum horizontal valid period: up to 1024 pixels
Horizontal valid period end position (H_FP)	From the end of the video image to the Hsync reference: 16 CLK or more

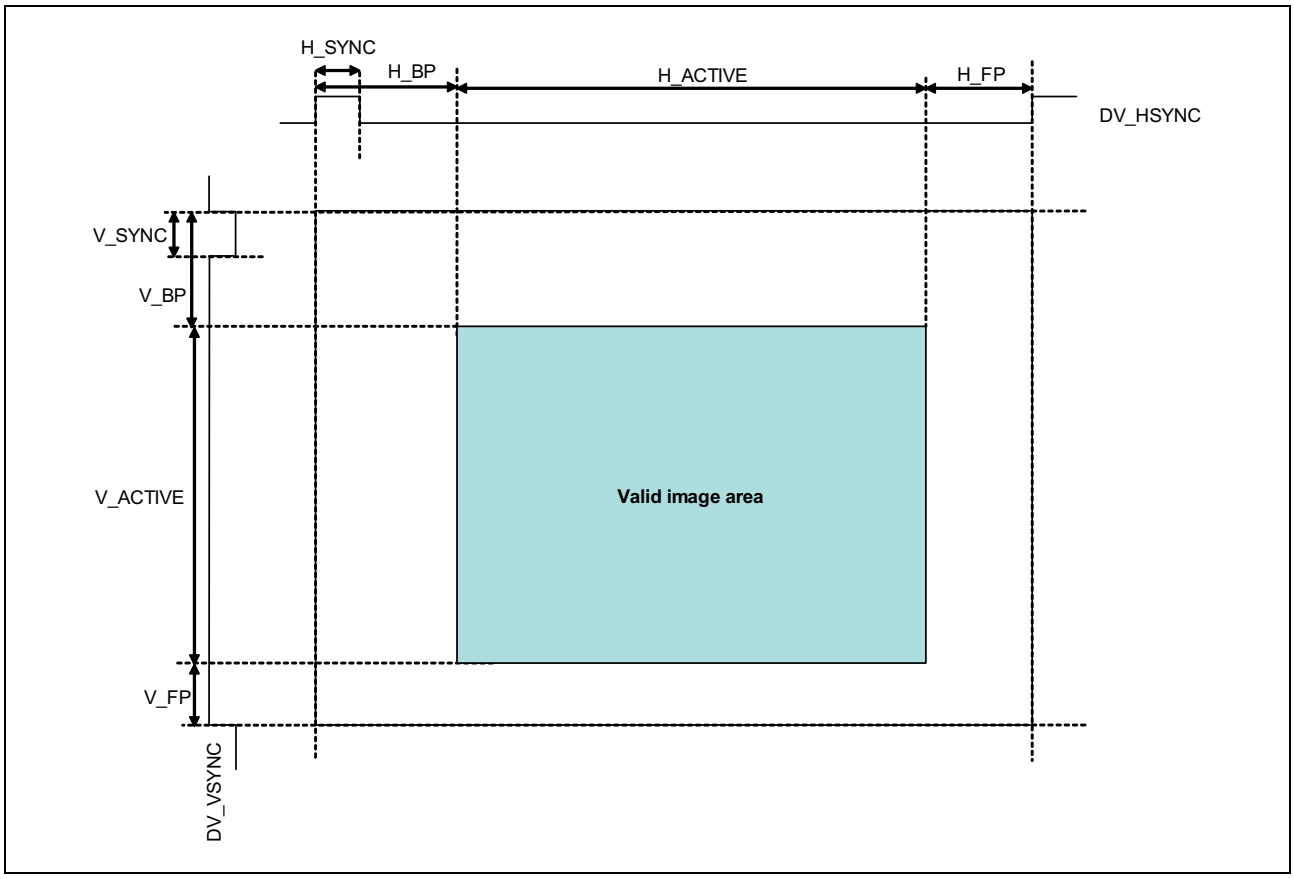


Figure 38.20 YCbCr/RGB Signal Reception Timing

38.3.1.13 Field Differentiation and Vsync Signal Phase Adjustment

The phase of the input Vsync signal and Hsync signal is detected and the field of the interlace signal is determined. When the reference point of the Vsync signal is detected within ±0.5 horizontal period with respect to the Hsync signal, it is determined as the interlace top field. When the reference point of the Vsync signal is detected outside ±0.5 horizontal period with respect to the Hsync signal, it is determined as the interlace bottom field.

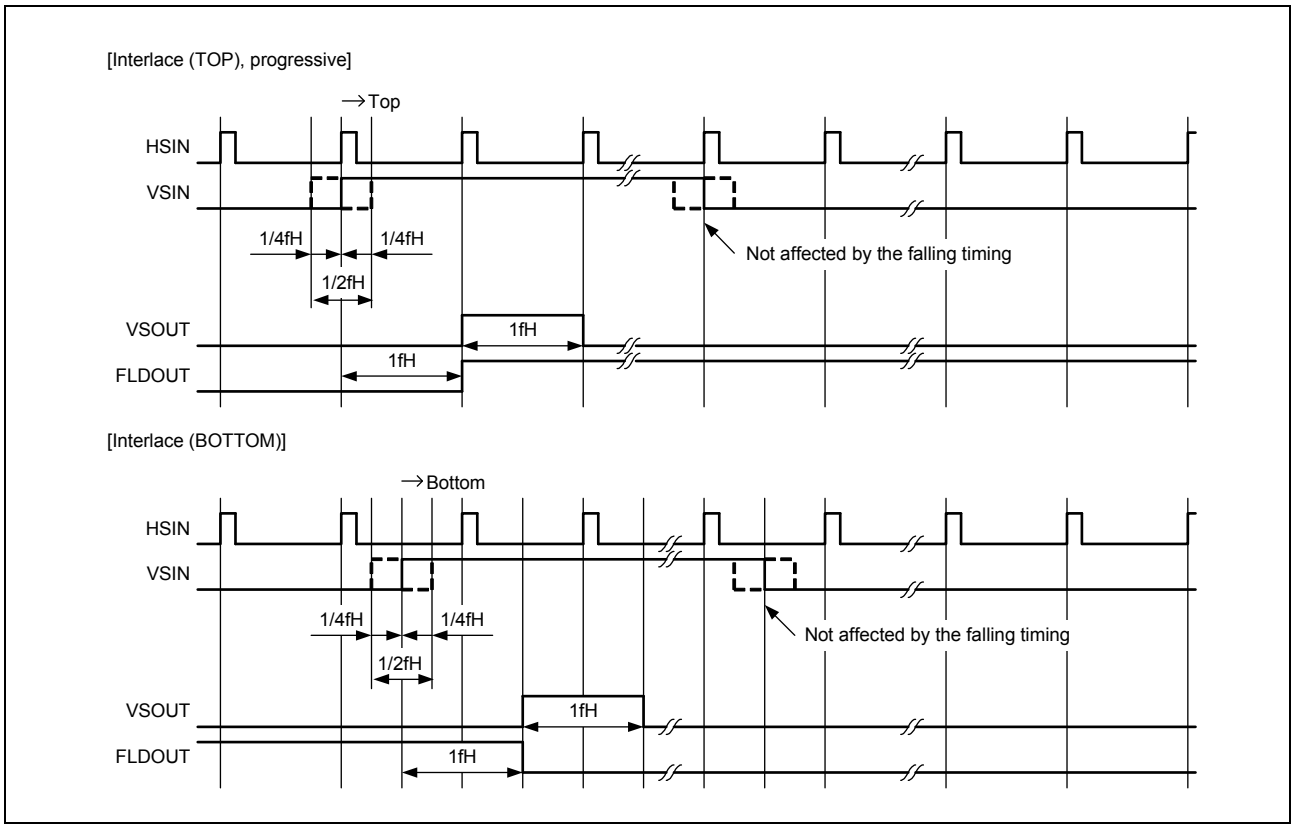


Figure 38.21 Vsync Signal Phase Adjustment

The timings of 1/2fH Vsync signal phase and 1/4fH Vsync signal phase are set with INP_FH50[9:0] and INP_FH25[9:0], respectively.

Table 38.33 Vsync Signal Phase Timing Setting

Register Name	Bit Name	Initial Value	Description
INP_VSYNC_PH_ADJ	INP_FH50[9:0]	858	Vsync Signal 1/2fH Phase Timing Should be 1/2 the horizontal cycle.
INP_VSYNC_PH_ADJ	INP_FH25[9:0]	429	Vsync Signal 1/4fH Phase Timing Should be 1/4 the horizontal cycle.

38.3.1.14 Vsync Signal Delay Adjustment in Line Units

The Vsync signal line delay adjust block can delay the Vsync signal and the field differentiation signal in line units.

When a video signal with a short vertical front porch is input, the vertical front porch is adjusted.

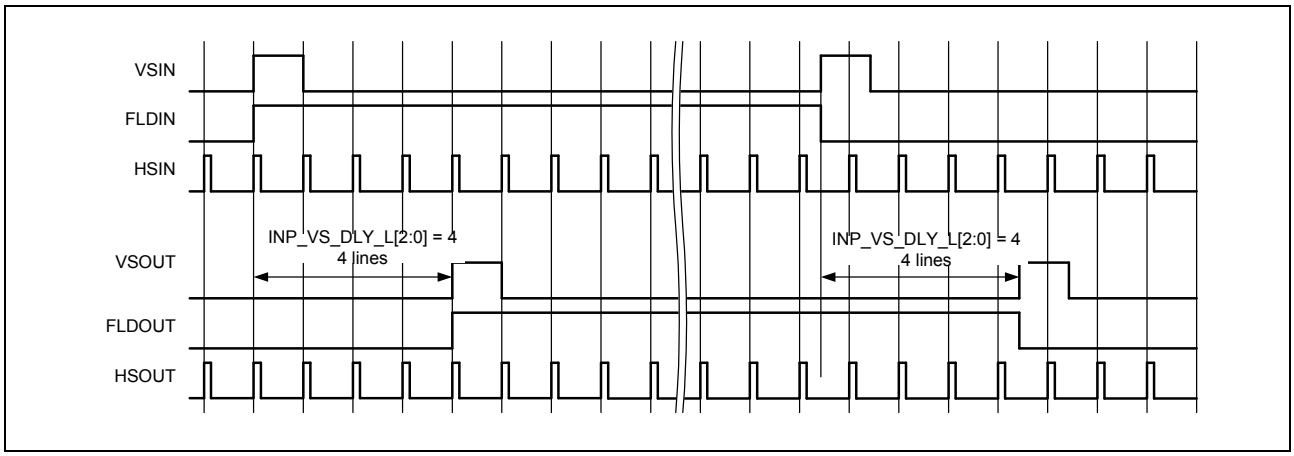


Figure 38.22 Timing of Vsync Signal Delay in Line Units

Table 38.34 Adjustment of Vsync Signal Delay in Line Units

Register Name	Bit Name	Initial Value	Description
INP_DLY_ADJ	INP_VS_DLY_L[2:0]	0	Number of Lines for Delaying Vsync Signal and Field Differentiation Signal Delay amount: 0 to 7 (lines)

38.3.1.15 Sync Signal Delay Adjustment

Delay can be adjusted independently for the Vsync signal, Hsync signal, and field differentiation signal in the units of clock.

Lacking margin of the horizontal front porch is adjusted according to the input synchronization disturbance.

Table 38.35 Sync Signal Delay Adjustment

Register Name	Bit Name	Initial Value	Description
INP_DLY_ADJ	INP_VS_DLY[7:0]	0	Vsync Signal Delay Amount Delay amount: 0 to 254 (clock cycles)
INP_DLY_ADJ	INP_HS_DLY[7:0]	0	Hsync Signal Delay Amount Delay amount: 0 to 254 (clock cycles)
INP_DLY_ADJ	INP_FLD_DLY[7:0]	0	Field Differentiation Signal Delay Amount Delay amount: 0 to 254 (clock cycles)

38.3.1.16 Color Matrix

By using a color matrix, input signal offsets and nine-axis gain can be adjusted. This enables brightness adjustment, gain adjustment, and YCbCr and GBR mutual conversion.

(1) GBR to GBR Conversion

YGIN_A = YGIN + IMGCNT_MTX_YG – 128

CBBIN_A = CBBIN + IMGCNT_MTX_B – 128

CRRIN_A = CRRIN + IMGCNT_MTX_R – 128

YGOUT = (IMGCNT_MTX_GG×YGIN_A + IMGCNT_MTX_GB×CBBIN_A + IMGCNT_MTX_GR×CRRIN_A) ÷ 256

CBBOUT = (IMGCNT_MTX_BG×YGIN_A + IMGCNT_MTX_BB×CBBIN_A + IMGCNT_MTX_BR×CRRIN_A) ÷ 256

CRROUT = (IMGCNT_MTX_RG×YGIN_A + IMGCNT_MTX_RB×CBBIN_A + IMGCNT_MTX_RR×CRRIN_A) ÷ 256

(2) GBR to YCbCr Conversion

YGIN_A = YGIN + IMGCNT_MTX_YG – 128

CBBIN_A = CBBIN + IMGCNT_MTX_B – 128

CRRIN_A = CRRIN + IMGCNT_MTX_R – 128

YGOUT = (IMGCNT_MTX_GG×YGIN_A + IMGCNT_MTX_GB×CBBIN_A + IMGCNT_MTX_GR×CRRIN_A) ÷ 256

CBBOUT = (IMGCNT_MTX_BG×YGIN_A + IMGCNT_MTX_BB×CBBIN_A + IMGCNT_MTX_BR×CRRIN_A) ÷ 256 + 128

CRROUT = (IMGCNT_MTX_RG×YGIN_A + IMGCNT_MTX_RB×CBBIN_A + IMGCNT_MTX_RR×CRRIN_A) ÷ 256 + 128

Table 38.36 Matrix Coefficient (Typical Value) for SMPTE 293M

	YGIN		CBBIN		CRRIN	
	Coefficient	Set Value	Coefficient	Set Value	Coefficient	Set Value
YGOUT	0.587	IMGCNT_MTX_GG = 150	0.114	IMGCNT_MTX_GB = 29	0.299	IMGCNT_MTX_GR = 77
CBBOUT	–0.331	IMGCNT_MTX_BG = 1963	0.500	IMGCNT_MTX_BB = 128	–0.169	IMGCNT_MTX_BR = 2005
CRROUT	–0.419	IMGCNT_MTX_RG = 1941	–0.081	IMGCNT_MTX_RB = 2027	0.500	IMGCNT_MTX_RR = 128

(3) YCbCr to GBR Conversion

YGIN_A = YGIN + IMGCNT_MTX_YG – 128

CBBIN_A = CBBIN – 128

CRRIN_A = CRRIN – 128

YGOUT = (IMGCNT_MTX_GG×YGIN_A + IMGCNT_MTX_GB×CBBIN_A + IMGCNT_MTX_GR×CRRIN_A) ÷ 256

CBBOUT = (IMGCNT_MTX_BG×YGIN_A + IMGCNT_MTX_BB×CBBIN_A + IMGCNT_MTX_BR×CRRIN_A) ÷ 256

CRROUT = (IMGCNT_MTX_RG×YGIN_A + IMGCNT_MTX_RB×CBBIN_A + IMGCNT_MTX_RR×CRRIN_A) ÷ 256

Table 38.37 Matrix Coefficient (Typical Value) for SMPTE 293M

	YGIN		CBBIN		CRRIN	
	Coefficient	Set Value	Coefficient	Set Value	Coefficient	Set Value
YGOUT	1.000	IMGCNT_MTX_GG = 256	–0.344	IMGCNT_MTX_GB = 1960	–0.714	IMGCNT_MTX_GR = 1865
CBBOUT	1.000	IMGCNT_MTX_BG = 256	1.772	IMGCNT_MTX_BB = 454	0.000	IMGCNT_MTX_BR = 0
CRROUT	1.000	IMGCNT_MTX_RG = 256	0.000	IMGCNT_MTX_RB = 0	1.402	IMGCNT_MTX_RR = 359

(4) YCbCr to YCbCr Conversion

YGIN_A = YGIN + IMGCNT_MTX_YG – 128

CBBIN_A = CBBIN – 128

CRRIN_A = CRRIN – 128

YGOUT = (IMGCNT_MTX_GG×YGIN_A + IMGCNT_MTX_GB×CBBIN_A + IMGCNT_MTX_GR×CRRIN_A) ÷ 256

CBBOUT = (IMGCNT_MTX_BG×YGIN_A + IMGCNT_MTX_BB×CBBIN_A + IMGCNT_MTX_BR×CRRIN_A) ÷ 256 + 128

CRROUT = (IMGCNT_MTX_RG×YGIN_A + IMGCNT_MTX_RB×CBBIN_A + IMGCNT_MTX_RR×CRRIN_A) ÷ 256 + 128

Table 38.38 YCbCr to GBR Conversion (1/2)

Register Name	Bit Name	Initial Value	Description
IMGCNT_MTX_MODE	IMGCNT_MTX_MD [1:0]	3	Operating Mode 0: GBR → GBR 1: GBR → YCbCr 2: YCbCr → GBR 3: YCbCr → YCbCr
IMGCNT_MTX_YG_ADJ0	IMGCNT_MTX_YG [7:0]	128	Offset (DC) Adjustment of Y/G Signal Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB], 512 [LSB])
IMGCNT_MTX_CBB_ADJ0	IMGCNT_MTX_B [7:0]	128	Offset (DC) Adjustment of B Signal Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
IMGCNT_MTX_CRR_ADJ0	IMGCNT_MTX_R [7:0]	128	Offset (DC) Adjustment of R Signal Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
IMGCNT_MTX_YG_ADJ0	IMGCNT_MTX_GG [10:0]	256	Y/G Signal Gain Adjustment for Y/G Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_YG_ADJ1	IMGCNT_MTX_GB [10:0]	0	Cb/B Signal Gain Adjustment for Y/G Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_YG_ADJ1	IMGCNT_MTX_GR [10:0]	0	Cr/R Signal Gain Adjustment for Y/G Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Table 38.38 YCbCr to GBR Conversion (2/2)

Register Name	Bit Name	Initial Value	Description
IMGCNT_MTX_CBB_ ADJ0	IMGCNT_MTX_BG [10:0]	0	Y/G Signal Gain Adjustment for Cb/B Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_CBB_ ADJ1	IMGCNT_MTX_BB [10:0]	256	Cb/B Signal Gain Adjustment for Cb/B Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_CBB_ ADJ1	IMGCNT_MTX_BR [10:0]	0	Cr/R Signal Gain Adjustment for Cb/B Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_CRR_ ADJ0	IMGCNT_MTX_RG [10:0]	0	Y/G Signal Gain Adjustment for Cr/R Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_CRR_ ADJ1	IMGCNT_MTX_RB [10:0]	0	Cb/B Signal Gain Adjustment for Cr/R Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
IMGCNT_MTX_CRR_ ADJ1	IMGCNT_MTX_RR [10:0]	256	Cr/R Signal Gain Adjustment for Cr/R Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

38.3.1.17 Video input by data enable (D1M1A, D1M1-V2 and D1M2(H) only)

(1) Description

When video input synchronize with data enable (= DE) instead of HSYNC, it is available by special hardware setting.

Support format

Parallel video input is available with DE.

- RGB888, RGB565, YCbCr444, YCbCr422
- Interlace input can't be supported.

Restrictions

- When video input is used by HSYNC, DEMODE0 and DEMODE1 must be set to 0 before video clock input.
- When video input by data enable is used, delay adjustment of sync signal is not available. INP_DLY_ADJ must be set to 0.

(2) Input timing by data enable

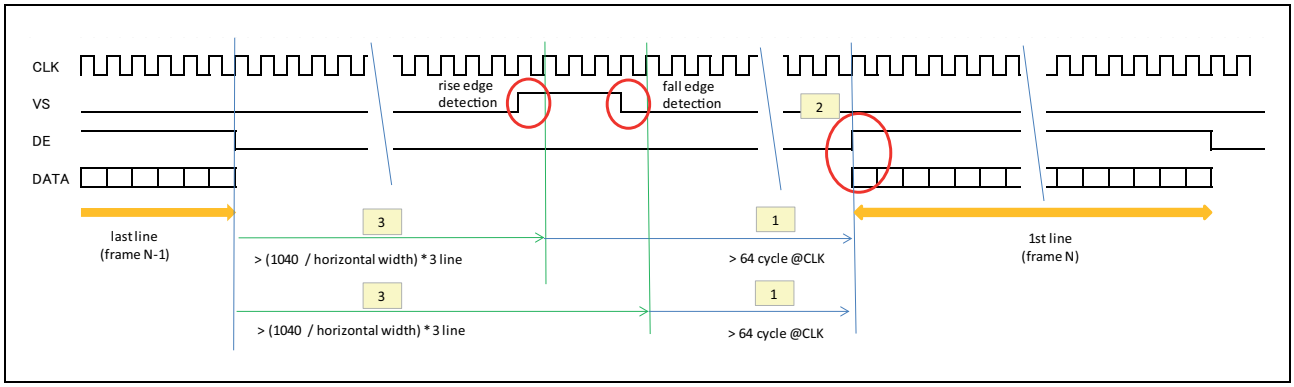


Figure 38.23 Input timing by data enable

1. VSYNC must be asserted before over 64 cycles which 1st line comes.
   In VDCE, internal 4 HSYNC pulses are generated between VS and the first DE.Therefore, vertical back porch (SC_SCL0_DS2.SC_RES_VS) must be fixed to 4.
2. The data is captured from the first active DE.
   For this setting, horizontal back porch (SC_SCL0_DS3.SC_RES_HS) must be setting to 1.Data is captured for cycles in SC_SCL0_DS3.SC_RES_HW.
3. VSYNC must not be asserted before the VDCE finished internal handling of the previous frame. [The period from the end of the last line to the next VSYNC detection]

– VSYNC rise edge detection case:
(vertical front porch) > (1040 / horizontal width) + 3
– VSYNC fall edge detection case:
(vertical front porch + vertical sync) > (1040 / horizontal width) + 3

Table 38.39 Video input by data enable

Register Name	Bit Name	Initial Value	Description
DEMODE0	DE_4HS_EN	0	Vertical position is shifted before 4 lines from the timing configured at other registers. 1: enable 0: disable For data enable input method, this bit must be set to 1. When this bit is 1, VDCE shift 4 lines internally after VSYNC detection. Therefore, SC_SCL0_DS2.SC_RES_VS must be fixed to 4.
	DE_HSBYPS_EN	0	Starting timing to capture video data by DV_HSYNC is shifted before 1 DV_CLK. 1: enable 0: disable For data enable input method, this bit must be set to 1. When this bit is 1 and SC_SCL0_DS3.SC_RES_HS is 1, VDCE can capture data from first cycle of data enable.
	DE_VLAST_EN	0	Interrupt timing changes to end of the final line by using setting of DEMODE1. 1: enable 0: disable This bit is available for the detection of the end of the last line by SC_SCL0_INT.SC_RES_LINE at data enable input. When this bit is 1, DE_VLAST[10:0] and SC_RES_LINE must be set. For the detail, please refer setting example. **Note:** This function can be only used for input vertical size is multiple of 16.
DEMODE1	DE_VLAST[10:0]	0	Active vertical line number at data enable input This register is available when DE_VLAST_EN is 1. Line number must be set by 16 line alignment, so Bit 4-0 of this register must be set to 0.

(3) How to set for data enable input

1. Fix the capturing position
   – SC_SCL0_DS2.SC_RES_VS = 4
   – SC_SCL0_DS3.SC_RES_HS = 1
2. DEMODE0 and DEMODE1 setting

   [No line detection]

   – DEMODE0=0x00000018
   (DE_4HS_EN=1, DE_HSBYPS_EN=1)

   [Line detection after the last line finished]

- DEMODE0=0x0000001C
  (DE_4HS_EN=1, DE_HSBYPS_EN=1, DE_VLAST_EN=1)
- DEMODE1= active line number
  e.g) when active vertical width is 480, DEMODE1 is 480.
- SC_SCL0_INT.SC_RES_LINE = active line number + 4
  e.g) when active vertical width is 480, SC_RES_LINE is 484.

3. Other setting
   - INP_DLY_ADJ = 0
   - Other VDCE settings are the same as video input setting with HSYNC.

38.3.2 Register Descriptions

Table 38.40 to **Table 38.45** show register Configuration.

- Symbols used in Register Description:

  Initial value: Register value after a power-on reset

  —: Undefined value

  R/W: Readable/writable. The written value can be read.

  R/WC0: Readable/writable. Writing 0 initializes the bit. Writing 1 is ignored.

  R/WC1: Readable/writable. Writing 1 initializes the bit. Writing 0 is ignored.

  R: Read-only. The write value should always be 0.

  —/W: Write-only. The read value is undefined.

Table 38.40 Register Configuration of Input Controller (Channel 0)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	External input block register update control register	INP_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1400	32
VDCE0	Input select control register	INP_SEL_CNT	R/W	H'0000 0000	<VDCE0_base> + 1404	32
VDCE0	External input sync signal control register	INP_EXT_SYNC_CNT	R/W	H'0000 0000	<VDCE0_base> + 1408	32
VDCE0	Vsync signal phase adjustment register	INP_VSYNC_PH_ADJ	R/W	H'035A 01AD	<VDCE0_base> + 140C	32
VDCE0	Sync signal delay adjustment register	INP_DLY_ADJ	R/W	H'0000 0000	<VDCE0_base> + 1410	32

Table 38.41 Register Configuration of Image Quality Adjustment Block (Channel 0)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Image quality adjustment block register update control register	IMGCNT_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1480	32
VDCE0	Image quality adjustment block matrix mode register	IMGCNT_MTX_MODE	R/W	H'0000 0003	<VDCE0_base> + 14A0	32
VDCE0	Image quality adjustment block matrix YG adjustment register 0	IMGCNT_MTX_YG_ADJ0	R/W	H'0080 0100	<VDCE0_base> + 14A4	32
VDCE0	Image quality adjustment block matrix YG adjustment register 1	IMGCNT_MTX_YG_ADJ1	R/W	H'0000 0000	<VDCE0_base> + 14A8	32
VDCE0	Image quality adjustment block matrix CBB adjustment register 0	IMGCNT_MTX_CBB_ADJ0	R/W	H'0080 0000	<VDCE0_base> + 14AC	32
VDCE0	Image quality adjustment block matrix CBB adjustment register 1	IMGCNT_MTX_CBB_ADJ1	R/W	H'0100 0000	<VDCE0_base> + 14B0	32
VDCE0	Image quality adjustment block matrix CRR adjustment register 0	IMGCNT_MTX_CRR_ADJ0	R/W	H'0080 0000	<VDCE0_base> + 14B4	32
VDCE0	Image quality adjustment block matrix CRR adjustment register 1	IMGCNT_MTX_CRR_ADJ1	R/W	H'0000 0100	<VDCE0_base> + 14B8	32
VDCE0	Dynamic range compression register	IMGCNT_DRC_REG	R/W	H'0000 0000	<VDCE0_base> + 14C0	32

Table 38.42 Register Configuration of Input Controller (Channel 1)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	External input block register update control register	INP_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1400	32
VDCE1	Input select control register	INP_SEL_CNT	R/W	H'0000 0000	<VDCE1_base> + 1404	32
VDCE1	External input sync signal control register	INP_EXT_SYNC_CNT	R/W	H'0000 0000	<VDCE1_base> + 1408	32
VDCE1	Vsync signal phase adjustment register	INP_VSYNC_PH_ADJ	R/W	H'035A 01AD	<VDCE1_base> + 140C	32
VDCE1	Sync signal delay adjustment register	INP_DLY_ADJ	R/W	H'0000 0000	<VDCE1_base> + 1410	32

Table 38.43 Register Configuration of Image Quality Adjustment Block (Channel 1)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Image quality adjustment block register update control register	IMGCNT_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1480	32
VDCE1	Image quality adjustment block matrix mode register	IMGCNT_MTX_MODE	R/W	H'0000 0003	<VDCE1_base> + 14A0	32
VDCE1	Image quality adjustment block matrix YG adjustment register 0	IMGCNT_MTX_YG_ADJ0	R/W	H'0080 0100	<VDCE1_base> + 14A4	32
VDCE1	Image quality adjustment block matrix YG adjustment register 1	IMGCNT_MTX_YG_ADJ1	R/W	H'0000 0000	<VDCE1_base> + 14A8	32
VDCE1	Image quality adjustment block matrix CBB adjustment register 0	IMGCNT_MTX_CBB_ADJ0	R/W	H'0080 0000	<VDCE1_base> + 14AC	32
VDCE1	Image quality adjustment block matrix CBB adjustment register 1	IMGCNT_MTX_CBB_ADJ1	R/W	H'0100 0000	<VDCE1_base> + 14B0	32
VDCE1	Image quality adjustment block matrix CRR adjustment register 0	IMGCNT_MTX_CRR_ADJ0	R/W	H'0080 0000	<VDCE1_base> + 14B4	32
VDCE1	Image quality adjustment block matrix CRR adjustment register 1	IMGCNT_MTX_CRR_ADJ1	R/W	H'0000 0100	<VDCE1_base> + 14B8	32
VDCE1	Dynamic range compression register	IMGCNT_DRC_REG	R/W	H'0000 0000	<VDCE1_base> + 14C0	32

Table 38.44 Register Configuration for data enable input (Channel 0)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Video input with data enable control register 0	DEMODE0	R/W	H'0001 20A0	<VDCE0_base> + 1B3C	32
VDCE0	Video input with data enable control register 1	DEMODE1	R/W	H'0001 20A0	<VDCE0_base> + 1B40	32

Table 38.45 Register Configuration for data enable input (Channel 1)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Video input with data enable control register 0	DEMODE0	R/W	H'0001 20A0	<VDCE1_base> + 1B3C	32
VDCE1	Video input with data enable control register 1	DEMODE1	R/W	H'0001 20A0	<VDCE1_base> + 1B40	32

NOTE

Register access sizes other than defined in the table above are not supported.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

38.3.2.1 External Input Block Register Update Control Register (INP_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	INP_EXT_UPDATE	—	—	—	INP_IMG_UPDATE
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INP_EXT_UPDATE	0	R/WC1	External Input Block Register Update 0: Registers are not updated. 1: Registers are updated.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INP_IMG_UPDATE	0	R/WC1	Sync Signal Adjustment Block Register Update 0: Registers are not updated. 1: Registers are updated.

38.3.2.2 Input Select Control Register (INP_SEL_CNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	Caution *	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	INP_FORMAT[2:0]			—	—	—	INP_PXD_EDGE	—	—	—	INP_VS_EDGE	—	—	—	INP_HS_EDGE
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

CAUTION

The initial value “0” of bit 20 must be changed to “1”.

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	Bit20	0	R/W	**CAUTION** The initial value “0” of this bit must be changed to “1”.
19 to 15	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
14 to 12	INP_FORMAT[2:0]	0	R/W	External Input Format Select 0: YcbCr444, RGB888 1: RGB666 2: RGB565 3: BT656 4: BT601 5: YCbCr422 6, 7: Setting prohibited
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	INP_PXD_EDGE	0	R/W	Clock Edge Select for Capturing External Input Video Image Signals DV_DATA23 to DV_DATA0 0: Rising edge 1: Falling edge
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INP_VS_EDGE	0	R/W	Clock Edge Select for Capturing External Input Vsync Signals DV_VSYNC 0: Rising edge 1: Falling edge
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INP_HS_EDGE	0	R/W	Clock Edge Select for Capturing External Input Hsync Signals DV_HSYNC 0: Rising edge 1: Falling edge

Note: INP_FORMAT[2:0], INP_PXD_EDGE, INP_VS_EDGE, and INP_HS_EDGE are updated when the INP_EXT_UPDATE bit in INP_UPDATE is 1. INP_SEL is updated when set.

38.3.2.3 External Input Sync Signal Control Register (INP_EXT_SYNC_CNT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	INP_ENDIAN_ON	—	—	—	INP_SWAP_ON	—	—	—	INP_VS_INV	—	—	—	INP_HS_INV
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	INP_H_EDGE_SEL	—	—	—	INP_F525_625	—	—	INP_H_POS[1:0]	
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	INP_ENDIAN_ON	0	R/W	External Input Bit Endian Change On/Off Control 0: Off 1: On
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	INP_SWAP_ON	0	R/W	External Input B/R Signal Swap On/Off Control 0: Off 1: On
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	INP_VS_INV	0	R/W	External Input Vsync Signal DV_VSYNC Inversion Control 0: Not inverted (positive polarity) 1: Inverted (negative polarity)
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	INP_HS_INV	0	R/W	External Input Hsync Signal DV_HSYNC Inversion Control 0: Not inverted (positive polarity) 1: Inverted (negative polarity)
15 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	INP_H_EDGE_SEL	0	R/W	Reference Select for External Input BT656 Hsync Signal 0: EAV 1: SAV
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INP_F525_625	0	R/W	Number of Lines for BT656 External Input 0: 525 lines 1: 625 lines
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	INP_H_POS[1:0]	0	R/W	Y/Cb/Y/Cr Data String Start Timing to Hsync Reference for BT656/601 or YCbCr422 External Input 0: Cb/Y/Cr/Y(BT656/601), Cb/Cr (YCbCr422) 1: Y/Cr/Y/Cb(BT656/601), setting prohibited (YCbCr422) 2: Cr/Y/Cb/Y(BT656/601), setting prohibited (YCbCr422) 3: Y/Cb/Y/Cr(BT656/601), Cr/Cb (YCbCr422)

Note: This register is updated when the INP_EXT_UPDATE bit in INP_UPDATE is 1.

38.3.2.4 Vsync Signal Phase Adjustment Register (INP_VSYNC_PH_ADJ)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	INP_FH50[9:0]									
Initial Value:	0	0	0	0	0	0	1	1	0	1	0	1	1	0	1	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	INP_FH25[9:0]									
Initial Value:	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	1
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	INP_FH50 [9:0]	858	R/W	Vsync Signal 1/2fH Phase Timing 1/2 clock cycle of the horizontal cycle should be set.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	INP_FH25 [9:0]	429	R/W	Vsync Signal 1/4fH Phase Timing 1/4 clock cycle of the horizontal cycle should be set.

Note: The INP_FH50[9:0] bits are updated when the INP_EXT_UPDATE and INP_IMG_UPDATE bits in INP_UPDATE are 1. The IMP_FH25[9:0] bits are updated when the INP_IMG_UPDATE bit is 1.

38.3.2.5 Sync Signal Delay Adjustment Register (INP_DLY_ADJ)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	INP_VS_DLY_L[2:0]			INP_FLD_DLY[7:0]							
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	INP_VS_DLY[7:0]								INP_HS_DLY[7:0]							
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 24	INP_VS_ DLY_L[2:0]	0	R/W	Number of lines for Delaying Vsync signal and Field Differentiation Signal Delay amount: 0 to 7 (lines)
23 to 16	INP_FLD_ DLY[7:0]	0	R/W	Field Differentiation Signal Delay Amount Delay amount: 0 to 254 (clock cycles)
15 to 8	INP_VS_ DLY[7:0]	0	R/W	Vsync Signal Delay Amount Delay amount: 0 to 254 (clock cycles)
7 to 0	INP_HS_ DLY[7:0]	0	R/W	Hsync Signal Delay Amount Delay amount: 0 to 254 (clock cycles)

Note: This register is updated when the INP_IMG_UPDATE bit in INP_UPDATE is 1.

38.3.2.6 Image Quality Adjustment Block Register Update Control Register (IMGCNT_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IMGCNT_VEN
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	IMGCNT_VEN	0	R/WC1	Image Quality Adjustment Block Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync signal.

38.3.2.7 Image Quality Adjustment Block Matrix Mode Register (IMGCNT_MTX_MODE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IMGCNT_MTX_MD[1:0]	
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	IMGCNT_MTX_MD[1:0]	3	R/W	Operating Mode 0: GBR → GBR 1: GBR → YCbCr 2: YCbCr → GBR 3: YCbCr → YCbCr

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

38.3.2.8 Image Quality Adjustment Block Matrix YG Adjustment Register 0 (IMGCNT_MTX_YG_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	IMGCNT_MTX_YG[7:0]							
Initial Value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IMGCNT_MTX_GG[10:0]										
Initial Value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	IMGCNT_ MTX_YG[7:0]	128	R/W	Offset (DC) Adjustment of Y/G Signal Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	IMGCNT_ MTX_GG[10:0]	256	R/W	Y/G Signal Gain Adjustment for Y/G Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

38.3.2.9 Image Quality Adjustment Block Matrix YG Adjustment Register 1 (IMGCNT_MTX_YG_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	IMGCNT_MTX_GB[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IMGCNT_MTX_GR[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	IMGCNT_ MTX_GB [10:0]	0	R/W	Cb/B Signal Gain Adjustment for Y/G Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	IMGCNT_ MTX_GR [10:0]	0	R/W	Cr/R Signal Gain Adjustment for Y/G Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

38.3.2.10 Image Quality Adjustment Block Matrix CBB Adjustment Register 0 (IMGCNT_MTX_CBB_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	IMGCNT_MTX_B[7:0]							
Initial Value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IMGCNT_MTX_BG[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	IMGCNT_ MTX_B[7:0]	128	R/W	Offset (DC) Adjustment of Cb/B Signal Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	IMGCNT_ MTX_BG[10:0]	0	R/W	Y/G Signal Gain Adjustment for Cb/B Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

38.3.2.11 Image Quality Adjustment Block Matrix CBB Adjustment Register 1 (IMGCNT_MTX_CBB_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	IMGCNT_MTX_BB[10:0]										
Initial Value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IMGCNT_MTX_BR[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	IMGCNT_ MTX_BB [10:0]	256	R/W	Cb/B Signal Gain Adjustment for Cb/B Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	IMGCNT_ MTX_BR [10:0]	0	R/W	Cr/R Signal Gain Adjustment for Cb/B Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

38.3.2.12 Image Quality Adjustment Block Matrix CRR Adjustment Register 0 (IMGCNT_MTX_CRR_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	IMGCNT_MTX_R[7:0]							
Initial Value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IMGCNT_MTX_RG[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	IMGCNT_ MTX_R[7:0]	128	R/W	Offset (DC) Adjustment of Cr/R Signal Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	IMGCNT_ MTX_RG[10:0]	0	R/W	Y/G Signal Gain Adjustment for Cr/R Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

38.3.2.13 Image Quality Adjustment Block Matrix CRR Adjustment Register 1 (IMGCNT_MTX_CRR_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	IMGCNT_MTX_RB[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	IMGCNT_MTX_RR[10:0]										
Initial Value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	IMGCNT_ MTX_RB[10:0]	0	R/W	Cb/B Signal Gain Adjustment for Cr/R Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	IMGCNT_ MTX_RR[10:0]	256	R/W	Cr/R Signal Gain Adjustment for Cr/R Signal Output Signed (two's complement) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when the IMGCNT_VEN bit in IMGCNT_UPDATE is 1.

38.3.2.14 Dynamic Range Compression Register (IMGCNT_DRC_REG)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRC_EN
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	DRC_EN	0	R/W	Use of Contrast Correction (Dynamic Range Compression) 0: Does not use contrast correction 1: Uses contrast correction

38.3.2.15 Video input with data enable control register 0 (DEMODE0)

Following functions are added for video input signal use-case expansion of the VDCE Video Input.

1. Vertical position can be shifted before 4 lines from the timing configured by other registers.
2. Starting timing to capture video data by DV_HSYNC can be shifted before 1 DV_CLK.
3. Interrupt timing can be changed to the end of the final line by additional line number comparator.

Additional functions on VDCE are controlled by the DEMODE0 and DEMODE1 registers.

CAUTION

Add setting of DEMODE0 to the initialization routine.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	DE_4HS_EN	DE_HSBYPS_EN	DE_VLAST_EN	—	—
Initial Value:	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	See above	R	Reserved When read, an undefined value is returned. The write value must always be 0.
4	DE_4HS_EN	0	R/W	Vertical position is shifted before 4 lines from the timing configured at other registers. 1: enable 0: disable
3	DE_HSBYPS_EN	0	R/W	Starting timing to capture video data by DV_HSYNC is shifted before 1 DV_CLK. 1: enable 0: disable
2	DE_VLAST_EN	0	R/W	Interrupt timing changes to end of the final line by using setting of DEMODE1. 1: enable 0: disable
1 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value must always be 0.

CAUTION

Reserved bits of DEMODE0 must be set to 0.

38.3.2.16 Video input with data enable control register 1 (DEMODE1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DE_VLAST[10:0]										
Initial Value:	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 11	—	See above	R	Reserved When read, an undefined value is returned. The write value should always be 0.
10 to 0	DE_ VLAST[10:0]	$0A0_H$	R/W	Active vertical line number Line number must be set by 4 line alignment, so Bit 1 and Bit 0 of this register must be set to 0. When DE_VLAST_EN and DE_4HS_EN is 1, this register is available. set the last line number to this register.

38.3.3 Usage Methods

38.3.3.1 Input Format Adjustment Method

Setting examples of each input format are shown below.

Table 38.46 External Input (BT656, 525i) Setting Example

Register Name	Bit Name	Description	Setting Value
INP_SEL_CNT	Bit20	Set this bit to “1”.	1
INP_SEL_CNT	INP_FORMAT[2:0]	Selects the external input format.	3
INP_SEL_CNT	INP_PXD_EDGE	Selects the clock edge for capturing the external input video signals.	0
INP_SEL_CNT	INP_VS_EDGE	Selects the clock edge for capturing the external input Vsync signals.	0
INP_SEL_CNT	INP_HS_EDGE	Selects the clock edge for capturing the external input Hsync signals.	0
INP_EXT_SYNC_CNT	INP_ENDIAN_ON	Changes the bit endian of the external input.	0
INP_EXT_SYNC_CNT	INP_SWAP_ON	Enables or disables the B/R signal swap of the external input.	0
INP_EXT_SYNC_CNT	INP_HS_INV	Enables or disables the Hsync signal inversion of the external input.	1
INP_EXT_SYNC_CNT	INP_VS_INV	Enables or disables the Vsync signal inversion of the external input.	1
INP_EXT_SYNC_CNT	INP_H_EDGE_SEL	Selects the Hsync reference for BT656 input.	0
INP_EXT_SYNC_CNT	INP_F525_625	Sets the number of lines for BT656 input.	0
INP_EXT_SYNC_CNT	INP_H_POS[1:0]	Sets the data start timing with respect to the Hsync in the BT656/601 format.	0
INP_VSYNC_PH_ADJ	INP_FH50[9:0]	Sets the 1/2fH phase in clock cycle units.	858
INP_VSYNC_PH_ADJ	INP_FH25[9:0]	Sets the 1/4fH phase in clock cycle units.	429
INP_DLY_ADJ	INP_VS_DLY_L[2:0]	Sets the number of lines for delaying the Vsync signal and field differentiation signal.	0
INP_DLY_ADJ	INP_VS_DLY[7:0]	Sets the amount of delay of the Vsync signal in clock cycle units.	0
INP_DLY_ADJ	INP_HS_DLY[7:0]	Sets the amount of delay of the Hsync signal in clock cycle units.	0
INP_DLY_ADJ	INP_FLD_DLY[7:0]	Sets the amount of delay of the field differentiation signal in clock units.	0

Note: Some registers require, after they are set, that the INP_EXT_UPDATE and INP_IMG_UPDATE bits in INP_UPDATE should be set to 1.

Table 38.47 External Input (BT601, 525i) Setting Example

Register Name	Bit Name	Description	Setting Value
INP_SEL_CNT	Bit20	Set this bit to “1”.	1
INP_SEL_CNT	INP_FORMAT[2:0]	Selects the external input format.	4
INP_SEL_CNT	INP_PXD_EDGE	Selects the clock edge for capturing the external input video signals.	0
INP_SEL_CNT	INP_VS_EDGE	Selects the clock edge for capturing the external input Vsync signals.	0
INP_SEL_CNT	INP_HS_EDGE	Selects the clock edge for capturing the external input Hsync signals.	0
INP_EXT_SYNC_CNT	INP_ENDIAN_ON	Changes the bit endian of the external input.	0
INP_EXT_SYNC_CNT	INP_SWAP_ON	Enables or disables the B/R signal swap of the external input.	0
INP_EXT_SYNC_CNT	INP_HS_INV	Enables or disables the Hsync signal inversion of the external input.	1
INP_EXT_SYNC_CNT	INP_VS_INV	Enables or disables the Vsync signal inversion of the external input.	1
INP_EXT_SYNC_CNT	INP_H_EDGE_SEL	Selects the Hsync reference for BT656 input.	0
INP_EXT_SYNC_CNT	INP_F525_625	Sets the number of lines for BT656 input.	0
INP_EXT_SYNC_CNT	INP_H_POS[1:0]	Sets the data start timing with respect to the Hsync in the BT656/601 format.	0
INP_VSYNC_PH_ADJ	INP_FH50[9:0]	Sets the 1/2fH phase in clock cycle units.	858
INP_VSYNC_PH_ADJ	INP_FH25[9:0]	Sets the 1/4fH phase in clock cycle units.	429
INP_DLY_ADJ	INP_VS_DLY_L[2:0]	Sets the number of lines for delaying the Vsync signal and field differentiation signal.	0
INP_DLY_ADJ	INP_VS_DLY[7:0]	Sets the amount of delay of the Vsync signal in clock cycle units.	0
INP_DLY_ADJ	INP_HS_DLY[7:0]	Sets the amount of delay of the Hsync signal in clock cycle units.	0
INP_DLY_ADJ	INP_FLD_DLY[7:0]	Sets the amount of delay of the field differentiation signal in clock units.	0

Note: Some registers require, after they are set, that the INP_EXT_UPDATE and INP_IMG_UPDATE bits in INP_UPDATE should be set to 1.

38.3.3.2 Usage Method of Conversion Color Matrix

Typical data conversion setting examples are shown below.

Table 38.48 Conversion Color Matrix

Register Name	Bit Name	GBR to GBR	GBR to YCbCr	YCbCr to GBR	YCbCr to YCbCr
IMGCNT_MTX_MODE	IMGCNT_MTX_MD[1:0]	0	1	2	3
IMGCNT_MTX_YG_ADJ0	IMGCNT_MTX_YG[7:0]	128	128	128	128
IMGCNT_MTX_YG_ADJ0	IMGCNT_MTX_GG[10:0]	256	150	256	256
IMGCNT_MTX_YG_ADJ1	IMGCNT_MTX_GB[10:0]	0	29	1960	0
IMGCNT_MTX_YG_ADJ1	IMGCNT_MTX_GR[10:0]	0	77	1865	0
IMGCNT_MTX_CBB_ADJ0	IMGCNT_MTX_B[7:0]	128	128	128	128
IMGCNT_MTX_CBB_ADJ0	IMGCNT_MTX_BG[10:0]	0	1963	256	0
IMGCNT_MTX_CBB_ADJ1	IMGCNT_MTX_BB[10:0]	256	128	454	256
IMGCNT_MTX_CBB_ADJ1	IMGCNT_MTX_BR[10:0]	0	2005	0	0
IMGCNT_MTX_CRR_ADJ0	IMGCNT_MTX_R[7:0]	128	128	128	128
IMGCNT_MTX_CRR_ADJ0	IMGCNT_MTX_RG[10:0]	0	1941	256	0
IMGCNT_MTX_CRR_ADJ1	IMGCNT_MTX_RB[10:0]	0	2027	0	0
IMGCNT_MTX_CRR_ADJ1	IMGCNT_MTX_RR[10:0]	256	128	359	256

Note: The registers require, after they are set, that the IMGCNT_VEN bit in IMGCNT_UPDATE should be set to 1.

38.4 Scaler

NOTE

The D1L2(H) devices do not feature the video input and down/up scaling functions. It can only read unscaled image data into Graphic Layers 0 and 1.

38.4.1 Scaler Functions

38.4.1.1 Overview of Functions

The scaler subjects the YCbCr and RGB signals output from the input controller, to sync signal generation; and reduction and enlargement of the images.

The scaler also records video image in the frame buffer.

The Video Display Controller has two scalers (scalers 0 and 1) in each channel. For image blending, either cascading the scalers or using the VIN synthesizer can be selected. When the scalers are cascaded for image blending, the image from scaler 0 is placed in the lower layer and that from scaler 1 is placed in the upper layer.

Scaler 0 receives a signal from the input controller in the channel where scaler 0 belongs.

For the VIN synthesizer, refer to Section 38.6, Image Synthesizer.

In scalers 0 and 1, either enlargement process or graphics 0 or 1 process can be used at a time.

The functional block diagrams of scalers 0 and 1 are shown below.

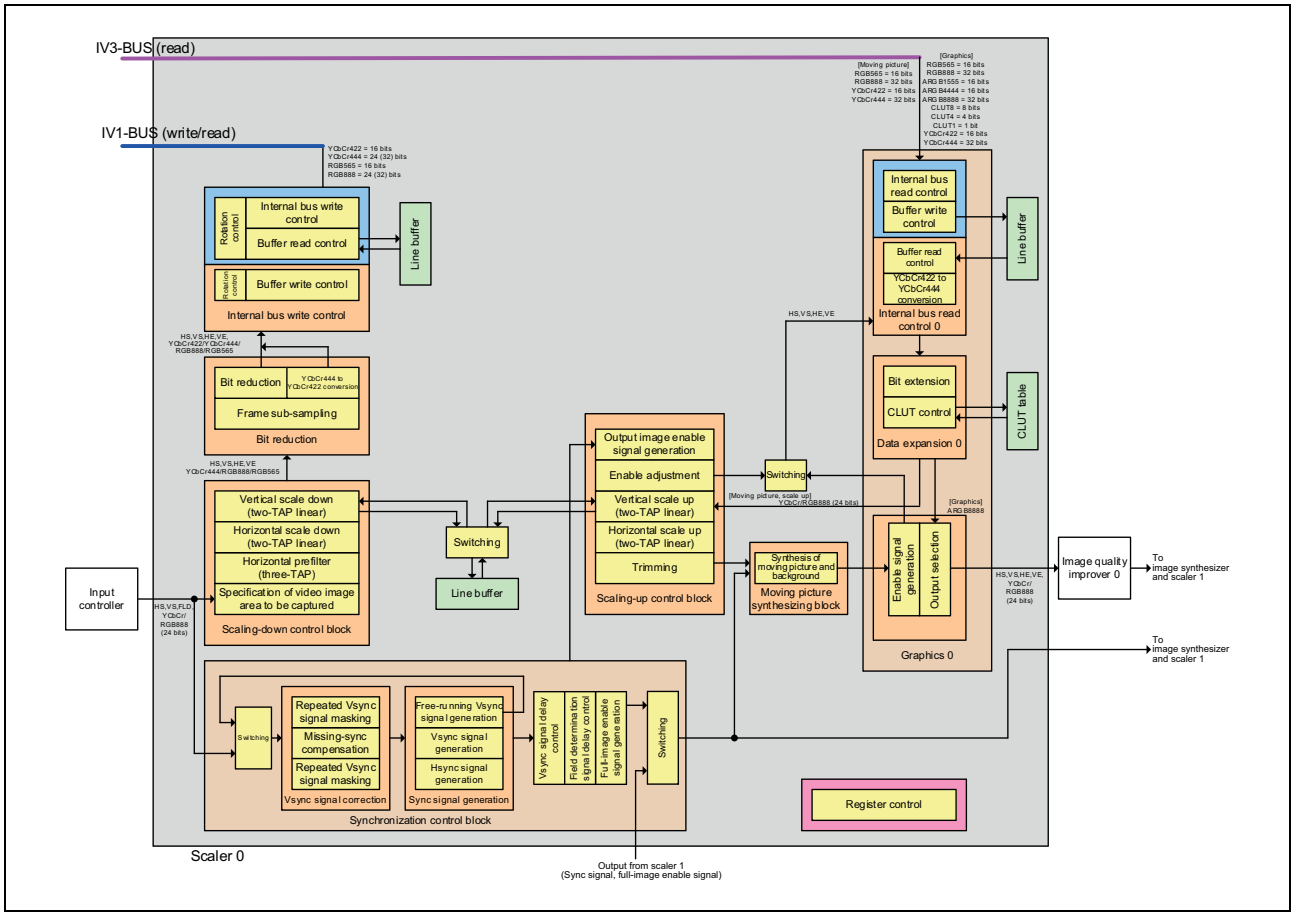


Figure 38.24 Functional Block Diagram of Scaler 0

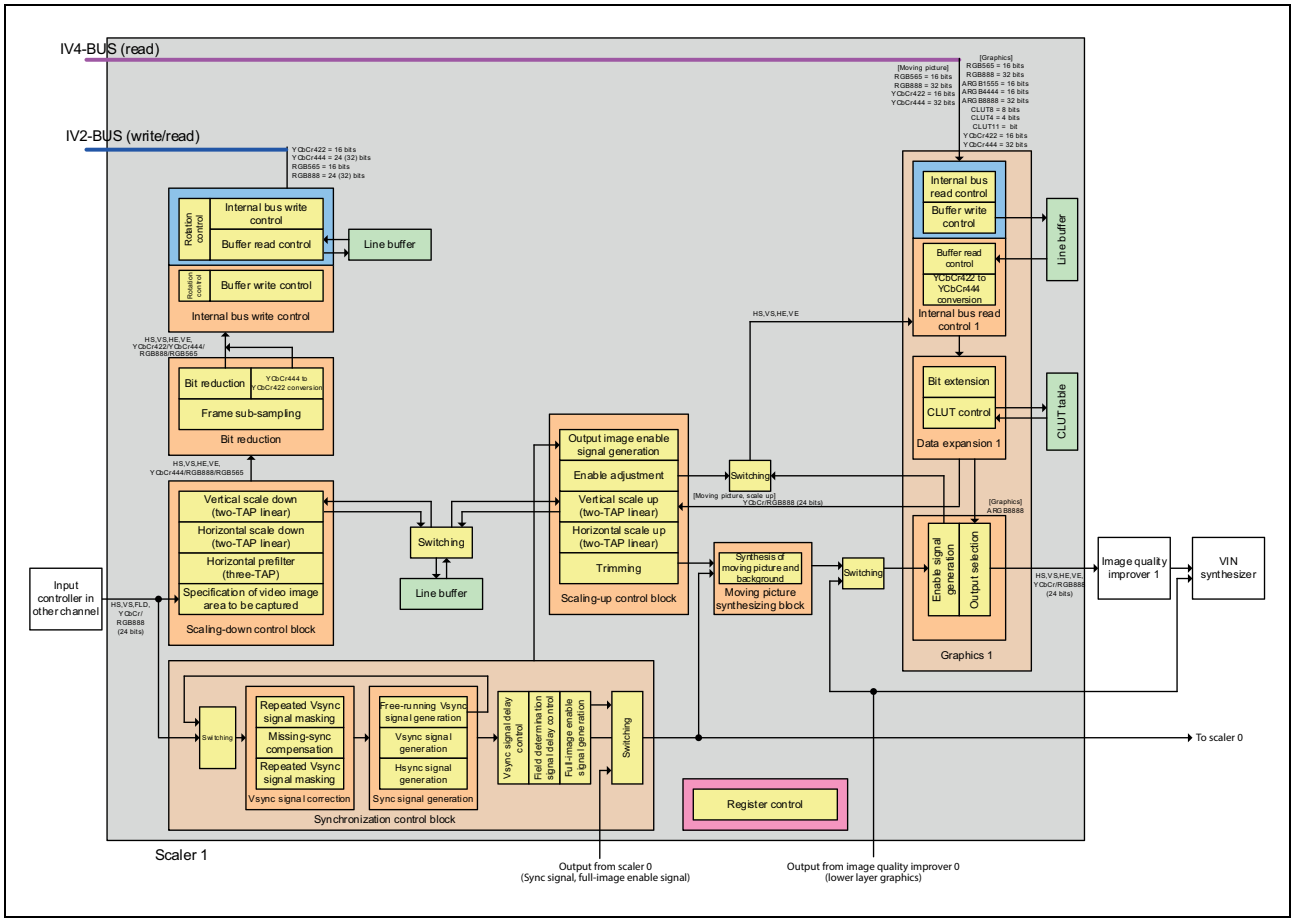


Figure 38.25 Functional Block Diagram of Scaler 1

The video display controller has two scalers and each scaler has a graphics block. The registers and bits in the scalers are named SC0_xxxx or SC1_xxxx and those in the graphics blocks are named GR0_xxxx or GR1_xxxx, but in this section, they are collectively called SC_xxxx or GR_xxxx.

38.4.1.2 Register Control

(1) Updating Registers

The Vsync signal is used to control the update timing of all the registers of the scaling and graphics blocks except some registers of the sync control block and some of the other blocks.

After 1 is set to the bits in the update control register, the contents of the relevant registers are modified at the rising edge of the Vsync signal. The update control register is automatically cleared to 0 after the modification.

Table 38.49 Register Update Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_UPDATE	SC_SCL0_UPDATE	0	SYNC Control Register Update 0: Registers are not updated. 1: Registers are updated.
SC_SCL0_UPDATE	SC_SCL0_VEN_D	0	Scaling-Up Control and Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
SC_SCL0_UPDATE	SC_SCL0_VEN_C	0	Scaling-Down Control and Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
SC_SCL0_UPDATE	SC_SCL0_VEN_B	0	Synchronization Control and Scaling-up Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
SC_SCL0_UPDATE	SC_SCL0_VEN_A	0	Scaling-Down Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
SC_SCL1_UPDATE	SC_SCL1_UPDATE_B	0	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated.
SC_SCL1_UPDATE	SC_SCL1_UPDATE_A	0	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated.
SC_SCL1_UPDATE	SC_SCL1_VEN_B	0	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
SC_SCL1_UPDATE	SC_SCL1_VEN_A	0	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GR_UPDATE	GR_UPDATE	0	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated.
GR_UPDATE	GR_P_VEN	0	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GR_UPDATE	GR_IBUS_VEN	0	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

The registers controlled by SC_SCL0_VEN_A, SC_SCL0_VEN_C, SC_SCL1_VEN_A, and SC_SCL1_VEN_B are modified at the rising edge of the input Vsync signal.

The registers controlled by SC_SCL0_VEN_B, SC_SCL0_VEN_D, GR_P_VEN, and GR_IBUS_VEN are modified at the rising edge of the output Vsync signal.

38.4.1.3 Synchronization Control

(1) Selecting Vsync Signal

The Vsync signal to be output from the scaler can be selected.

When an external input signal is to be displayed, an external input Vsync signal should be selected to be output.

When an external input signal is not provided, a free-running Vsync signal should be selected to be output.

Table 38.50 Vsync Signal Selection Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC3	SC_RES_VS_SEL	1	Vsync Signal Output Select 0: External input Vsync signal 1: Internally generated free-running Vsync signal

(2) Masking Repeated Vsync Signals

It is possible to prevent receiving the Vsync signal with a period shorter than the standard period. This is achieved by setting the start timing to receive the next Vsync signal after receiving an input Vsync signal.

The Vsync signal reception masking period is set with the SC_RES_VMASK[15:0] bits.

Masking period [usec] = SC_RES_VMASK × 128 ÷ pixel clock frequency [MHz]

This function is enabled or disabled by the SC_RES_VMASK_ON bit.

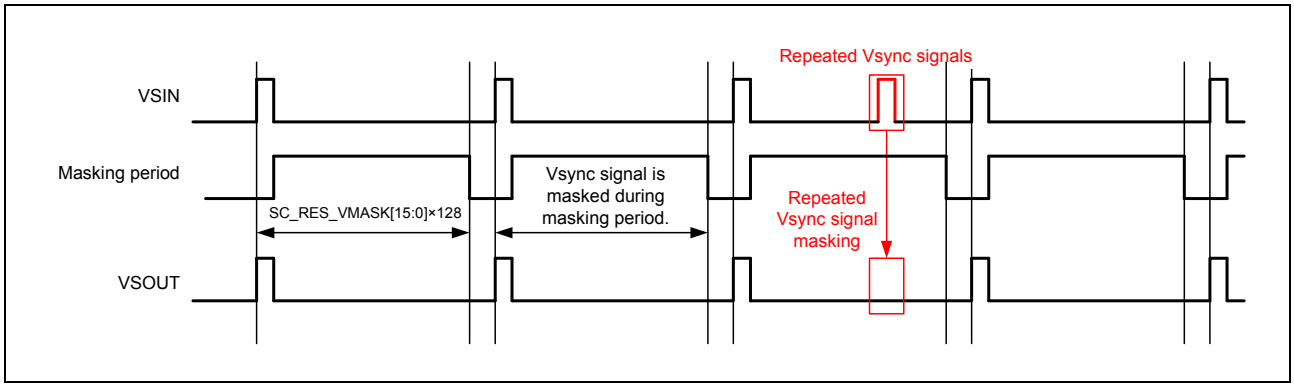


Figure 38.26 Timing for Masking Repeated Vsync Signals

Table 38.51 Repeated Vsync Signal Mask Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC1	SC_RES_VMASK_ON	1	Repeated Vsync Signal Masking Control 0: Repeated Vsync signal masking control is disabled. 1: Repeated Vsync signal masking control is enabled.
SC_SCL0_FRC1	SC_RES_VMASK[15:0]	2800	Repeated Vsync Signal Masking Period Sets the repeated Vsync signal masking period beginning at a Vsync signal in terms of 128 pixel-clock periods. Masking period [usec] = SC_RES_VMASK × 128 ÷ pixel clock frequency [MHz]

(3) Compensating for Missing Vsync Signals

It is possible to prevent output of the Vsync signal with a period longer than the standard period. This is achieved by setting the wait time after reception of an input Vsync signal until reception of the next Vsync signal.

If no Vsync signals are received during the wait time, an internally generated sync signal is inserted.

The wait time can be set using the SC_RES_VLACK[15:0] bits.

Wait time [usec] = SC_RES_VLACK × 128 ÷ pixel clock frequency [MHz]

This function is enabled or disabled by the SC_RES_VLACK_ON bit.

If no Vsync signals are input during the Vsync signal reception time, the SC_RES_QVLACK bit is set to the high level.

If Vsync signals are continuously detected four or more times during the Vsync signal reception time, the SC_RES_QVLOCK bit is set to the high level.

The SC_RES_QVLOCK bit is valid even when both the SC_RES_VMASK_ON and SC_RES_VLACK_ON bits are set to turn off the corresponding functions.

Note that, however, the SC_RES_VMASK and SC_RES_VLACK bits must be set correctly.

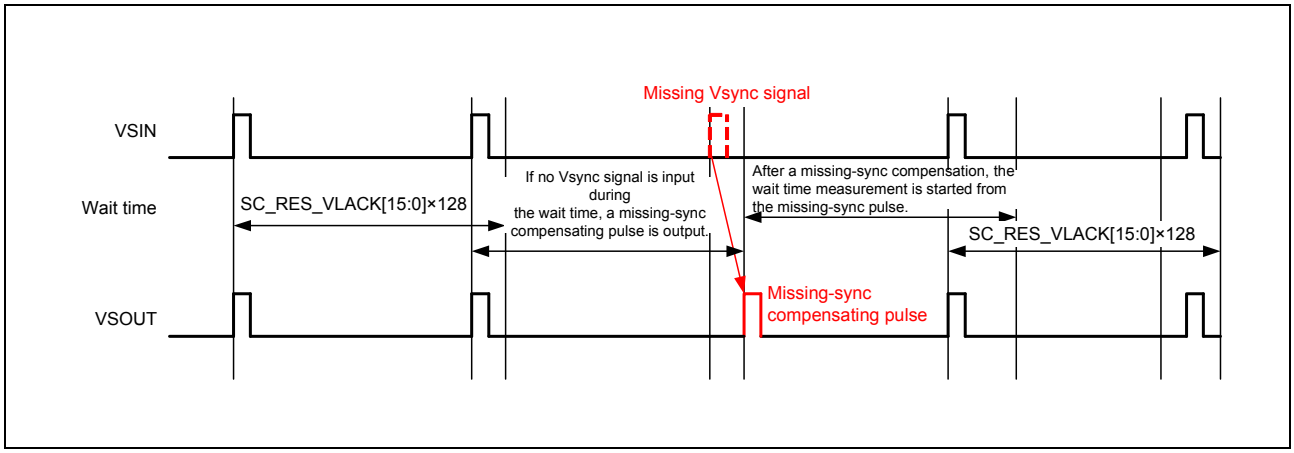


Figure 38.27 Compensation of Missing Vsync Signals

Table 38.52 Missing Vsync Compensation Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC2	SC_RES_VLACK_ON	1	Missing Vsync Signal Compensation 0: Compensation of missing Vsync signals is disabled. 1: Compensation of missing Vsync signals is enabled.
SC_SCL0_FRC2	SC_RES_VLACK[15:0]	3600	Missing-Sync Compensating Pulse Output Wait Time Sets the wait time before outputting a missing-sync compensating pulse after a Vsync signal. Wait time [usec] = SC_RES_VLACK × 128 ÷ pixel clock frequency [MHz]
SC_SCL0_FRC9	SC_RES_QVLACK	—	Missing Vsync Signal Detection Flag 1: Missing Vsync signal input has been detected. 0: No missing Vsync signal input has been detected.
SC_SCL0_FRC9	SC_RES_QVLOCK	—	Locked Vsync Signal Detection Flag 1: Repeated and missing Vsync signal input has not been detected for four or more vertical periods. 0: Repeated or missing Vsync signal input has been detected.

For the Vsync signal, repeated-signal masking is first carried out and then missing-signal compensation is carried out, followed by another repeated-signal masking.

Repetition masking is inserted after missing-Vsync compensation to prevent output of the Vsync signal even in cases such as the input of a Vsync signal immediately after the input of a pulse to compensate for a missing Vsync signal.

On/off control of the missing-Vsync compensation also applies to the second repeated-Vsync masking; and masking period setting of the first repeated-Vsync masking also applies to the second repeated-Vsync masking.

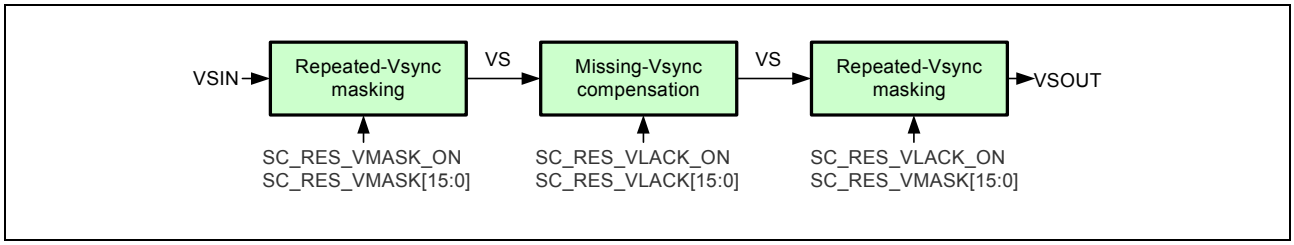


Figure 38.28 Repeated-Vsync Masking and Missing-Vsync Compensation

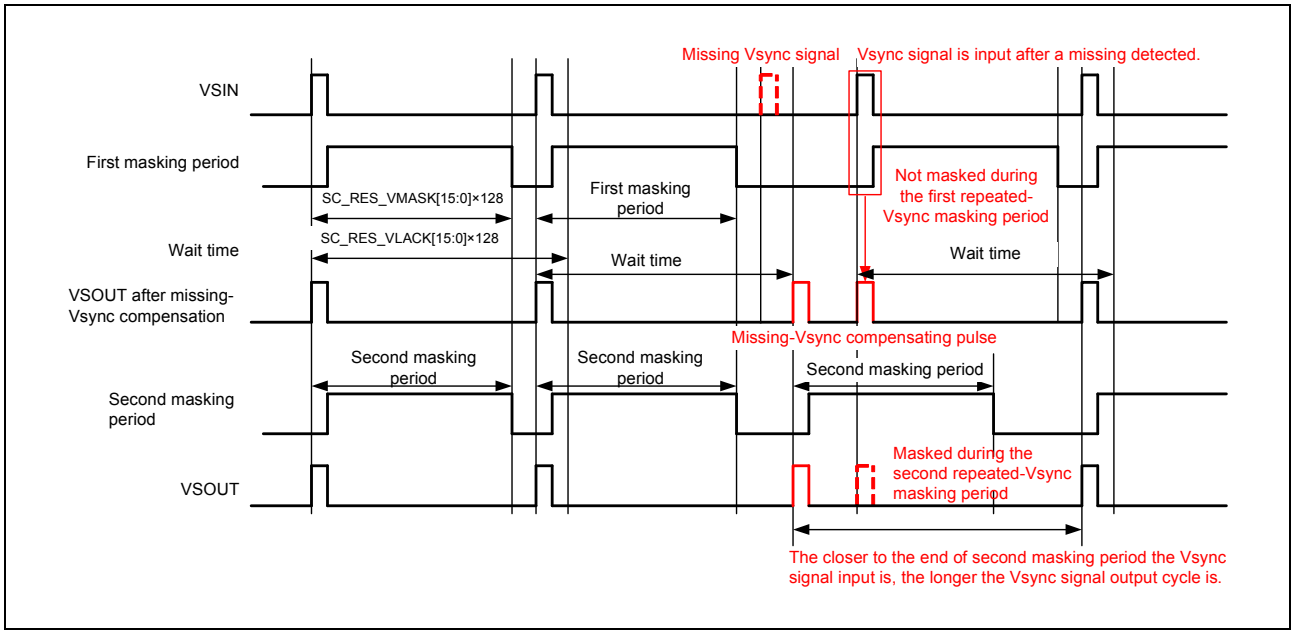


Figure 38.29 Timing for Masking Repeated Vsync Signals and Missing Vsync Signal Compensation

(4) Free-Running Period

Free-running Vsync and Hsync periods can be set.

Hsync period [usec] = (SC_RES_FH + 1) ÷ pixel clock frequency [MHz]

Vsync period [usec] = horizontal period [usec] × (SC_RES_FV + 1)

Table 38.53 Free-Running Period Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC4	SC_RES_FV[10:0]	524	Free-Running Vsync Period Setting Free-running Vsync period = (SC_RES_FV + 1) × horizontal period [usec]
SC_SCL0_FRC4	SC_RES_FH[10:0]	799	Hsync Period Setting Hsync period [usec] = (SC_RES_FH +1) ÷ pixel clock frequency [MHz]

When selecting an external input Vsync signal, set the SC_RES_VS_SEL bit to 0. At this time, the internally generated free-running Vsync signal is not output.

In the meantime, the Hsync signal is always generated according to the free-running signal setting and output from the scaler.

(5) Vsync Signal Delay Control

Delay of Vsync signal output from the scaler can be controlled.

The delay is used to adjust the frame buffer read timing.

Table 38.54 Vsync Output Delay Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC5	SC_RES_VSDLY[7:0]	1	Vsync Signal Delay Control Adjusts the Vsync signal delay in the output Hsync period units. Vsync signal delay [usec]: SC_RES_VSDLY × output Hsync period [usec]

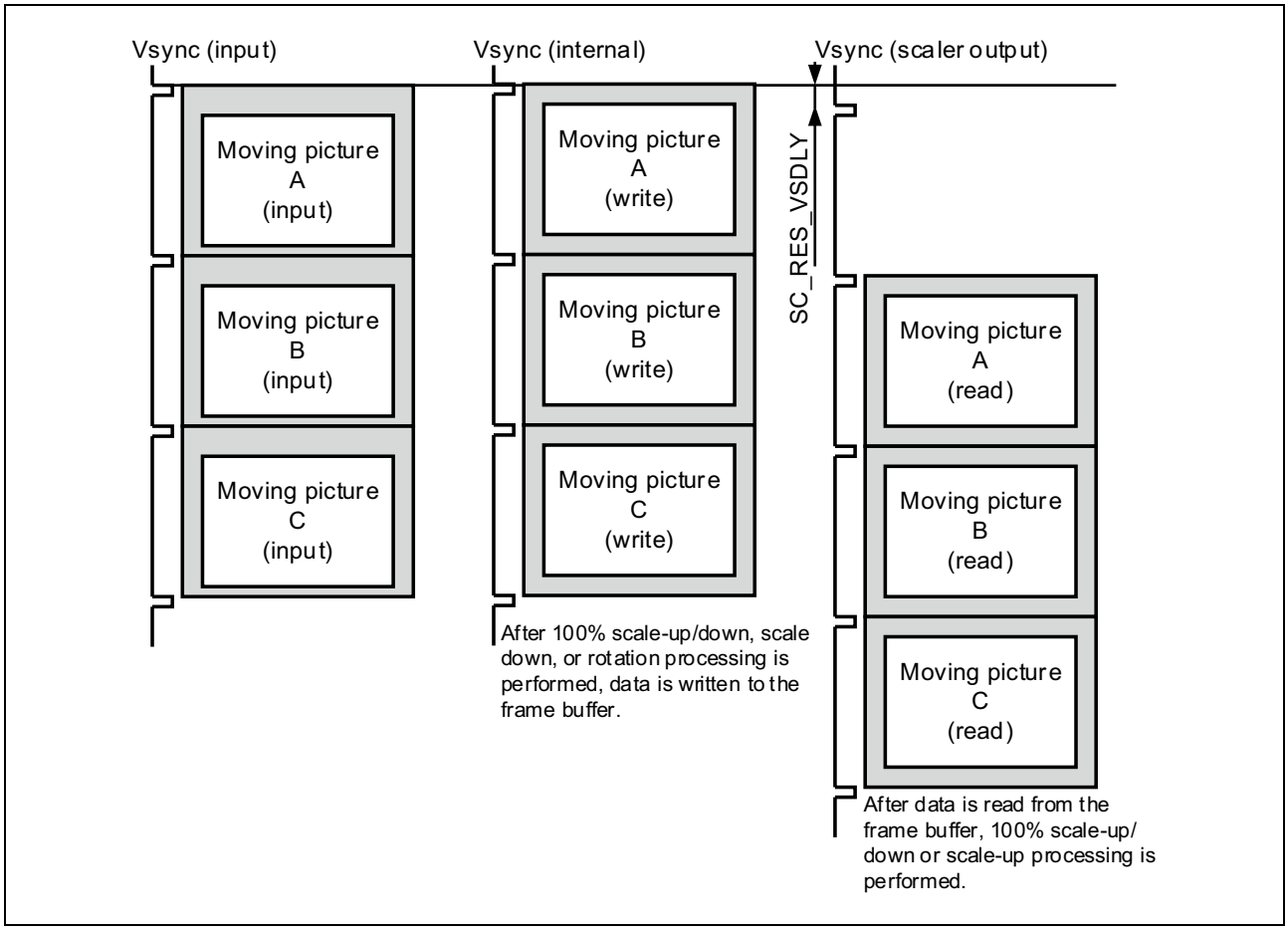


Figure 38.30 Vsync Signal Phases (Two Frame-Buffer Planes Used)

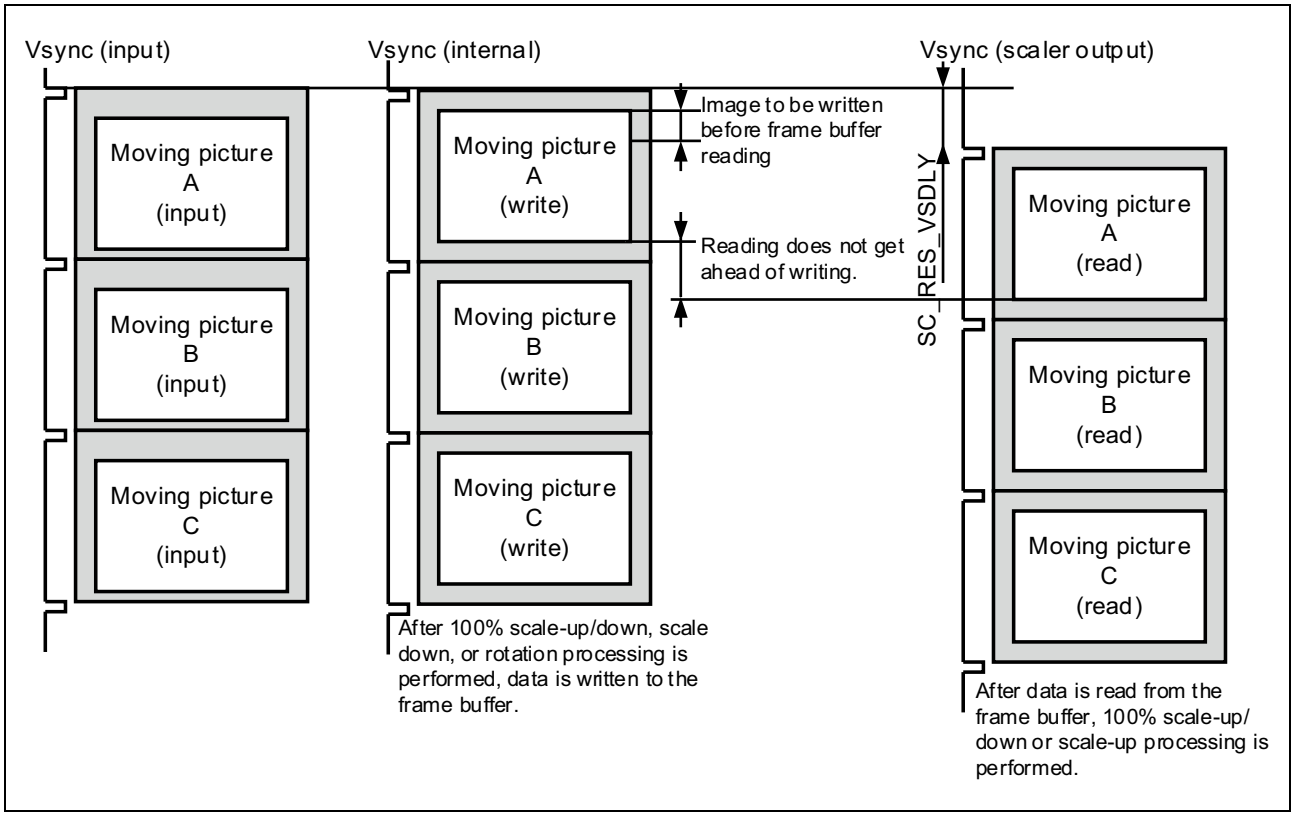


Figure 38.31 Vsync Signal Phases (One Frame-Buffer Plane Used)

38.4.2 Setting Video Input Size

(1) Setting Image Area to be Captured

The image area to be captured can be set for reduction or enlargement.

The area is defined by specifying its start position and width based on the input Hsync and Vsync signals.

Table 38.55 Control of Image Area to be Captured

Register Name	Bit Name	Initial Value	Description
SC_SCL0_DS2	SC_RES_VS[10:0]	18	Vertical Position Setting for Video Signal Capturing (VSYNC + (V backporch -1) lines) Note: The set value should be four or more (lines). SC_RES_VS + SC_RES_VW should be equal to or less than 2039 (lines).
SC_SCL0_DS2	SC_RES_VW[10:0]	240	Vertical Width of Video Signal to be Captured (lines) Note: SC_RES_VS + SC_RES_VW should be equal to or less than 2039 (lines).
SC_SCL0_DS3	SC_RES_HS[10:0]	244	Horizontal Position Setting for Video Signal Capturing (HSYNC + H backporch video-image clock cycles) Note: The set value should be 16 or more (clock cycles). SC_RES_HS + SC_RES_HW should be equal to or less than 2015 (clock cycles).
SC_SCL0_DS3	SC_RES_HW[10:0]	1440	Horizontal Width of Video Signal to be Captured (video-image clock cycles) Note: SC_RES_HS + SC_RES_HW should be equal to or less than 2015 (clock cycles).

(2) Generating a Full-Screen Enable Signal

The valid period of the full screen to be output from the scaler can be set.

The valid period is defined by specifying its start position and width based on the Hsync and Vsync signals output from the scaler.

Table 38.56 Full-Screen Enable Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC6	SC_RES_F_VS[10:0]	35	Vertical Enable Signal Start Position for Full Screen. (VSYNC + V backporch lines) Note: The set value should be four or more (lines). SC_RES_F_VS + SC_RES_F_VW should be equal to or less than 2039 (lines).
SC_SCL0_FRC6	SC_RES_F_VW[10:0]	480	Vertical Enable Signal Width for Full Screen (lines) Note: SC_RES_F_VS + SC_RES_F_VW should be equal to or less than 2039 (lines).
SC_SCL0_FRC7	SC_RES_F_HS[10:0]	144	Horizontal Enable Signal Start Position for Full Screen. (HSYNC + H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). SC_RES_F_HS + SC_RES_F_HW should be equal to or less than 2015 (clock cycles).
SC_SCL0_FRC7	SC_RES_F_HW[10:0]	640	Horizontal Enable Signal Width for Full Screen (pixel-clock cycles) Note: SC_RES_F_HS + SC_RES_F_HW should be equal to or less than 2015 (clock cycles).

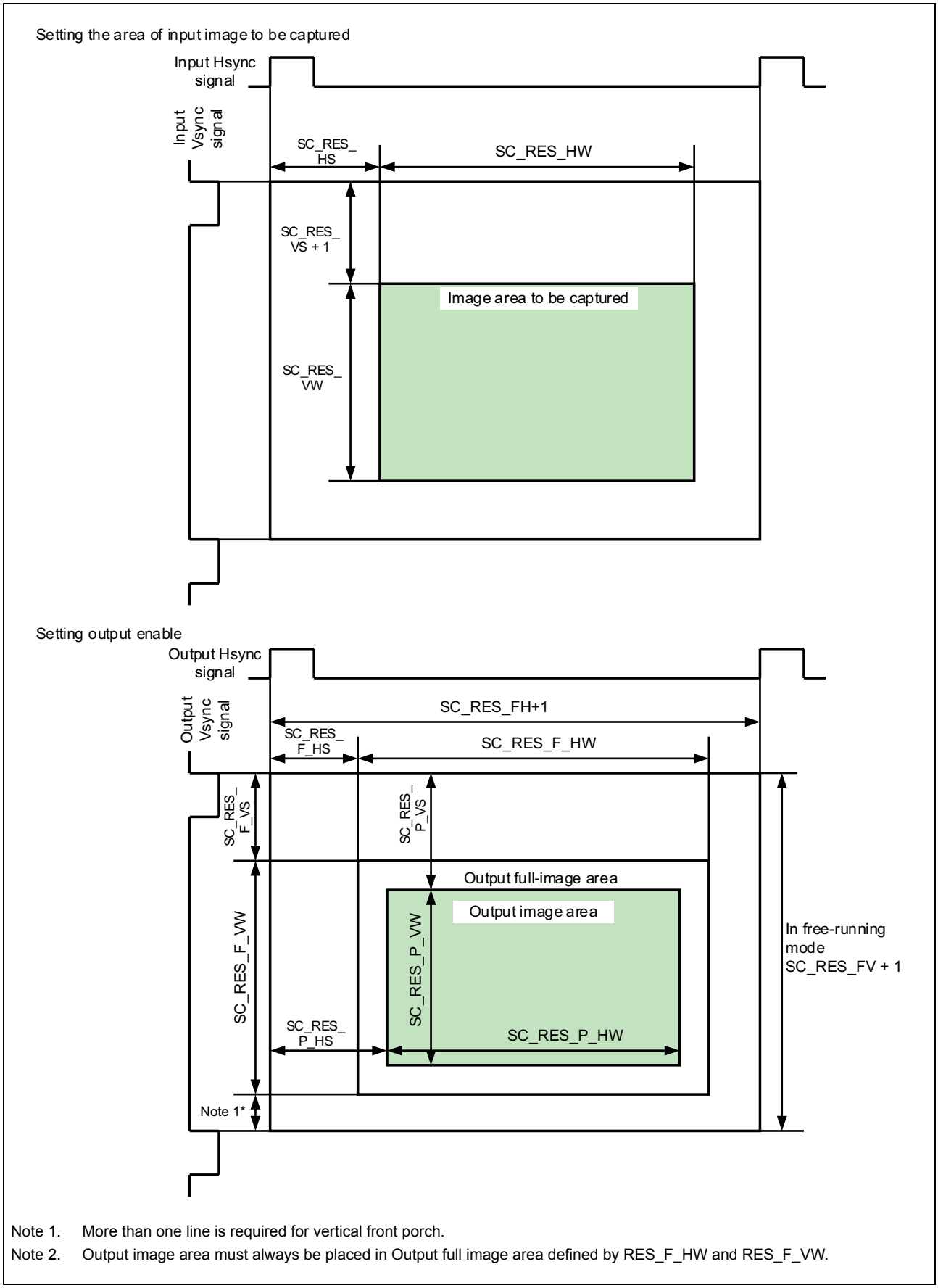
(3) Generating an Image Output Enable Signal

The valid period of the image to be output can be set.

The valid period is defined by specifying its start position and width based on the Hsync and Vsync signals output from the scaler.

Table 38.57 Image Output Enable Controls

Register Name	Bit Name	Initial Value	Description
SC_SCL0_US2	SC_RES_P_VS[10:0]	35	Vertical Enable Signal Start Position for Output Image. (VSYNC + V backporch lines) Note: The set value should be four or more (lines). SC_RES_P_VS + SC_RES_P_VW should be equal to or less than 2039 (lines).
SC_SCL0_US2	SC_RES_P_VW[10:0]	480	Vertical Enable Signal Width for Output Image (lines) Note: SC_RES_P_VS + SC_RES_P_VW should be equal to or less than 2039 (lines).
SC_SCL0_US3	SC_RES_P_HS[10:0]	144	Horizontal Enable Signal Start Position for Output Image. (HSYNC + H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). SC_RES_P_HS + SC_RES_P_HW should be equal to or less than 2015 (clock cycles).
SC_SCL0_US3	SC_RES_P_HW[10:0]	640	Horizontal Enable Signal Width for Output Image (pixel-clock cycles) Note: SC_RES_P_HS + SC_RES_P_HW should be equal to or less than 2015 (clock cycles).



Note 1. More than one line is required for vertical front porch.
Note 2. Output image area must always be placed in Output full image area defined by RES_F_HW and RES_F_VW.

Figure 38.32 Enable Settings

38.4.2.1 Scaling Settings

(1) Scaling Processing Block

The scaling-down control block scales down the input image from the input controller.

Table 38.58 Scaling Process

Mirroring	Horizontal Scaling	Vertical Scaling	Scaling-Down Control Block	Scaling-Up Control Block
Normal	Horizontal scale down	Vertical scale down	Horizontal scale down/ vertical scale down	Horizontal 100% scale up/ vertical 100% scale up
	Horizontal scale down	Vertical scale up	Horizontal scale down/ vertical 100% scale up	Horizontal 100% scale up/ vertical scale up
	Horizontal scale up	Vertical scale down	Horizontal 100% scale up/ vertical scale down	Horizontal scale up/ vertical 100% scale up
	Horizontal scale up	Vertical scale up	Horizontal 100% scale up/ vertical 100% scale up	Horizontal scale up/ vertical scale up
Horizontal mirroring	Horizontal scale down	Vertical scale down	Horizontal scale down/ vertical scale down	Horizontal 100% scale up/ vertical 100% scale up
	Horizontal scale down	Vertical scale up	Horizontal scale down/ vertical 100% scale up	Horizontal 100% scale up/ vertical scale-up
	Horizontal scale up	Vertical scale down	Horizontal 100% scale up/ vertical scale down	Horizontal scale up/ vertical 100% scale up
	Horizontal scale up	Vertical scale up	Horizontal 100% scale up/ vertical 100% scale up	Horizontal scale up/ vertical scale up

It is impossible to use vertical reduction by the scaling-down control block and vertical enlargement by the scaling-up control block simultaneously because they are mutually exclusive.

38.4.2.2 Horizontal Prefilter

The horizontal prefilter can be turned on or off for brightness (Y) and RGB signals to suppress the frequency band of the signals during horizontal size reduction. The input format depends on the SC_RES_MD[1:0] bit setting in the writing mode register (SC_SCL1_WR1).

When the horizontal reduction ratio is high and there is too much folding frequency component to ignore, the horizontal prefilter should be turned on.

Table 38.59 Horizontal Prefilter Settings

Input Format	SC_RES_PFIL_SEL	Operation
YCbCr input	1	Turns on the horizontal prefilter for Y signal and turns off the horizontal prefilter for Cb/Cr signal.
	0	Turns off the horizontal prefilter.
RGB input	1	Turns on the horizontal prefilter for RGB signal.
	0	Turns off the horizontal prefilter.

Table 38.60 Horizontal Prefilter Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_DS4	SC_RES_PFIL_SEL	0	Prefilter Mode Select for Brightness Signals 0: The prefilter is turned off. 1: The prefilter is turned on. (1/4 + 1/2 + 1/4)

38.4.2.3 Horizontal Scale-Down

The number of horizontally arranged pixels can be decreased at a desired ratio in the range of 1/1 to 1/8 using pixel conversion.

For the scaling filter, either hold or linear interpolation mode can be selected.

(1) One-TAP Hold Interpolation

When the interpolation position is between input pixels X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows.

$$X_{interpo} = X_n$$

(2) Two-TAP Linear Interpolation

When the interpolation position is between input pixels X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows based on the interpolation position "phase".

$$X_{interpo} = (X_n \times (4096 - phase) + X_{n+1} \times phase) / 4096$$

(3) Calculation of Horizontal Scale Down Ratio

The value to be set to the horizontal scale-down ratio SC_RES_DS_H_RATIO can be obtained using the following equation based on the number of input pixels SC_RES_HW and number of output pixels SC_RES_OUT_HW, where the decimals are rounded off.

SC_RES_DS_H_RATIO = round (SC_RES_HW ÷ SC_RES_OUT_HW × 4096)

Note that, for 100% horizontal scale-up, the SC_RES_HW and SC_RES_OUT_HW values should be identical and the SC_RES_DS_H_RATIO bits should be set to 4096.

(4) Handling for Lack of Last-Input Pixel

Interpolation is carried out between the second-last-input and last-input pixels to produce the last-output pixel at the right end of a screen. The interpolation position of the last-output pixel may be close to the second-last-input pixel depending on the horizontal scale-down ratio; in this case, it may appear that the last-input pixel is lacking.

The undesirable influence by lack of last-input pixel can be decreased by appropriately adjusting the horizontal scale-down ratio using the following equations.

Pre-adjustment horizontal scale-down ratio RATIO_org should be calculated first to find adjustment value σ, and then scale-down ratio SC_RES_DS_H_RATIO should be determined.

RATIO_org = round (SC_RES_HW ÷ SC_RES_OUT_HW × 4096)

σ = (RATIO_org × (SC_RES_OUT_HW − 1) − (SC_RES_HW − 1) × 4096) ÷ (SC_RES_OUT_HW − 1)

SC_RES_DS_H_RATIO = roundup (RATIO_org − σ)

Table 38.61 Horizontal Scale Down Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_DS1	SC_RES_DS_H_ON	1	Horizontal Scale Down On/Off 0: Off 1: On
SC_SCL0_DS7	SC_RES_OUT_HW[10:0]	640	Number of Valid Horizontal Pixels Output by Scaling-down Control Block (Video-image clock cycles)
SC_SCL0_DS4	SC_RES_DS_H_INTERPOTYP	1	Horizontal Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
SC_SCL0_DS4	SC_RES_DS_H_RATIO[15:0]	9224	Horizontal Scale Down Ratio ([15:12]: Integer part, [11:0]: Decimal part) round(SC_RES_HW ÷ SC_RES_OUT_HW × 4096) SC_RES_DS_H_RATIO < 4096: Setting prohibited SC_RES_DS_H_RATIO = 4096: 100% scale up SC_RES_DS_H_RATIO > 4096: Scale down

Note: The SC_RES_OUT_HW value should be aligned in 4-pixel units and equal to or smaller than the SC_RES_HW value.

38.4.2.4 Vertical Scale-Down

The number of lines can be decreased in the vertical direction at a desired ratio in the range of 1/1 to 1/8 using pixel conversion.

For the scaling filter, either hold or linear interpolation mode can be selected.

(1) One-TAP Hold Interpolation

When the interpolation position is between input lines X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows.

$$X_{interpo} = X_n$$

(2) Two-TAP Linear Interpolation

When the interpolation position is between input lines X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows based on the interpolation position "phase".

$$X_{interpo} = (X_n \times (4096 - \text{phase}) + X_{n+1} \times \text{phase}) / 4096$$

(3) Calculation of Vertical Scale Down Ratio

The value to be set to the vertical scale-down ratio SC_RES_V_RATIO can be obtained using the following equation based on the number of input lines SC_RES_VW and number of output lines SC_RES_OUT_VW, where the decimals are rounded off.

SC_RES_V_RATIO = round (SC_RES_VW ÷ SC_RES_OUT_VW × 4096)

Note that the SC_RES_VW and SC_RES_OUT_VW values should be identical for vertical enlargement or 100% vertical enlargement.

For 100% vertical enlargement, reduction is carried out assuming SC_RES_V_RATIO as 4096.

(4) Handling for Lack of Last-Input Line

Interpolation is carried out between the second-last-input and last-input lines to produce the last-output line at the lower end of a screen. The interpolation position of the last-output line may be close to the

second-last-input line depending on the vertical scale-down ratio; in this case, it may appear that the last-input line is lacking.

The undesirable influence by the lack of last-input line can be decreased by appropriately adjusting the vertical scale-down ratio using the following equations.

Pre-adjustment vertical scale-down ratio RATIO_org should be calculated first to find adjustment value σ, and then scale-down ratio SC_RES_V_RATIO should be determined.

RATIO_org = round (SC_RES_VW ÷ SC_RES_OUT_VW × 4096)

σ = (RATIO_org × (SC_RES_OUT_VW − 1) − (SC_RES_VW − 1) × 4096) ÷ (SC_RES_OUT_VW − 1)

SC_RES_V_RATIO = round (RATIO_org − σ)

Table 38.62 Vertical Scale Down Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_DS1	SC_RES_DS_V_ON	1	Vertical Scale Down On/Off 0: Off 1: On
SC_SCL0_DS7	SC_RES_OUT_VW [10:0]	240	Number of Valid Lines in Vertical Direction Output by Scaling-Down Control Block (lines) This bit setting is used for the number of lines to be written to the frame buffer. When SC_SCL1_WR1.SC_RES_LOOP is 0 (frame write mode), these bits specify the number of lines for one frame. When SC_SCL1_WR1.SC_RES_LOOP is 1 (line write mode), these bits specify the number of lines for writing in a ring configuration.
SC_SCL0_DS5	SC_RES_V_ INTERPOTYP	1	Vertical Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
SC_SCL0_DS6	SC_RES_V_RATIO [15:0]	2044	Vertical Scale UP/Down Ratio ([15:12]: Integer part, [11:0]: Decimal part) For scale down: round(SC_RES_VW ÷ SC_RES_OUT_VW × 4096) For scale up: round(SC_RES_IN_VW ÷ SC_RES_P_VW × 4096) SC_RES_V_RATIO < 4096: Scale up SC_RES_V_RATIO = 4096: 100% scale up SC_RES_V_RATIO > 4096: Scale down

Note: SC_RES_V_RATIO and SC_RES_V_INTERPOTYP are both shared by vertical reduction and vertical enlargement.
It is impossible to use vertical reduction and vertical enlargement simultaneously because they are mutually exclusive.
The SC_RES_OUT_VW value should be aligned in 4-line units and equal to or smaller than the SC_RES_VW value.

38.4.2.5 Horizontal Scale Up

The number of horizontally arranged pixels can be increased at a desired ratio in the range of 1/1 to 8/1 using pixel conversion.

For the scaling filter, either hold or linear interpolation mode can be selected.

(1) One-TAP Hold Interpolation

When the interpolation position is between input pixels X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows.

$X_{interpo} = X_n$

(2) Two-TAP Linear Interpolation

When the interpolation position is between input pixels X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows based on the interpolation position "phase".

$$X_{interpo} = (X_n \times (4096 - \text{phase}) + X_{n+1} \times \text{phase}) / 4096$$

(3) Calculation of Horizontal Scale Up Ratio

The value to be set to the horizontal scale-up ratio SC_RES_US_H_RATIO can be obtained using the following equation based on the number of input pixels SC_RES_IN_HW and number of output pixels SC_RES_P_HW, where the decimals are rounded off.

SC_RES_US_H_RATIO = round (SC_RES_IN_HW ÷ SC_RES_P_HW × 4096)

Note that, for 100% horizontal scale-up, the SC_RES_IN_HW and SC_RES_P_HW values should be identical and the SC_RES_US_H_RATIO bits should be set to 4096.

(4) Folding Handling

Since interpolation is carried out between the last-input pixel and second-last-input folding pixel to produce the last-output pixel at the right end of a screen, folding may undesirably stand out depending on the horizontal scale up ratio.

The undesirable influence by folding pixels can be decreased by appropriately adjusting the horizontal scale-up ratio using the following equations.

Pre-adjustment horizontal scale-up ratio RATIO_org should be calculated first to find adjustment value σ, and then scale-up ratio SC_RES_US_H_RATIO should be determined.

RATIO_org = round (SC_RES_IN_HW ÷ SC_RES_P_HW × 4096)

σ = (RATIO_org × (SC_RES_P_HW − 1) − (SC_RES_IN_HW − 1) × 4096 ÷ (SC_RES_P_HW − 1)

SC_RES_US_H_RATIO = round (RATIO_org − σ)

Table 38.63 Horizontal Scale Up Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_US1	SC_RES_US_H_ON	1	Horizontal Scale Up On/Off 0: Off 1: On
SC_SCL0_US4	SC_RES_IN_HW[10:0]	640	Number of Valid Horizontal Pixels Input to Scaling-down Control Block (Pixel-clock cycles)
SC_SCL0_US6	SC_RES_US_H_ INTERPOTYP	1	Horizontal Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
SC_SCL0_US5	SC_RES_US_H_RATIO [15:0]	9224	Horizontal Scale Up Ratio ([15:12]: Integer part, [11:0]: Decimal part) round(SC_RES_IN_HW ÷ SC_RES_P_HW × 4096) SC_RES_US_H_RATIO < 4096: Scale up SC_RES_US_H_RATIO = 4096: 100% scale-up SC_RES_US_H_RATIO > 4096: Setting prohibited

38.4.2.6 Vertical Scale-Up

The number of lines can be increased in the vertical direction at a desired ratio in the range of 1/1 to 8/1 using pixel conversion.

For the scaling filter, either hold or linear interpolation mode can be selected.

(1) One-TAP Hold Interpolation

When the interpolation position is between input lines X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows.

$$X_{interpo} = X_n$$

(2) Two-TAP Linear Interpolation

When the interpolation position is between input lines X_n and X_{n+1}, the $X_{interpo}$ interpolation value is defined as follows based on the interpolation position "phase".

$$X_{interpo} = (X_n \times (4096 - phase) + X_{n+1} \times phase) / 4096$$

(3) Calculation of Vertical Scale Up Ratio

The value to be set to the vertical scale-up ratio SC_RES_V_RATIO can be obtained using the following equation based on the number of input lines SC_RES_IN_VW and number of output lines SC_RES_P_VW, where the decimals are rounded off.

SC_RES_V_RATIO = round (SC_RES_IN_VW ÷ SC_RES_P_VW × 4096)

Note that, for 100% vertical enlargement or vertical reduction, the SC_RES_IN_VW and SC_RES_P_VW values should be identical.

(4) Folding Handling

The last line to be output at the bottom of the screen is produced by interpolation between the last line and line for folding (second-last line to be input). According to the vertical scale-up rate, this may cause folding to stand out.

The undesirable influence by folding lines can be decreased by appropriately adjusting the vertical scale-up ratio using the following equations.

Pre-adjustment vertical scale-up ratio RATIO_org should be calculated first to find adjustment value σ, and then scale-up ratio SC_RES_V_RATIO should be determined.

RATIO_org = round (SC_RES_IN_VW ÷ SC_RES_P_VW × 4096)

σ = (RATIO_org × (SC_RES_P_VW − 1) − (SC_RES_IN_VW − 1) × 4096) ÷ (SC_RES_P_VW − 1)

SC_RES_V_RATIO = round (RATIO_org − σ)

Table 38.64 Vertical Scale Up Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_US1	SC_RES_US_V_ON	1	Vertical Scale Up On/Off 0: Off 1: On
SC_SCL0_US4	SC_RES_IN_VW[10:0]	240	Number of Valid Lines in Vertical Direction Input to Scaling-down Control Block (Lines)
SC_SCL0_DS5	SC_RES_V_INTERPOTYP	1	Vertical Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
SC_SCL0_DS6	SC_RES_V_RATIO[15:0]	2044	Vertical Scale Up Ratio ([15:12]: Integer part, [11:0]: Decimal part) For scale down: round(SC_RES_VW ÷ SC_RES_OUT_VW × 4096) For scale up: round(SC_RES_IN_VW ÷ SC_RES_P_VW × 4096) SC_RES_V_RATIO < 4096: Scale up SC_RES_V_RATIO = 4096: 100% scale up SC_RES_V_RATIO > 4096: Scale down

Note: SC_RES_V_RATIO and SC_RES_V_INTERPOTYP are both shared by vertical reduction and vertical enlargement.
It is impossible to use vertical reduction and vertical enlargement simultaneously because they are mutually exclusive.

38.4.2.7 IP Conversion

(1) Initial Phase Control

When interlaced signals are input, line flickering caused by the line offset between the top and bottom fields can be decreased before being displayed by independently adjusting the initial scaling phases of the fields.

For various operations, appropriate settings should be made referring to the relevant registers as listed in **Table 38.65**.

Table 38.65 Initial Scaling Phase Settings (Standard Values) for IP Conversion

Mirroring	Horizontal Scaling	Vertical Scaling	Reference Bit (Setting)
Normal	Horizontal scale down	Vertical scale down	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale down	Vertical scale up	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale up	Vertical scale down	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale up	Vertical scale up	SC_RES_TOP_INIPHASE = 2048
Horizontal mirroring	Horizontal scale down	Vertical scale down	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale down	Vertical scale up	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale up	Vertical scale down	SC_RES_TOP_INIPHASE = 2048
	Horizontal scale up	Vertical scale up	SC_RES_TOP_INIPHASE = 2048

Note: Set 0 to the initial phase control registers where the specific value is not shown in the table.
Set 0 to all the initial phase control registers when progressive signals are input.

Table 38.66 Initial Scaling Phase Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_DS5	SC_RES_BTM_INIPHASE [11:0]	0	Vertical Interpolation Start Phase for Bottom Field 0 to 4095 (0 to approx. 1.0)
SC_SCL0_DS5	SC_RES_TOP_INIPHASE [11:0]	2048	Vertical Interpolation Start Phase for Top Field 0 to 4095 (0 to approx. 1.0)
SC_SCL0_US6	SC_RES_US_HB_INIPHASE [11:0]	0	Horizontal Interpolation Start Phase for Bottom Field 0 to 4095 (0 to approx. 1.0)
SC_SCL0_US6	SC_RES_US_HT_INIPHASE [11:0]	0	Horizontal Interpolation Start Phase for Top Field 0 to 4095 (0 to approx. 1.0)

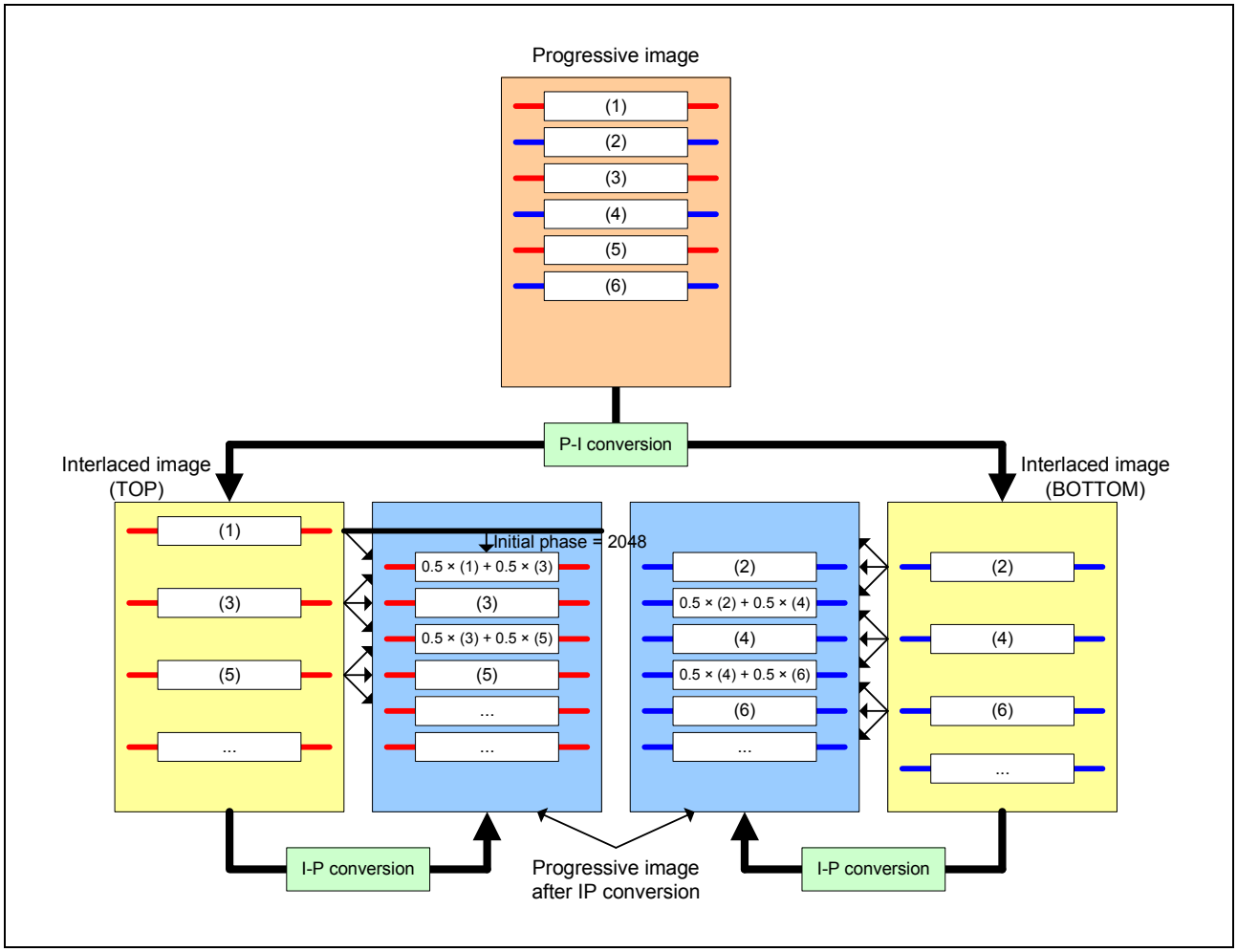


Figure 38.33 IP Conversion Processing Schematic Diagram

(2) Field Determination Signal Control

When interlaced signals are input, the field determination signal can be controlled, which is output to the scaling-up control block during vertical scaling.

When progressive signals are input or vertical scaling is carried out by the scaling-down control block, the field determination signal output to the scaling-up control block is fixed to the specific level, and thus either 0 or 1 can be set to the SC_RES_FLD_DLY_SEL bit.

Table 38.67 Settings for Field Determination Signal Control

Input Signal	Vertical Processing	Frame Buffer	SC_RES_FLD_DLY_SEL
Progressive	—	—	—
Interlace	Vertical scale down	—	—
	Vertical scale up	One plane or less	0
		Two planes or more	1

Table 38.68 Field Determination Signal Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_FRC5	SC_RES_FLD_DLY_SEL	1	Field Determination Signal Delay Control 0: No delay 1: Delay of one vertical cycle

38.4.2.8 Control of Interrupt on Specified Image Line before Scaling-down, and Reading of Current Image Line before Scaling-down

When the location of the image line input to the scaling-down control block matches the SC_SCL1_LINE setting, an interrupt processing is done. In addition, the current location of the line input to the scaling-down control block can be read from a register.

Table 38.69 Control of Interrupt on Specified Image Line before Scaling-down, and Reading of Current Image line before Scaling-down

Register Name	Bit Name	Initial Value	Description
SC_SCL0_INT	SC_RES_LINE[10:0]	All 0	Setting of Interrupt on Image Line Input to Scaling-down Control Block When the location of the image line input to the scaling-down control block matches the SC_SCL0_LINE setting, an interrupt signal is output.
SC_SCL0_MON0	SC_RES_LIN_STAT[10:0]	All 0	Current Location of Image Line Input to Scaling-down Control Block

38.4.2.9 Trimming

The upper, lower, right, and left parts of a post-scaling image can be trimmed off as specified by the SC_RES_VCUT and SC_RES_HCUT bits before being output.

The frame lines of the post-scaling image can also be displayed by setting the SC_RES_DISP_ON bit to 1.

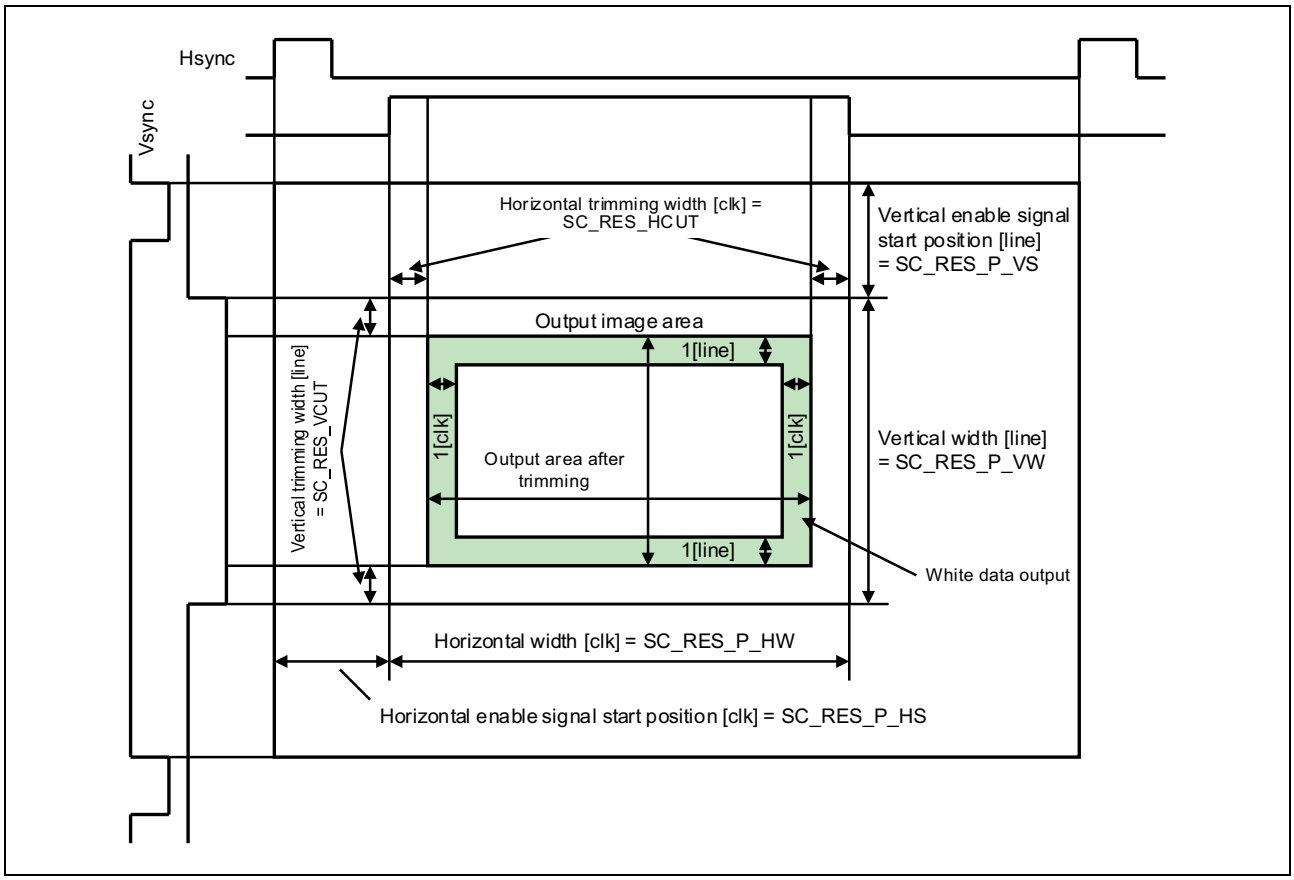


Figure 38.34 Area Relationship for Trimming (Frame Lines Displayed)

Table 38.70 Trimming Control

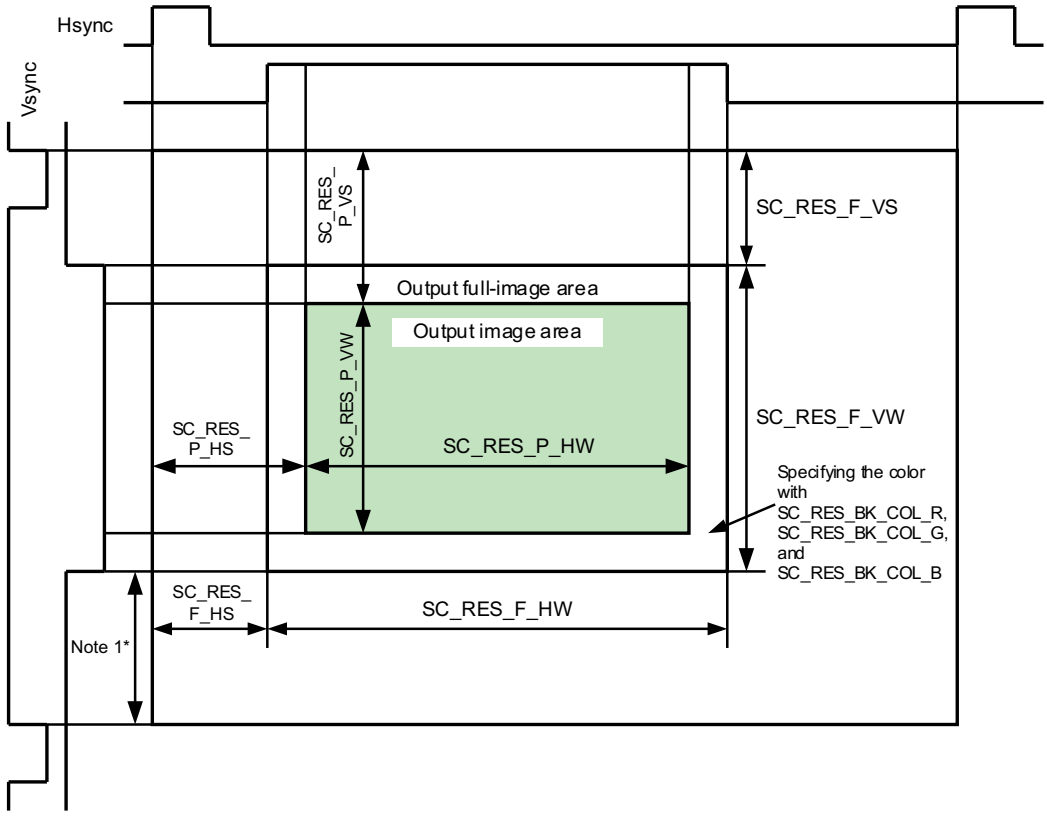
Register Name	Bit Name	Initial Value	Description
SC_SCL0_US7	SC_RES_HCUT[7:0]	0	Horizontal Amount of Cut-off Post-Scaling Image (Right and Left Parts) Sets the number of pixel-clock cycles.
SC_SCL0_US7	SC_RES_VCUT[7:0]	0	Vertical Amount of Cut-off Post-Scaling Image (Upper and Lower Parts) Sets the number of lines.
SC_SCL0_US8	SC_RES_DISP_ON	0	Post-Scaling Image Frame Display On/Off 0: Frame display on 1: Frame display off

38.4.2.10 Screen Synthesis

During the valid full-screen period, the image area can be overlaid before being output. If the image area to be output is smaller than a full-screen, the background color specified by the SC_RES_BK_COL_R, SC_RES_BK_COL_G, and SC_RES_BK_COL_B bits are displayed to fill the background.

Table 38.71 Screen Synthesis Control

Register Name	Bit Name	Initial Value	Description
SC_SCL0_OVR1	SC_RES_BK_COL_R [7:0]	128	Background Color Setting R/Cr Signal R:8 bits; unsigned (0 to 255 [LSB]) Cr:8 bits; 128 offset binary; unsigned (0 to 255 [LSB])
SC_SCL0_OVR1	SC_RES_BK_COL_B [7:0]	128	Background Color Setting B/Cb Signal B:8 bits; unsigned (0 to 255 [LSB]) Cb:8 bits; 128 offset binary; unsigned (0 to 255 [LSB])
SC_SCL0_OVR1	SC_RES_BK_COL_G [7:0]	0	Background Color Setting G/Y Signal G/Y:8 bits; unsigned (0 to 255 [LSB])



Note 1. More than one line is required for vertical front porch.
Note 2. Output image area must always be placed in Output full image area defined by RES_F_HW and RES_F_VW.

Figure 38.35 Area Relationship with Output Image Size Smaller than a Full Screen

38.4.2.11 Selecting Format for Writing Video Image Signals to Frame Buffer

A format can be selected for writing video image signals to the frame buffer.

Although 24-bit YCbCr signals or 24-bit RGB signals are input to the scaling control block, they are converted into 16-bit YCbCr422 signals, 16-bit RGB565 signals, 32-bit YCbCr444 signals, or 32-bit RGB888 signals before being written to the frame buffer.

As bit reduction processing of RGB565, rounding off or 2 × 2 pattern dither can be selected with the SC_RES_DTH_ON bit. For details on pattern dither, see **Section 38.8.1.7**.

Table 38.72 Frame Buffer Writing Mode Setting

RES_BITDEC_ON	SC_RES_MD[1:0]	Writing Mode
0	3	YCbCr444 (normal, horizontal mirroring)
0	2	RGB888 (normal, horizontal mirroring)
1	1	RGB565 (normal, horizontal mirroring)
0	0	YCbCr422 (normal, horizontal mirroring)

Note 1. Only scaler 0 can output signals to the image renderer.

Table 38.73 Video Signal Writing Format Selection Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR1	SC_RES_MD[1:0]	0	Frame Buffer Video-Signal Writing Format 0: YCbCr422 (16 bits) 1: RGB565 (16 bits) 2: RGB888 (24 (32) bits) 3: YCbCr444 (24 (32) bits)
SC_SCL1_WR6	SC_RES_BITDEC_ON	0	Bit Reduction On/Off 0: Off 1: On
SC_SCL1_WR6	SC_RES_DTH_ON	0	Dither Correction On/Off 0: Off (rounded off) 1: On (2 × 2 pattern dither)

38.4.2.12 Horizontal Mirroring

Horizontal mirroring can be carried out for scaled-down images before being written to the frame buffer.

Table 38.74 and **Table 38.75** show the relationship between various writing modes for image processing and video signals.

Table 38.74 Relationship between Writing Modes and Video Signals

RES_DS_WR_MD[2:0]	Writing Modes	YCbCr444	YCbCr422	RGB565	RGB888
0	Normal writing	Enabled	Enabled	Enabled	Enabled
1	Horizontal mirroring	Enabled	Enabled	Enabled	Enabled
2 to 7	Setting prohibited	—	—	—	—

Table 38.75 Horizontal Mirroring Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR1	SC_RES_DS_WR_MD [2:0]	0	Frame Buffer Writing Mode for Image Processing 0: Normal 1: Horizontal mirroring 2 to 7: Setting prohibited

38.4.2.13 Writing to Frame Buffer

(1) Frame Buffer Write Control

Frame buffer writing is enabled or disabled.

Table 38.76 Frame Buffer Writing Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR5	SC_RES_WENB	0	Frame Buffer Write Enable After making the setting to enable writing, writing starts from the second frame. 0: Frame buffer writing is disabled. 1: Frame buffer writing is enabled.

(2) Frame Buffer Writing Rate Selection

A frame buffer writing rate can be selected from among 1/1, 1/2, 1/4, and 1/8 the vertical frequency of the input signal.

When 1/2, 1/4, or 1/8 is selected, either the top or bottom field can be selected for writing.

Table 38.77 Frame Buffer Write Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR5	SC_RES_FS_RATE [1:0]	0	Writing Rate Sets the frame buffer writing rate to the vertical frequency of the input signal. 0: 1/1 an input signal (The SC_RES_FLD_SEL setting is invalid.) 1: 1/2 an input signal 2: 1/4 an input signal 3: 1/8 an input signal
SC_SCL1_WR5	SC_RES_FLD_SEL	0	Write Field Select 0: Top field 1: Bottom field
SC_SCL1_WR5	SC_RES_INTER	1	Field Operating Mode Select 0: Progressive 1: Interlace

(3) Frame Buffer Write Addresses

Frame buffer addresses are specified using the base address, line offset address, frame offset address, data size of a line, and the number of lines in a frame. When an interlaced video image is input, the top and bottom field data can be separately stored in the frame buffer.

The SC_RES_BASE[31:0], SC_RES_LN_OFF[14:0], and SC_RES_FLM_OFF[22:0] bits should be set in 128-byte units (the lower seven bits should be fixed to 0).

For the data size of a line and the number of lines in a frame, the relevant register values set for the scaling-down control block are used.

Table 38.78 Frame Buffer Write Address Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR1	SC_RES_TB_ADD_MOD	0	Top and Bottom Data Write Address Specification Method 0: A write address is specified in common for top and bottom data. 1: Separate write addresses are specified for top and bottom data.
SC_SCL1_WR2	SC_RES_BASE [31:0]	0	Frame Buffer Base Address Sets the start address of the frame buffer to store the frame data for the top field when SC_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC_RES_TB_ADD_MOD = 0.
SC_SCL1_WR8	SC_RES_BASE_B [31:0]	0	Frame Buffer Base Address for Bottom Sets the start address of the frame buffer to store the frame data for the bottom field when SC_RES_TB_ADD_MOD = 1.
SC_SCL1_WR3	SC_RES_LN_OFF [14:0]	2048	Frame Buffer Line Offset Address Sets the line offset address for calculating the line start address for the top field when SC_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC_RES_TB_ADD_MOD = 0. Line 0: SC_RES_BASE Line 1: SC_RES_BASE + SC_RES_LN_OFF × 1 : Line n: SC_RES_BASE + SC_RES_LN_OFF × n
SC_SCL1_WR9	SC_RES_LN_OFF_B [14:0]	2048	Frame Buffer Line Offset Address for Bottom Sets the line offset address for calculating the line start address for the bottom field when SC_RES_TB_ADD_MOD = 1. Line 0: SC_RES_BASE_B Line 1: SC_RES_BASE_B + SC_RES_LN_OFF_B × 1 : Line n: SC_RES_BASE_B+ SC_RES_LN_OFF_B × n
SC_SCL1_WR4	SC_RES_FLM_OFF [22:0]	524288	Frame Buffer Frame Offset Address Sets the frame offset address for calculating the start address of each frame for the top field when SC_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC_RES_TB_ADD_MOD = 0. Buffer 0: SC_RES_BASE Buffer 1: SC_RES_BASE + SC_RES_FLM_OFF × 1 : Buffer n: SC_RES_BASE + SC_RES_FLM_OFF × n
SC_SCL1_WR10	SC_RES_FLM_OFF_B [22:0]	524288	Frame Buffer Frame Offset Address for Bottom Sets the frame offset address for calculating the start address of each frame for the bottom field when SC_RES_TB_ADD_MOD = 1. Buffer 0: SC_RES_BASE_B Buffer 1: SC_RES_BASE_B + SC_RES_FLM_OFF_B × 1 : Buffer n: SC_RES_BASE_B + SC_RES_FLM_OFF_B × n

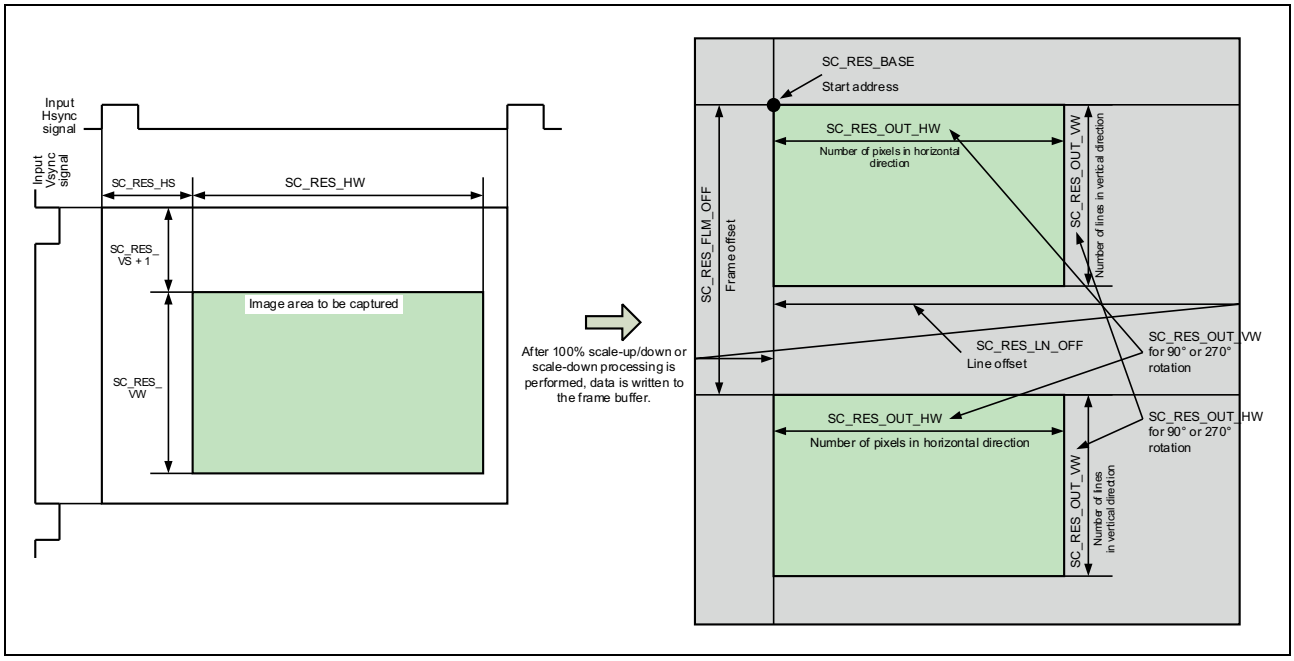


Figure 38.36 Data Arrangement in Frame Buffer

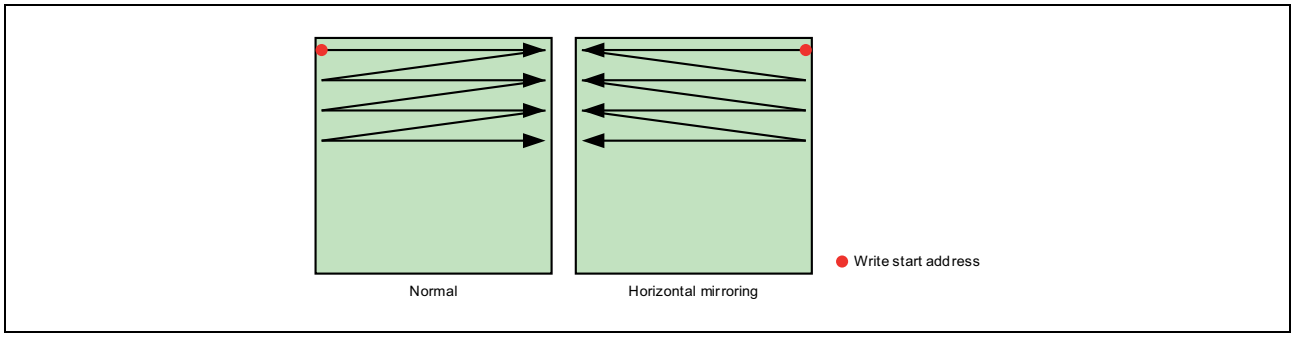


Figure 38.37 Data Arrangement in Frame Buffer in Various Writing Modes

(4) Frame Buffer Management

The scaling control block can handle multiple frames as the frame buffer.

Data is written to the buffer in cyclic mode according to the number of frames specified by the SC_RES_FLM_NUM bits.

To use the frame buffer as the ring buffer in line mode, the SC_RES_FLM_NUM bits should be set to 0 (1 frame) and the SC_RES_LOOP bit to 1.

Table 38.79 Frame Buffer Write Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR3	SC_RES_FLM_ NUM[9:0]	1	Number of Frames of Buffer to be Written to Sets the number of frames for the top field when SC_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC_RES_TB_ADD_MOD = 0 Number of frames defined by SC_RES_FLM_NUM + 1 are used.
SC_SCL1_WR9	SC_RES_FLM_ NUM_B[9:0]	1	Number of Frames of Buffer to be Written to for Bottom Field when SC_RES_TB_ADD_MOD = 1 Number of frames defined by SC_RES_FLM_NUM_B + 1 are used.
SC_SCL1_WR1	SC_RES_LOOP	0	Frame Buffer Write Mode Select 0: Frame mode 1: Line mode (read as ring buffer)
SC_SCL1_WR7	SC_RES_FLM_ CNT[9:0]	—	Frame Number Before Frame Being Accessed Frame number before the frame being accessed in the top field when SC_RES_TB_ADD_MOD = 1 or that in the top or bottom field when SC_RES_TB_ADD_MOD = 0.
SC_SCL1_WR11	SC_RES_FLM_ CNT_B[9:0]	—	Frame Number Before Frame Being Accessed in Bottom Field Frame number before the frame being accessed in the bottom field when SC_RES_TB_ADD_MOD = 1.

(5) Buffer Overflow Handling

If writing to the frame buffer cannot be completed due to bus-traffic related problems, an overflow interrupt can be output to the interrupt controller.

Table 38.80 Buffer Overflow Detection

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR7	SC_RES_OVERFLOW	—	Line Buffer Overflow Detect 1: Line buffer has overflowed. 0: Line buffer has not overflowed.

(6) Frame Buffer Write End Flag

When writing one frame of data to the frame buffer is completed, a frame buffer write end interrupt can be output to the interrupt controller.

38.4.2.14 Selecting a Scaling-up Process or Graphics 0 or 1 Process

Scaling-up process and graphics 0 or 1 process are mutually exclusive and thus frame buffer cannot be read out simultaneously for the processes.

When displaying input video image signals or displaying enlarged graphics, data is read from the frame buffer via the scaling-up control block.

However, graphics can be enlarged and displayed by the scaling-up control block only when the RGB565, RGB888, YCbCr422, or YCbCr444 format is used.

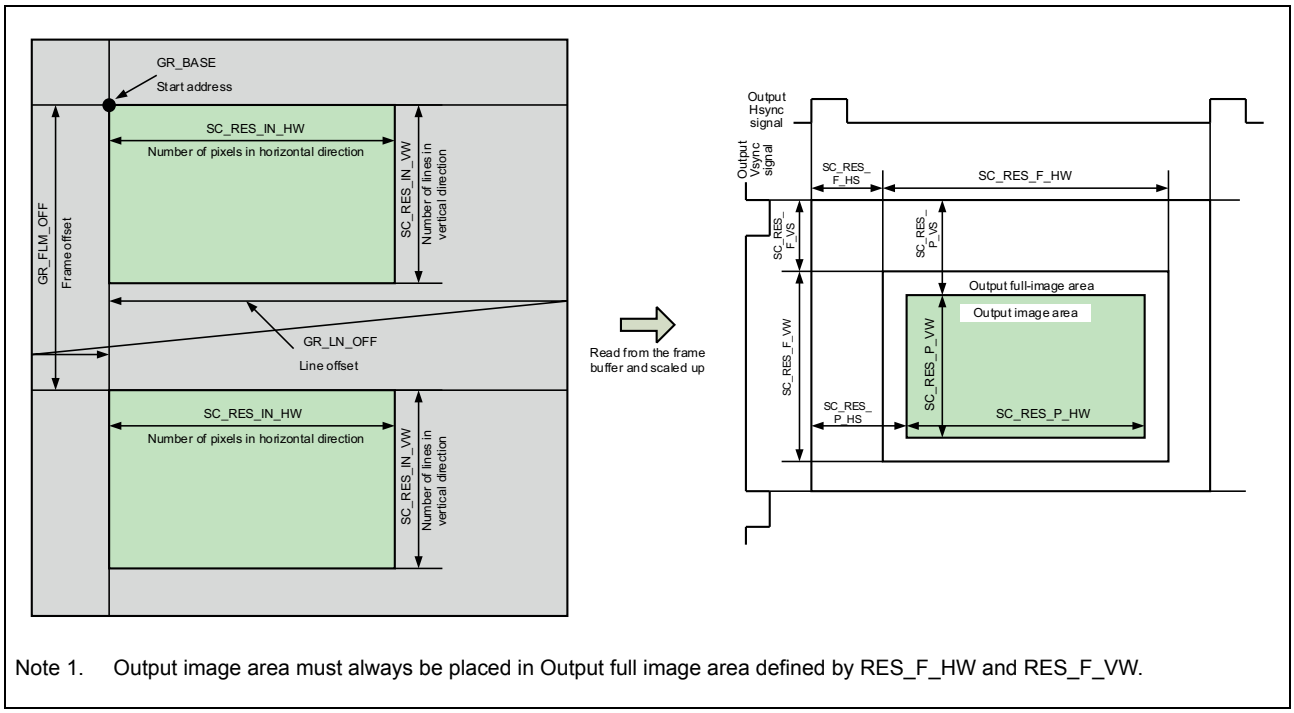
When displaying graphics without enlargement, the data is read from the frame buffer via the graphics 0 or 1 processing block.

With the SC_RES_IBUS_SYNC_SEL bit, sync signals for reading out the frame buffer and read size setting bits are selected.

Table 38.81 Selection of Scaling-Up Process and Graphics 0 or 1 Process

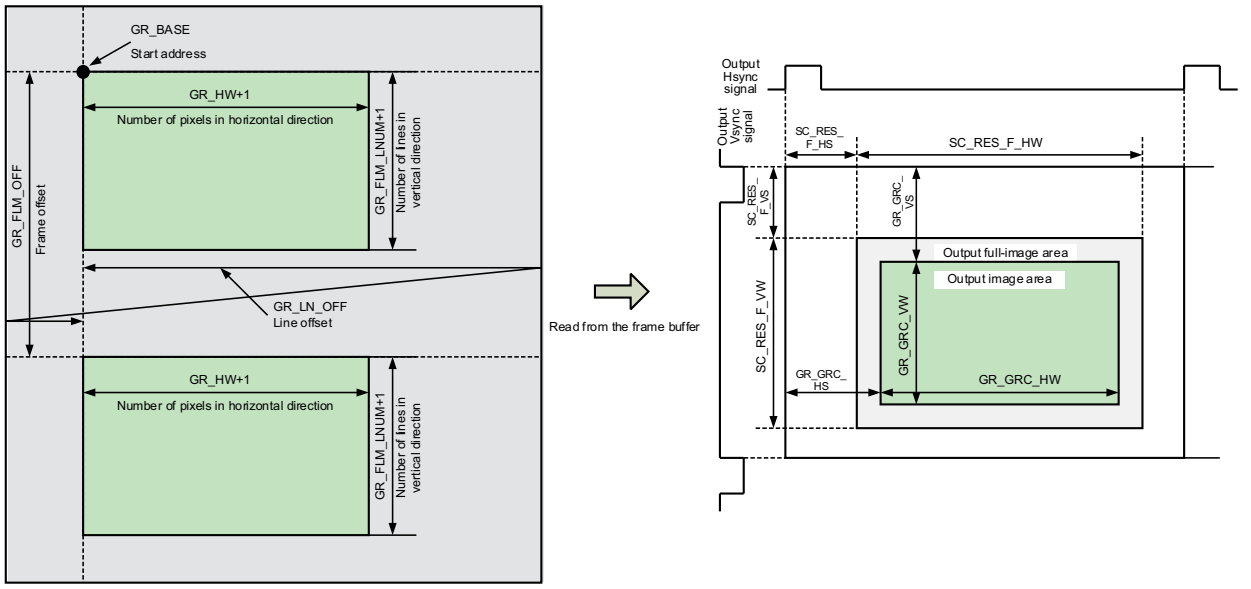
Type of Output Scaling Display	SC_RES_ IBUS_SYNC_SEL	Sync Signals for Frame Buffer Read	Frame Buffer Read Size Setting Bits	Display Enabling Bits
Input video signal display Enlarged graphics display	0	Output from scaling-up control block	SC_RES_IN_VW SC_RES_IN_HW	SC_RES_P_VS SC_RES_P_VW SC_RES_P_HS SC_RES_P_HW
Graphics display	1	Output from graphics 0 or 1 processing block	GR_FLM_LNUM*[1] GR_HW*	GR_GRC_VS GR_GRC_VW GR_GRC_HS GR_GRC_HW

Note 1. The value set to the register + 1 is the actual read size.



Note 1. Output image area must always be placed in Output full image area defined by RES_F_HW and RES_F_VW.

Figure 38.38 Area Setting for Input Video Image Signal Display and Enlarged Graphics Display



Note 1. Output image area must always be placed in Output full image area defined by RES_F_HW and RES_F_VW.

Figure 38.39 Area Setting for Graphics Display

Table 38.82 Scaling-Up Process or Graphics 0 or 1 Process Selection

Register Name	Bit Name	Initial Value	Description
SC_SCL0_US8	SC_RES_IBUS_SYNC_SEL	0	Sync Signal Select for Frame Buffer Read Block 0: Sync signals from the scaling-up control block 1: Sync signals from the graphics processing block

The GR_DISP_SEL bits are used to select a display by the scaling-up control block (video image display or enlarged graphics display) or graphics display.

For details on the graphics processing, refer to the Section 38.6, Image Synthesizer.

38.4.2.15 Selecting Field for Frame Buffer Reading

For the next frame buffer to be read, the top or bottom field can be selected. This field selection is used in the scaling-up control block.

Table 38.83 Field Specification for Frame Buffer Reading

Register Name	Bit Name	Initial Value	Description
GR_FLM1	GR_FLD_SEL	0	Enables or disables top or bottom field selection for the next frame buffer to be read. 0: Field selection is disabled. 1: Top or bottom field can be selected.
GR_FLM3	GR_FLD_NXT	0	Selects the top or bottom field for the next frame buffer. 0: Bottom 1: Top

38.4.2.16 Frame Buffer Reading Processing

(1) Write Pointer Control

The write pointer is incremented by one every time frame data writing is completed.

(2) Read Pointer Control

The read pointer is updated at the rising edge of the vertical sync signal on the reading side according to the difference between the read and write pointer values as follows.

(a) When (write pointer value) – (read pointer value) ≤ 1
The read pointer value is not updated (the same frame is displayed continuously).

(b) When (write pointer value) – (read pointer value) = 2
The read pointer is incremented by one with the next updating timing.

(c) When (write pointer value) – (read pointer value) ≥ 3
The read pointer is incremented by two with the next updating timing.
(One frame is skipped.)

(3) Frame Buffer Read Control

(a) When SC_RES_WENB = 0
Frame data is not written to the frame buffer, and the frame buffer is not read.

(b) When frame data writing is terminated with SC_RES_WENB = 1
As the pointer buffer value is determined, the frame buffer is read.

Table 38.84 Frame Buffer Control

Register Name	Bit Name	Initial Value	Description
SC_SCL1_WR5	SC_RES_WENB	0	Frame Buffer Write Enable After making the setting to enable writing, writing starts from the second frame. 0: Writing is disabled. 1: Writing is enabled.
GR_FLM1	GR_FLM_SEL	0	Frame Buffer Address Setting Signal Selection 0: Links to scaling-down process. (This setting is prohibited when separate write addresses are specified for the top and bottom fields; that is, SC0_RES_TB_ADD_MOD = 1 in SC0_SCL1_WR1.) 1: Selects GR0_FLM_NUM[9:0].
GR_FLM2	GR_BASE	0	Frame Buffer Base Address Sets the start address of the frame buffer where frame data is to be stored. The lower seven bits should be fixed to 000 0000.

For other frame buffer read operation and graphics processing, refer to Section 38.6, Image Synthesizer.

38.4.2.17 Cascaded Connection

To display one video image plane + three graphics planes or to display four graphics planes, scaler 0 and graphics block 1 in scaler 1 are cascaded.

When cascaded connection is selected, the scaling-up processing in scaler 1 is not available.

Table 38.85 Cascaded Connection

Register Name	Bit Name	Initial Value	Description
GR1_AB1	GR1_CUS_CON_ON	0	Cascaded Connection Enable/Disable 0: Cascaded connection is disabled. 1: Cascaded connection is enabled.

38.4.3 Register Descriptions

Table 38.86 and **Table 38.87** show the register configuration.

Symbols used in Register Description:

Initial value: Register value after a reset

—: Undefined value

R/W: Readable/writable. The written value can be read.

R/WC0: Readable/writable. Writing 0 initializes the bit. Writing 1 is ignored.

R/WC1: Readable/writable. Writing 1 initializes the bit. Writing 0 is ignored.

R: Read-only. The write value should always be 0.

—/W: Write-only. The read value is undefined.

Table 38.86 Register Configuration of the Scaler (Channel 0) (1/5)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	SCL0 register update control register (SC0)	SC0_SCL0_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1500	32
VDCE0	Mask control register (SC0)	SC0_SCL0_FRC1	R/W	H'0AF0 0001	<VDCE0_base> + 1504	32
VDCE0	Missing Vsync compensation control register (SC0)	SC0_SCL0_FRC2	R/W	H'0E10 0001	<VDCE0_base> + 1508	32
VDCE0	Output sync select register (SC0)	SC0_SCL0_FRC3	R/W	H'0000 0001	<VDCE0_base> + 150C	32
VDCE0	Free-running period control register (SC0)	SC0_SCL0_FRC4	R/W	H'020C 031F	<VDCE0_base> + 1510	32
VDCE0	Output delay control register (SC0)	SC0_SCL0_FRC5	R/W	H'0000 0101	<VDCE0_base> + 1514	32
VDCE0	Full-screen vertical size register (SC0)	SC0_SCL0_FRC6	R/W	H'0023 01E0	<VDCE0_base> + 1518	32
VDCE0	Full-screen horizontal size register (SC0)	SC0_SCL0_FRC7	R/W	H'0090 0280	<VDCE0_base> + 151C	32
VDCE0	Vsync detection register (SC0)	SC0_SCL0_FRC9	R	H'0000 0000	<VDCE0_base> + 1524	32
VDCE0	Status monitor 0 register (SC0)	SC0_SCL0_MON0	R	H'0000	<VDCE0_base> + 1528	16
VDCE0	Interrupt control register (SC0)	SC0_SCL0_INT	R/W	H'0000	<VDCE0_base> + 152A	16
VDCE0	Scaling-down control register (SC0)	SC0_SCL0_DS1	R/W	H'0000 0011	<VDCE0_base> + 152C	32
VDCE0	Vertical capture size register (SC0)	SC0_SCL0_DS2	R/W	H'0012 00F0	<VDCE0_base> + 1530	32
VDCE0	Horizontal capture size register (SC0)	SC0_SCL0_DS3	R/W	H'00F4 05A0	<VDCE0_base> + 1534	32
VDCE0	Horizontal scale down register (SC0)	SC0_SCL0_DS4	R/W	H'1000 2408	<VDCE0_base> + 1538	32
VDCE0	Initial vertical phase register (SC0)	SC0_SCL0_DS5	R/W	H'1800 0000	<VDCE0_base> + 153C	32
VDCE0	Vertical scaling register (SC0)	SC0_SCL0_DS6	R/W	H'0000 07FC	<VDCE0_base> + 1540	32
VDCE0	Scaling-down control block output size register (SC0)	SC0_SCL0_DS7	R/W	H'00F0 0280	<VDCE0_base> + 1544	32

Table 38.86 Register Configuration of the Scaler (Channel 0) (2/5)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Scaling-up control register (SC0)	SC0_SCL0_US1	R/W	H'0000 0011	<VDCE0_base> + 1548	32
VDCE0	Output image vertical size register (SC0)	SC0_SCL0_US2	R/W	H'0023 01E0	<VDCE0_base> + 154C	32
VDCE0	Output image horizontal size register (SC0)	SC0_SCL0_US3	R/W	H'0090 0280	<VDCE0_base> + 1550	32
VDCE0	Scaling-up control block input size register (SC0)	SC0_SCL0_US4	R/W	H'00F0 0280	<VDCE0_base> + 1554	32
VDCE0	Horizontal scale up register (SC0)	SC0_SCL0_US5	R/W	H'0000 2408	<VDCE0_base> + 1558	32
VDCE0	Horizontal scale up initial phase register (SC0)	SC0_SCL0_US6	R/W	H'1000 0000	<VDCE0_base> + 155C	32
VDCE0	Trimming register (SC0)	SC0_SCL0_US7	R/W	H'0000 0000	<VDCE0_base> + 1560	32
VDCE0	Frame buffer read select register (SC0)	SC0_SCL0_US8	R/W	H'0000 0000	<VDCE0_base> + 1564	32
VDCE0	Background color register (SC0)	SC0_SCL0_OVR1	R/W	H'0080 0080	<VDCE0_base> + 156C	32
VDCE0	SCL1 register update control register (SC0)	SC0_SCL1_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1580	32
VDCE0	Writing mode register (SC0)	SC0_SCL1_WR1	R/W	H'0000 0000	<VDCE0_base> + 1588	32
VDCE0	Write address register 1T (SC0)	SC0_SCL1_WR2	R/W	H'0000 0000	<VDCE0_base> + 158C	32
VDCE0	Write address register 2T (SC0)	SC0_SCL1_WR3	R/W	H'0800 0001	<VDCE0_base> + 1590	32
VDCE0	Write address register 3T (SC0)	SC0_SCL1_WR4	R/W	H'0008 0000	<VDCE0_base> + 1594	32
VDCE0	Frame sub-sampling register (SC0)	SC0_SCL1_WR5	R/W	H'0000 1000	<VDCE0_base> + 159C	32
VDCE0	Bit reduction register (SC0)	SC0_SCL1_WR6	R/W	H'0000 0000	<VDCE0_base> + 15A0	32
VDCE0	Write detection register (SC0)	SC0_SCL1_WR7	R	H'0000 0000	<VDCE0_base> + 15A4	32
VDCE0	Write address register 1B (SC0)	SC0_SCL1_WR8	R/W	H'0000 0000	<VDCE0_base> + 15A8	32
VDCE0	Write address register 2B (SC0)	SC0_SCL1_WR9	R/W	H'0800 0001	<VDCE0_base> + 15AC	32
VDCE0	Write address register 3B (SC0)	SC0_SCL1_WR10	R/W	H'0008 0000	<VDCE0_base> + 15B0	32
VDCE0	Write detection register B (SC0)	SC0_SCL1_WR11	R	H'0000 0000	<VDCE0_base> + 15B4	32
VDCE0	Graphics 0 register update control register	GR0_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1600	32
VDCE0	Frame buffer read control register (graphics 0)	GR0_FLM_RD	R/W	H'0000 0000	<VDCE0_base> + 1604	32
VDCE0	Frame buffer control register 1 (graphics 0)	GR0_FLM1	R/W	H'0000 0000	<VDCE0_base> + 1608	32
VDCE0	Frame buffer control register 2 (graphics 0)	GR0_FLM2	R/W	H'0000 0000	<VDCE0_base> + 160C	32
VDCE0	Frame buffer control register 3 (graphics 0)	GR0_FLM3	R/W	H'0800 0001	<VDCE0_base> + 1610	32

Table 38.86 Register Configuration of the Scaler (Channel 0) (3/5)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Frame buffer control register 4 (graphics 0)	GR0_FLM4	R/W	H'0008 0000	<VDCE0_base> + 1614	32
VDCE0	Frame buffer control register 5 (graphics 0)	GR0_FLM5	R/W	H'0000 03FF	<VDCE0_base> + 1618	32
VDCE0	Frame buffer control register 6 (graphics 0)	GR0_FLM6	R/W	H'8000 0000	<VDCE0_base> + 161C	32
VDCE0	Alpha blending control register 1 (graphics 0)	GR0_AB1	R/W	H'0000 0000	<VDCE0_base> + 1620	32
VDCE0	Alpha blending control register 2 (graphics 0)	GR0_AB2	R/W	H'0000 0000	<VDCE0_base> + 1624	32
VDCE0	Alpha blending control register 3 (graphics 0)	GR0_AB3	R/W	H'0000 0000	<VDCE0_base> + 1628	32
VDCE0	Alpha blending control register 7 (graphics 0)	GR0_AB7	R/W	H'00FF 0000	<VDCE0_base> + 1638	32
VDCE0	Alpha blending control register 8 (graphics 0)	GR0_AB8	R/W	H'0000 0000	<VDCE0_base> + 163C	32
VDCE0	Alpha blending control register 9 (graphics 0)	GR0_AB9	R/W	H'0000 0000	<VDCE0_base> + 1640	32
VDCE0	Alpha blending control register 10 (graphics 0)	GR0_AB10	R/W	H'0000 0000	<VDCE0_base> + 1644	32
VDCE0	Alpha blending control register 11 (graphics 0)	GR0_AB11	R/W	H'0000 0000	<VDCE0_base> + 1648	32
VDCE0	Background color control register (graphics 0)	GR0_BASE	R/W	H'0000 8080	<VDCE0_base> + 164C	32
VDCE0	CLUT table control register (graphics 0)	GR0_CLUT	R/W	H'0000 0000	<VDCE0_base> + 1650	32
VDCE0	SCL0 register update control register (SC1)	SC1_SCL0_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1C00	32
VDCE0	Mask control register (SC1)	SC1_SCL0_FRC1	R/W	H'0AF0 0001	<VDCE0_base> + 1C04	32
VDCE0	Missing Vsync compensation control register (SC1)	SC1_SCL0_FRC2	R/W	H'0E10 0001	<VDCE0_base> + 1C08	32
VDCE0	Output sync select register (SC1)	SC1_SCL0_FRC3	R/W	H'0000 0001	<VDCE0_base> + 1C0C	32
VDCE0	Free-running period control register (SC1)	SC1_SCL0_FRC4	R/W	H'020C 031F	<VDCE0_base> + 1C10	32
VDCE0	Output delay control register (SC1)	SC1_SCL0_FRC5	R/W	H'0000 0101	<VDCE0_base> + 1C14	32
VDCE0	Full-screen vertical size register (SC1)	SC1_SCL0_FRC6	R/W	H'0023 01E0	<VDCE0_base> + 1C18	32
VDCE0	Full-screen horizontal size register (SC1)	SC1_SCL0_FRC7	R/W	H'0090 0280	<VDCE0_base> + 1C1C	32
VDCE0	Vsync detection register (SC1)	SC1_SCL0_FRC9	R	H'0000 0000	<VDCE0_base> + 1C24	32
VDCE0	Scaling-down control register (SC1)	SC1_SCL0_DS1	R/W	H'0000 0011	<VDCE0_base> + 1C2C	32
VDCE0	Vertical scaling register (SC1)	SC1_SCL0_DS6	R/W	H'0000 07FC	<VDCE0_base> + 1C40	32
VDCE0	Scaling-up control register (SC1)	SC1_SCL0_US1	R/W	H'0000 0011	<VDCE0_base> + 1C48	32
VDCE0	Output image vertical size register (SC1)	SC1_SCL0_US2	R/W	H'0023 01E0	<VDCE0_base> + 1C4C	32

Table 38.86 Register Configuration of the Scaler (Channel 0) (4/5)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Output image horizontal size register (SC1)	SC1_SCL0_US3	R/W	H'0090 0280	<VDCE0_base> + 1C50	32
VDCE0	Scaling-up control block input size register (SC1)	SC1_SCL0_US4	R/W	H'00F0 0280	<VDCE0_base> + 1C54	32
VDCE0	Horizontal scale up register (SC1)	SC1_SCL0_US5	R/W	H'0000 2408	<VDCE0_base> + 1C58	32
VDCE0	Horizontal scale up initial phase register (SC1)	SC1_SCL0_US6	R/W	H'1000 0000	<VDCE0_base> + 1C5C	32
VDCE0	Trimming register (SC1)	SC1_SCL0_US7	R/W	H'0000 0000	<VDCE0_base> + 1C60	32
VDCE0	Frame buffer read select register (SC1)	SC1_SCL0_US8	R/W	H'0000 0000	<VDCE0_base> + 1C64	32
VDCE0	Background color register (SC1)	SC1_SCL0_OVR1	R/W	H'0080 0080	<VDCE0_base> + 1C6C	32
VDCE0	Graphics 1 register update control register	GR1_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1D00	32
VDCE0	Frame buffer read control register (graphics 1)	GR1_FLM_RD	R/W	H'0000 0000	<VDCE0_base> + 1D04	32
VDCE0	Frame buffer control register 1 (graphics 1)	GR1_FLM1	R/W	H'0000 0000	<VDCE0_base> + 1D08	32
VDCE0	Frame buffer control register 2 (graphics 1)	GR1_FLM2	R/W	H'0000 0000	<VDCE0_base> + 1D0C	32
VDCE0	Frame buffer control register 3 (graphics 1)	GR1_FLM3	R/W	H'0800 0001	<VDCE0_base> + 1D10	32
VDCE0	Frame buffer control register 4 (graphics 1)	GR1_FLM4	R/W	H'0008 0000	<VDCE0_base> + 1D14	32
VDCE0	Frame buffer control register 5 (graphics 1)	GR1_FLM5	R/W	H'0000 03FF	<VDCE0_base> + 1D18	32
VDCE0	Frame buffer control register 6 (graphics 1)	GR1_FLM6	R/W	H'8000 0000	<VDCE0_base> + 1D1C	32
VDCE0	Alpha blending control register 1 (graphics 1)	GR1_AB1	R/W	H'0000 0000	<VDCE0_base> + 1D20	32
VDCE0	Alpha blending control register 2 (graphics 1)	GR1_AB2	R/W	H'0000 0000	<VDCE0_base> + 1D24	32
VDCE0	Alpha blending control register 3 (graphics 1)	GR1_AB3	R/W	H'0000 0000	<VDCE0_base> + 1D28	32
VDCE0	Alpha blending control register 4 (graphics 1)	GR1_AB4	R/W	H'0000 0000	<VDCE0_base> + 1D2C	32
VDCE0	Alpha blending control register 5 (graphics 1)	GR1_AB5	R/W	H'0000 0000	<VDCE0_base> + 1D30	32
VDCE0	Alpha blending control register 6 (graphics 1)	GR1_AB6	R/W	H'0000 0000	<VDCE0_base> + 1D34	32
VDCE0	Alpha blending control register 7 (graphics 1)	GR1_AB7	R/W	H'00FF 0000	<VDCE0_base> + 1D38	32
VDCE0	Alpha blending control register 8 (graphics 1)	GR1_AB8	R/W	H'0000 0000	<VDCE0_base> + 1D3C	32
VDCE0	Alpha blending control register 9 (graphics 1)	GR1_AB9	R/W	H'0000 0000	<VDCE0_base> + 1D40	32
VDCE0	Alpha blending control register 10 (graphics 1)	GR1_AB10	R/W	H'0000 0000	<VDCE0_base> + 1D44	32
VDCE0	Alpha blending control register 11 (graphics 1)	GR1_AB11	R/W	H'0000 0000	<VDCE0_base> + 1D48	32

Table 38.86 Register Configuration of the Scaler (Channel 0) (5/5)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Background color control register (graphics 1)	GR1_BASE	R/W	H'0000 8080	<VDCE0_base> + 1D4C	32
VDCE0	CLUT table control register (graphics 1)	GR1_CLUT	R/W	H'0000 0000	<VDCE0_base> + 1D50	32
VDCE0	Status monitor register (graphics 1)	GR1_MON	R	H'0000 0000	<VDCE0_base> + 1D54	32

Table 38.87 Register Configuration of the Scaler (Channel 1) (1/4)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	SCL0 register update control register (SC0)	SC0_SCL0_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1500	32
VDCE1	Mask control register (SC0)	SC0_SCL0_FRC1	R/W	H'0AF0 0001	<VDCE1_base> + 1504	32
VDCE1	Missing Vsync compensation control register (SC0)	SC0_SCL0_FRC2	R/W	H'0E10 0001	<VDCE1_base> + 1508	32
VDCE1	Output sync select register (SC0)	SC0_SCL0_FRC3	R/W	H'0000 0001	<VDCE1_base> + 150C	32
VDCE1	Free-running period control register (SC0)	SC0_SCL0_FRC4	R/W	H'020C 031F	<VDCE1_base> + 1510	32
VDCE1	Output delay control register (SC0)	SC0_SCL0_FRC5	R/W	H'0000 0101	<VDCE1_base> + 1514	32
VDCE1	Full-screen vertical size register (SC0)	SC0_SCL0_FRC6	R/W	H'0023 01E0	<VDCE1_base> + 1518	32
VDCE1	Full-screen horizontal size register (SC0)	SC0_SCL0_FRC7	R/W	H'0090 0280	<VDCE1_base> + 151C	32
VDCE1	Vsync detection register (SC0)	SC0_SCL0_FRC9	R	H'0000 0000	<VDCE1_base> + 1524	32
VDCE1	Status monitor 0 register (SC0)	SC0_SCL0_MON0	R	H'0000	<VDCE1_base> + 1528	16
VDCE1	Interrupt control register (SC0)	SC0_SCL0_INT	R/W	H'0000	<VDCE1_base> + 152A	16
VDCE1	Scaling-down control register (SC0)	SC0_SCL0_DS1	R/W	H'0000 0011	<VDCE1_base> + 152C	32
VDCE1	Vertical capture size register (SC0)	SC0_SCL0_DS2	R/W	H'0012 00F0	<VDCE1_base> + 1530	32
VDCE1	Horizontal capture size register (SC0)	SC0_SCL0_DS3	R/W	H'00F4 05A0	<VDCE1_base> + 1534	32
VDCE1	Horizontal scale down register (SC0)	SC0_SCL0_DS4	R/W	H'1000 2408	<VDCE1_base> + 1538	32
VDCE1	Initial vertical phase register (SC0)	SC0_SCL0_DS5	R/W	H'1800 0000	<VDCE1_base> + 153C	32
VDCE1	Vertical scaling register (SC0)	SC0_SCL0_DS6	R/W	H'0000 07FC	<VDCE1_base> + 1540	32
VDCE1	Scaling-down control block output size register (SC0)	SC0_SCL0_DS7	R/W	H'00F0 0280	<VDCE1_base> + 1544	32
VDCE1	Scaling-up control register (SC0)	SC0_SCL0_US1	R/W	H'0000 0011	<VDCE1_base> + 1548	32
VDCE1	Output image vertical size register (SC0)	SC0_SCL0_US2	R/W	H'0023 01E0	<VDCE1_base> + 154C	32
VDCE1	Output image horizontal size register (SC0)	SC0_SCL0_US3	R/W	H'0090 0280	<VDCE1_base> + 1550	32

Table 38.87 Register Configuration of the Scaler (Channel 1) (2/4)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Scaling-up control block input size register (SC0)	SC0_SCL0_US4	R/W	H'00F0 0280	<VDCE1_base> + 1554	32
VDCE1	Horizontal scale up register (SC0)	SC0_SCL0_US5	R/W	H'0000 2408	<VDCE1_base> + 1558	32
VDCE1	Horizontal scale up initial phase register (SC0)	SC0_SCL0_US6	R/W	H'1000 0000	<VDCE1_base> + 155C	32
VDCE1	Trimming register (SC0)	SC0_SCL0_US7	R/W	H'0000 0000	<VDCE1_base> + 1560	32
VDCE1	Frame buffer read select register (SC0)	SC0_SCL0_US8	R/W	H'0000 0000	<VDCE1_base> + 1564	32
VDCE1	Background color register (SC0)	SC0_SCL0_OVR1	R/W	H'0080 0080	<VDCE1_base> + 156C	32
VDCE1	SCL1 register update control register (SC0)	SC0_SCL1_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1580	32
VDCE1	Writing mode register (SC0)	SC0_SCL1_WR1	R/W	H'0000 0000	<VDCE1_base> + 1588	32
VDCE1	Write address register 1T (SC0)	SC0_SCL1_WR2	R/W	H'0000 0000	<VDCE1_base> + 158C	32
VDCE1	Write address register 2T (SC0)	SC0_SCL1_WR3	R/W	H'0800 0001	<VDCE1_base> + 1590	32
VDCE1	Write address register 3T (SC0)	SC0_SCL1_WR4	R/W	H'0008 0000	<VDCE1_base> + 1594	32
VDCE1	Frame sub-sampling register (SC0)	SC0_SCL1_WR5	R/W	H'0000 1000	<VDCE1_base> + 159C	32
VDCE1	Bit reduction register (SC0)	SC0_SCL1_WR6	R/W	H'0000 0000	<VDCE1_base> + 15A0	32
VDCE1	Write detection register (SC0)	SC0_SCL1_WR7	R	H'0000 0000	<VDCE1_base> + 15A4	32
VDCE1	Write address register 1B (SC0)	SC0_SCL1_WR8	R/W	H'0000 0000	<VDCE1_base> + 15A8	32
VDCE1	Write address register 2B (SC0)	SC0_SCL1_WR9	R/W	H'0800 0001	<VDCE1_base> + 15AC	32
VDCE1	Write address register 3B (SC0)	SC0_SCL1_WR10	R/W	H'0008 0000	<VDCE1_base> + 15B0	32
VDCE1	Write detection register B (SC0)	SC0_SCL1_WR11	R	H'0000 0000	<VDCE1_base> + 15B4	32
VDCE1	Graphics 0 register update control register	GR0_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1600	32
VDCE1	Frame buffer read control register (graphics 0)	GR0_FLM_RD	R/W	H'0000 0000	<VDCE1_base> + 1604	32
VDCE1	Frame buffer control register 1 (graphics 0)	GR0_FLM1	R/W	H'0000 0000	<VDCE1_base> + 1608	32
VDCE1	Frame buffer control register 2 (graphics 0)	GR0_FLM2	R/W	H'0000 0000	<VDCE1_base> + 160C	32
VDCE1	Frame buffer control register 3 (graphics 0)	GR0_FLM3	R/W	H'0800 0001	<VDCE1_base> + 1610	32
VDCE1	Frame buffer control register 4 (graphics 0)	GR0_FLM4	R/W	H'0008 0000	<VDCE1_base> + 1614	32
VDCE1	Frame buffer control register 5 (graphics 0)	GR0_FLM5	R/W	H'0000 03FF	<VDCE1_base> + 1618	32
VDCE1	Frame buffer control register 6 (graphics 0)	GR0_FLM6	R/W	H'8000 0000	<VDCE1_base> + 161C	32

Table 38.87 Register Configuration of the Scaler (Channel 1) (3/4)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Alpha blending control register 1 (graphics 0)	GR0_AB1	R/W	H'0000 0000	<VDCE1_base> + 1620	32
VDCE1	Alpha blending control register 2 (graphics 0)	GR0_AB2	R/W	H'0000 0000	<VDCE1_base> + 1624	32
VDCE1	Alpha blending control register 3 (graphics 0)	GR0_AB3	R/W	H'0000 0000	<VDCE1_base> + 1628	32
VDCE1	Alpha blending control register 7 (graphics 0)	GR0_AB7	R/W	H'00FF 0000	<VDCE1_base> + 1638	32
VDCE1	Alpha blending control register 8 (graphics 0)	GR0_AB8	R/W	H'0000 0000	<VDCE1_base> + 163C	32
VDCE1	Alpha blending control register 9 (graphics 0)	GR0_AB9	R/W	H'0000 0000	<VDCE1_base> + 1640	32
VDCE1	Alpha blending control register 10 (graphics 0)	GR0_AB10	R/W	H'0000 0000	<VDCE1_base> + 1644	32
VDCE1	Alpha blending control register 11 (graphics 0)	GR0_AB11	R/W	H'0000 0000	<VDCE1_base> + 1648	32
VDCE1	Background color control register (graphics 0)	GR0_BASE	R/W	H'0000 8080	<VDCE1_base> + 164C	32
VDCE1	CLUT table control register (graphics 0)	GR0_CLUT	R/W	H'0000 0000	<VDCE1_base> + 1650	32
VDCE1	SCL0 register update control register (SC1)	SC1_SCL0_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1C00	32
VDCE1	Mask control register (SC1)	SC1_SCL0_FRC1	R/W	H'0AF0 0001	<VDCE1_base> + 1C04	32
VDCE1	Missing Vsync compensation control register (SC1)	SC1_SCL0_FRC2	R/W	H'0E10 0001	<VDCE1_base> + 1C08	32
VDCE1	Output sync select register (SC1)	SC1_SCL0_FRC3	R/W	H'0000 0001	<VDCE1_base> + 1C0C	32
VDCE1	Free-running period control register (SC1)	SC1_SCL0_FRC4	R/W	H'020C 031F	<VDCE1_base> + 1C10	32
VDCE1	Output delay control register (SC1)	SC1_SCL0_FRC5	R/W	H'0000 0101	<VDCE1_base> + 1C14	32
VDCE1	Full-screen vertical size register (SC1)	SC1_SCL0_FRC6	R/W	H'0023 01E0	<VDCE1_base> + 1C18	32
VDCE1	Full-screen horizontal size register (SC1)	SC1_SCL0_FRC7	R/W	H'0090 0280	<VDCE1_base> + 1C1C	32
VDCE1	Vsync detection register (SC1)	SC1_SCL0_FRC9	R	H'0000 0000	<VDCE1_base> + 1C24	32
VDCE1	Scaling-down control register (SC1)	SC1_SCL0_DS1	R/W	H'0000 0011	<VDCE1_base> + 1C2C	32
VDCE1	Vertical scaling register (SC1)	SC1_SCL0_DS6	R/W	H'0000 07FC	<VDCE1_base> + 1C40	32
VDCE1	Scaling-up control register (SC1)	SC1_SCL0_US1	R/W	H'0000 0011	<VDCE1_base> + 1C48	32
VDCE1	Output image vertical size register (SC1)	SC1_SCL0_US2	R/W	H'0023 01E0	<VDCE1_base> + 1C4C	32
VDCE1	Output image horizontal size register (SC1)	SC1_SCL0_US3	R/W	H'0090 0280	<VDCE1_base> + 1C50	32
VDCE1	Scaling-up control block input size register (SC1)	SC1_SCL0_US4	R/W	H'00F0 0280	<VDCE1_base> + 1C54	32
VDCE1	Horizontal scale up register (SC1)	SC1_SCL0_US5	R/W	H'0000 2408	<VDCE1_base> + 1C58	32

Table 38.87 Register Configuration of the Scaler (Channel 1) (4/4)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Horizontal scale up initial phase register (SC1)	SC1_SCL0_US6	R/W	H'1000 0000	<VDCE1_base> + 1C5C	32
VDCE1	Trimming register (SC1)	SC1_SCL0_US7	R/W	H'0000 0000	<VDCE1_base> + 1C60	32
VDCE1	Frame buffer read select register (SC1)	SC1_SCL0_US8	R/W	H'0000 0000	<VDCE1_base> + 1C64	32
VDCE1	Background color register (SC1)	SC1_SCL0_OVR1	R/W	H'0080 0080	<VDCE1_base> + 1C6C	32
VDCE1	Graphics 1 register update control register	GR1_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1D00	32
VDCE1	Frame buffer read control register (graphics 1)	GR1_FLM_RD	R/W	H'0000 0000	<VDCE1_base> + 1D04	32
VDCE1	Frame buffer control register 1 (graphics 1)	GR1_FLM1	R/W	H'0000 0000	<VDCE1_base> + 1D08	32
VDCE1	Frame buffer control register 2 (graphics 1)	GR1_FLM2	R/W	H'0000 0000	<VDCE1_base> + 1D0C	32
VDCE1	Frame buffer control register 3 (graphics 1)	GR1_FLM3	R/W	H'0800 0001	<VDCE1_base> + 1D10	32
VDCE1	Frame buffer control register 4 (graphics 1)	GR1_FLM4	R/W	H'0008 0000	<VDCE1_base> + 1D14	32
VDCE1	Frame buffer control register 5 (graphics 1)	GR1_FLM5	R/W	H'0000 03FF	<VDCE1_base> + 1D18	32
VDCE1	Frame buffer control register 6 (graphics 1)	GR1_FLM6	R/W	H'8000 0000	<VDCE1_base> + 1D1C	32
VDCE1	Alpha blending control register 1 (graphics 1)	GR1_AB1	R/W	H'0000 0000	<VDCE1_base> + 1D20	32
VDCE1	Alpha blending control register 2 (graphics 1)	GR1_AB2	R/W	H'0000 0000	<VDCE1_base> + 1D24	32
VDCE1	Alpha blending control register 3 (graphics 1)	GR1_AB3	R/W	H'0000 0000	<VDCE1_base> + 1D28	32
VDCE1	Alpha blending control register 4 (graphics 1)	GR1_AB4	R/W	H'0000 0000	<VDCE1_base> + 1D2C	32
VDCE1	Alpha blending control register 5 (graphics 1)	GR1_AB5	R/W	H'0000 0000	<VDCE1_base> + 1D30	32
VDCE1	Alpha blending control register 6 (graphics 1)	GR1_AB6	R/W	H'0000 0000	<VDCE1_base> + 1D34	32
VDCE1	Alpha blending control register 7 (graphics 1)	GR1_AB7	R/W	H'00FF 0000	<VDCE1_base> + 1D38	32
VDCE1	Alpha blending control register 8 (graphics 1)	GR1_AB8	R/W	H'0000 0000	<VDCE1_base> + 1D3C	32
VDCE1	Alpha blending control register 9 (graphics 1)	GR1_AB9	R/W	H'0000 0000	<VDCE1_base> + 1D40	32
VDCE1	Alpha blending control register 10 (graphics 1)	GR1_AB10	R/W	H'0000 0000	<VDCE1_base> + 1D44	32
VDCE1	Alpha blending control register 11 (graphics 1)	GR1_AB11	R/W	H'0000 0000	<VDCE1_base> + 1D48	32
VDCE1	Background color control register (graphics 1)	GR1_BASE	R/W	H'0000 8080	<VDCE1_base> + 1D4C	32
VDCE1	CLUT table control register (graphics 1)	GR1_CLUT	R/W	H'0000 0000	<VDCE1_base> + 1D50	32
VDCE1	Status monitor register (graphics 1)	GR1_MON	R	H'0000 0000	<VDCE1_base> + 1D54	32

NOTE

Register access sizes other than defined in the table above are not supported.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

38.4.3.1 SCL0 Register Update Control Register (SC0_SCL0_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SC0_SCL0_VEN_D	SC0_SCL0_VEN_C	—	—	—	SC0_SCL0_UPDATE	—	—	—	SC0_SCL0_VEN_B	—	—	—	SC0_SCL0_VEN_A
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/WC1	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	SC0_SCL0_VEN_D	0	R/WC1	Scaling-Up Control and Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
12	SC0_SCL0_VEN_C	0	R/WC1	Scaling-Down Control and Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SC0_SCL0_UPDATE	0	R/WC1	SYNC Control Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_SCL0_VEN_B	0	R/WC1	Synchronization Control and Scaling-up Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_SCL0_VEN_A	0	R/WC1	Scaling-Down Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.4.3.2 Mask Control Register (SC0_SCL0_FRC1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC0_RES_VMASK[15:0]															
Initial value:	0	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_VMASK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SC0_RES_VMASK [15:0]	2800	R/W	Repeated Vsync Signal Masking Period Sets the repeated Vsync signal masking period beginning at a Vsync signal in terms of 128 pixel-clock periods. Masking period [usec] = SC0_RES_VMASK[15:0] × 128 ÷ pixel clock frequency [MHz]
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_ VMASK_ON	1	R/W	Repeated Vsync Signal Masking Control 0: Repeated Vsync signal masking control is disabled. 1: Repeated Vsync signal masking control is enabled.

Note: This register is updated when the SC0_SCL0_UPDATE bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.3 Missing Vsync Compensation Control Register (SC0_SCL0_FRC2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC0_RES_VLACK[15:0]															
Initial value:	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_VLACK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SC0_RES_ VLACK [15:0]	3600	R/W	Missing-Sync Compensating Pulse Output Wait Time Sets the wait time before outputting a missing-sync compensating pulse after a Vsync signal. Wait time [usec] = SC0_RES_VLACK[15:0] × 128 ÷ pixel clock frequency [MHz]
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_ VLACK_ON	1	R/W	Missing Vsync Signal Compensation 0: Compensation of missing Vsync signals is disabled. 1: Compensation of missing Vsync signals is enabled.

Note 1. This register is updated when the SC0_SCL0_UPDATE bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

Note 2. The generation of a missing-sync signal is signalled by the interrupt S0_VSYNCERR, which is listed in Section 38.1.4, Interrupts.

38.4.3.4 Output Sync Select Register (SC0_SCL0_FRC3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SC0_RES_VS_IN_SEL	—	—	—	—	—	—	—	SC0_RES_VS_SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SC0_RES_VS_IN_SEL	0	R/W	Horizontal and Vertical Sync Signal Output and Full-Screen Enable Signal Select Be sure to clear this bit to 0 when cascaded connection is enabled (GR1_AB1.GR1_CUS_CON_ON = 1). 0: Horizontal and vertical sync signal output and full-screen enable signal from scaler 0 1: Horizontal and vertical sync signal output and full-screen enable signal from scaler 1
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_VS_SEL	1	R/W	Vsync Signal Output Select 0: Externally input Vsync signal 1: Internally generated free-running Vsync signal

Note: This register is updated when the SC0_SCL0_UPDATE bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.5 Free-Running Period Control Register (SC0_SCL0_FRC4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_FV[10:0]										
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_FH[10:0]										
Initial value:	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_FV [10:0]	524	R/W	Free-Running Vsync Period Setting Free-running Vsync period = (SC0_RES_FV[10:0] + 1) × horizontal period [usec]
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_FH [10:0]	799	R/W	Hsync Period Setting Hsync period [usec] = (SC0_RES_FH[10:0] +1) ÷ pixel clock frequency [MHz]

Note: This register is updated when the SC0_SCL0_UPDATE bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.6 Output Delay Control Register (SC0_SCL0_FRC5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SC0_RES_FLD_DLY_SEL	SC0_RES_VSDLY[7:0]							
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SC0_RES_ FLD_DLY_SEL	1	R/W	Field Determination Signal Delay Control 0: No delay 1: Delay of one vertical cycle
7 to 0	SC0_RES_ VSDLY[7:0]	1	R/W	Vsync Signal Delay Control Adjusts the Vsync signal delay in the output Hsync period units. Vsync signal delay [usec]: SC0_RES_VSDLY[7:0] × output Hsync period [usec]

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.7 Full-Screen Vertical Size Register (SC0_SCL0_FRC6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_F_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_F_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_F_VS[10:0]	35	R/W	Vertical Enable Signal Start Position for Full Screen. (VSYNC + V backporch lines) Note: The set value should be four or more (lines). SC0_RES_F_VS[10:0] + SC0_RES_F_VW[10:0] should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_F_VW[10:0]	480	R/W	Vertical Enable Signal Width for Full Screen (lines) Note: SC0_RES_F_VS[10:0] + SC0_RES_F_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.8 Full-Screen Horizontal Size Register (SC0_SCL0_FRC7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_F_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_F_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_F_HS[10:0]	144	R/W	Horizontal Enable Signal Start Position for Full Screen. (HSYNC+H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). SC0_RES_F_HS[10:0] + SC0_RES_F_HW[10:0] should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_F_HW[10:0]	640	R/W	Horizontal Enable Signal Width for Full Screen (pixel-clock cycles) **Notes:** 1. SC0_RES_F_HS[10:0] + SC0_RES_F_HW[10:0] should be equal to or less than 2015 (clock cycles). 2. In the case of video output format is Serial RGB (OUT_SET.OUT_FORMAT[1:0] = 3), it is nessesary to add 2 to the value in SC0_RES_F_HW.

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.9 Vsync Detection Register (SC0_SCL0_FRC9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_QVLOCK	—	—	—	SC0_RES_QVLACK
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_RES_QVLOCK	0	R	Locked Vsync Signal Detection Flag 1: Repeated and missing Vsync signal input has not been detected for four or more vertical periods. 0: Repeated or missing Vsync signal input has been detected.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_QVLACK	0*1	R	Missing Vsync Signal Detection Flag 1: Missing Vsync signal input has been detected. 0: Missing Vsync signal input has not been detected.

Note 1. SC0_RES_QVLACK changes immediately from 0 to 1, if the video output clock is enabled. It changes back to 0 after detection of the first VSync. So interrupt on missing Vsync shall be enabled only from the second Vsync onwards.

38.4.3.10 Status Monitor 0 Register (SC0_SCL0_MON0)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_LIN_STAT[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_LIN_STAT[10:0]	All 0	R	Current location of the image line input to the scaling-down control block.

38.4.3.11 Interrupt Control Register (SC0_SCL0_INT)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_LINE[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_LINE[10:0]	All 0	R/W	Setting of Interrupt on Image Line Input to Scaling-down Control Block When the location of the image line input to the scaling-down control block matches the SC0_RES_LINE[10:0] setting, an interrupt signal is output.

Note: This register is updated when the SC0_SCL0_VEN_A bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.12 Scaling-Down Control Register (SC0_SCL0_DS1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_DS_V_ON	—	—	—	SC0_RES_DS_H_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_RES_DS_V_ON	1	R/W	Vertical Scale Down On/Off 0: Off 1: On **Note:** For D1L2(H) devices this bit must be set to 0.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_DS_H_ON	1	R/W	Horizontal Scale Down On/Off 0: Off 1: On **Note:** For D1L2(H) devices this bit must be set to 0.

Note: This register is updated when the SC0_SCL0_VEN_A bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.13 Vertical Capture Size Register (SC0_SCL0_DS2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_VS [10:0]	18	R/W	Vertical Position Setting for Video Signal Capturing (VSYNC + (V backporch - 1) lines) Note: The set value should be four or more (lines). SC0_RES_VS[10:0] + SC0_RES_VW[10:0] should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_VW [10:0]	240	R/W	Vertical Width of Video Signal to be Captured (Lines) Note: SC0_RES_VS[10:0] + SC0_RES_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when the SC0_SCL0_VEN_A bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.14 Horizontal Capture Size Register (SC0_SCL0_DS3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_HW[10:0]										
Initial value:	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_HS [10:0]	244	R/W	Horizontal Position Setting for Video Signal Capturing (HSYNC + H backporch video-image clock cycles) Note: The set value should be 16 or more (clock cycles). SC0_RES_HS[10:0] + SC0_RES_HW[10:0] should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_HW [10:0]	1440	R/W	Horizontal Width of Video Signal to be Captured (Video-image clock cycles) Note: SC0_RES_HS[10:0] + SC0_RES_HW[10:0] should be equal to or less than 2015 (clock cycles).

Note: This register is updated when the SC0_SCL0_VEN_A bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.15 Horizontal Scale Down Register (SC0_SCL0_DS4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	SC0_RES_PFIL_SEL	SC0_RES_DS_H_INTERPOTYP	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_DS_H_RATIO[15:0]															
Initial value:	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31, 30	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
29	SC0_RES_PFIL_SEL	0	R/W	Prefilter Mode Select for Brightness Signals 0: The prefilter is turned off. 1: The prefilter is turned on. (1/4 + 1/2 + 1/4)
28	SC0_RES_DS_H_INTERPOTYP	1	R/W	Horizontal Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
27 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	SC0_RES_DS_H_RATIO[15:0]	9224	R/W	Horizontal Scale Down Ratio ([15:12]: Integer part, [11:0]: Decimal part) round(SC0_RES_HW[10:0] ÷ SC0_RES_OUT_HW[10:0] × 4096) SC0_RES_DS_H_RATIO[15:0] < 4096: Setting prohibited SC0_RES_DS_H_RATIO[15:0] = 4096: 100% scale up SC0_RES_DS_H_RATIO[15:0] > 4096: Scale down

Note: This register is updated when the SC0_SCL0_VEN_A bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.16 Initial Vertical Phase Register (SC0_SCL0_DS5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	SC0_RES_V_INTERPOTYP	SC0_RES_TOP_INIPHASE[11:0]											
Initial value:	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SC0_RES_BTM_INIPHASE[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	SC0_RES_V_INTERPOTYP	1	R/W	Vertical Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
27 to 16	SC0_RES_TOP_INIPHASE [11:0]	2048	R/W	Vertical Interpolation Start Phase for Top Field 0 to 4095 (0 to approx. 1.0)
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	SC0_RES_BTM_INIPHASE [11:0]	0	R/W	Vertical Interpolation Start Phase for Bottom Field 0 to 4095 (0 to approx. 1.0)

Note: This register is updated when the SC0_SCL0_VEN_A and SC0_SCL0_VEN_B bits in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) are 1.

38.4.3.17 Vertical Scaling Register (SC0_SCL0_DS6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_V_RATIO[15:0]															
Initial value:	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	SC0_RES_V_RATIO [15:0]	2044	R/W	Vertical Scale Up/Down Ratio ([15:12]: Integer part, [11:0]: Decimal part) For scale down: round (SC0_RES_VW[10:0] ÷ SC0_RES_OUT_VW[10:0] × 4096) For scale up: round (SC0_RES_IN_VW[10:0] ÷ SC0_RES_P_VW[10:0] × 4096) SC0_RES_V_RATIO[15:0] < 4096: Scale up SC0_RES_V_RATIO[15:0] = 4096: 100% scale up SC0_RES_V_RATIO[15:0] > 4096: Scale down

Note: These bits updated when the SC0_SCL0_VEN_A and SC0_SCL0_VEN_B bits in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) are 1. Accordingly, even a scaled-up graphics display requires both an input Vsync signal and output Vsync signal.

38.4.3.18 Scaling-Down Control Block Output Size Register (SC0_SCL0_DS7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_OUT_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_OUT_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_OUT_VW[10:0]	240	R/W	Number of Valid Lines in Vertical Direction Output by Scaling-down Control Block (lines) This bit setting is used for the number of lines to be written to the frame buffer. When SC0_SCL1_WR1.SC0_RES_LOOP is 0 (frame write mode), specify the number of lines for one frame. When SC0_SCL1_WR1.SC0_RES_LOOP is 1 (line write mode), specify the number of lines for repeated write. Note: The SC0_RES_OUT_VW[10:0] value should be aligned in 4-line units and equal to or smaller than the SC0_RES_VW[10:0] value.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_OUT_HW[10:0]	640	R/W	Number of Valid Horizontal Pixels Output by Scaling-Down Control Block (video-image clock cycles) Note: The SC0_RES_OUT_HW[10:0] value should be aligned in 4-pixel units and equal to or smaller than the SC0_RES_HW[10:0] value.

Note: This register is updated when the SC0_SCL0_VEN_A and SC0_SCL0_VEN_C bits in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) are 1.

38.4.3.19 Scaling-Up Control Register (SC0_SCL0_US1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_US_V_ON	—	—	—	SC0_RES_US_H_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_RES_US_V_ON	1	R/W	Vertical Scale Up On/Off 0: Off 1: On **Note:** For D1L2(H) devices this bit must be set to 0.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_US_H_ON	1	R/W	Horizontal Scale Up On/Off 0: Off 1: On **Note:** For D1L2(H) devices this bit must be set to 0.

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.20 Output Image Vertical Size Register (SC0_SCL0_US2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_P_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_P_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_P_VS[10:0]	35	R/W	Vertical Enable Signal Start Position for Output Image (VSYNC + V backporch lines) Note: The set value should be four or more (lines). SC0_RES_P_VS[10:0] + SC0_RES_P_VW[10:0] should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_P_VW[10:0]	480	R/W	Vertical Enable Signal Width for Output Image (lines) Note: SC0_RES_P_VS[10:0] + SC0_RES_P_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.21 Output Image Horizontal Size Register (SC0_SCL0_US3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_P_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_P_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_P_HS[10:0]	144	R/W	Horizontal Enable Signal Start Position for Output Image (HSYNC+H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). SC0_RES_P_HS[10:0] + SC0_RES_P_HW[10:0] should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_P_HW[10:0]	640	R/W	Horizontal Enable Signal Width for Output Image (pixel-clock cycles) Note: SC0_RES_P_HS[10:0] + SC0_RES_P_HW[10:0] should be equal to or less than 2015 (clock cycles).

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.22 Scaling-Up Control Block Input Size Register (SC0_SCL0_US4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC0_RES_IN_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC0_RES_IN_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC0_RES_IN_VW[10:0]	240	R/W	Number of Valid Lines in Vertical Direction Input to Scaling-down Control Block (lines)
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC0_RES_IN_HW[10:0]	640	R/W	Number of Valid Horizontal Pixels Input to Scaling-down Control Block (pixel-clock cycles)

Note: This register is updated when the SC0_SCL0_VEN_B and SC0_SCL0_VEN_D bits in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) are 1.

38.4.3.23 Horizontal Scale Up Register (SC0_SCL0_US5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_US_H_RATIO[15:0]															
Initial value:	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	SC0_RES_US_H_RATIO [15:0]	9224*1	R/W	Horizontal Scale Up Ratio ([15:12]: Integer part, [11:0]: Decimal part) round (SC0_RES_IN_HW[10:0] ÷ SC0_RES_P_HW[10:0] × 4096) SC0_RES_US_H_RATIO[15:0] < 4096: Scale up SC0_RES_US_H_RATIO[15:0] = 4096: 100% scale up SC0_RES_US_H_RATIO[15:0] > 4096: Setting prohibited

Note 1. Initial value is in invalid range. When up-scaling is used, this value must be changed to the proper value.

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.24 Horizontal Scale Up Initial Phase Register (SC0_SCL0_US6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	SC0_RES_US_H_INTERPOTYP	SC0_RES_US_HT_INIPHASE[11:0]											
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SC0_RES_US_HB_INIPHASE[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	SC0_RES_US_H_INTERPOTYP	1	R/W	Horizontal Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
27 to 16	SC0_RES_US_HT_INIPHASE[11:0]	0	R/W	Horizontal Interpolation Start Phase for Top Field 0 to 4095 (0 to approx. 1.0)
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	SC0_RES_US_HB_INIPHASE[11:0]	0	R/W	Horizontal Interpolation Start Phase for Bottom Field 0 to 4095 (0 to approx. 1.0)

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.25 Trimming Register (SC0_SCL0_US7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_HCUT[7:0]								SC0_RES_VCUT[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 8	SC0_RES_HCUT[7:0]	0	R/W	Horizontal Amount of Cut-off Post-Scaling Image (Right and Left Parts) Sets the number of pixel-clock cycles.
7 to 0	SC0_RES_VCUT[7:0]	0	R/W	Vertical Amount of Cut-off Post-Scaling Image (Upper and Lower Parts) Sets the number of lines.

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.26 Frame Buffer Read Select Register (SC0_SCL0_US8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_IBUS_SYNC_SEL	—	—	—	SC0_RES_DISP_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_RES_IBUS_SYNC_SEL	0	R/W	Sync Signal Select for Frame Buffer Read Block 0: Sync signals from the scaling-up control block 1: Sync signals from the graphics processing block
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_DISP_ON	0	R/W	Post-Scaling Image Frame Display On/Off 0: Frame display off 1: Frame display on

Note: SC0_RES_IBUS_SYNC_SEL is updated when the SC0_SCL0_VEN_B and SC0_SCL0_VEN_D bits in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) are 1.
SC0_RES_DISP_ON is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.27 Background Color Register (SC0_SCL0_OVR1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SC0_RES_BK_COL_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_BK_COL_G[7:0]								SC0_RES_BK_COL_B[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	SC0_RES_BK_COL_R[7:0]	128	R/W	Background Color Setting R/Cr Signal R: 8 bits; unsigned (0 to 255 [LSB]) Cr: 8 bits; 128 offset binary; unsigned (0 to 255 [LSB])
15 to 8	SC0_RES_BK_COL_G[7:0]	0	R/W	Background Color Setting G/Y Signal G/Y: 8 bits; unsigned (0 to 255 [LSB])
7 to 0	SC0_RES_BK_COL_B[7:0]	128	R/W	Background Color Setting B/Cb Signal B: 8 bits; unsigned (0 to 255 [LSB]) Cb: 8 bits; 128 offset binary; unsigned (0 to 255 [LSB])

Note: This register is updated when the SC0_SCL0_VEN_B bit in the SC0_SCL0 register update control register (SC0_SCL0_UPDATE) is 1.

38.4.3.28 SCL1 Register Update Control Register (SC0_SCL1_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	SC0_SCL1_UPDATE_B	—	—	—	SC0_SCL1_UPDATE_A
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC0_SCL1_VEN_B	—	—	—	SC0_SCL1_VEN_A
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	SC0_SCL1_UPDATE_B	0	R/WC1	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated.
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	SC0_SCL1_UPDATE_A	0	R/WC1	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated.
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_SCL1_VEN_B	0	R/WC1	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_SCL1_VEN_A	0	R/WC1	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.4.3.29 Writing Mode Register (SC0_SCL1_WR1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_WRSWA [2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SC0_RES_TB_ADD_MOD	SC0_RES_DS_WR_MD[2:0]			SC0_RES_MD[1:0]		SC0_RES_LOOP	Caution *
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

CAUTION

The initial value “0” of bit 0 must be changed to “1”.

Bit	Bit Name	Initial Value	R/W	Description
31 to 19	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
18 to 16	SC0_RES_WRSWA[2:0]	All 0	R/W	8-Bit, 16-Bit, or 32-Bit Swap Setting These bits control swapping in frame buffer writing as follows. Bit 0 0: Not swapped in 8-bit units. 1: Swapped in 8-bit units. Bit 1 0: Not swapped in 16-bit units. 1: Swapped in 16-bit units. Bit 2 0: Not swapped in 32-bit units. 1: Swapped in 32-bit units. According to the setting of these bits, data is swapped as follows. Each number in parentheses ((1) to (8)) indicates 8-bit data. 000: (1) (2) (3) (4) (5) (6) (7) (8) [Not swapped] 001: (2) (1) (4) (3) (6) (5) (8) (7) [Swapped in 8-bit units] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Swapped in 16-bit units] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Swapped in 16-bit units + 8-bit units] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Swapped in 32-bit units] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Swapped in 32-bit units + 8-bit units] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Swapped in 32-bit units + 16-bit units] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Swapped in 32-bit units + 16-bit units + 8-bit units]
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7	SC0_RES_TB_ADD_MOD	0	R/W	Top and Bottom Data Write Address Specification Method 0: A write address is specified in common for top and bottom data. 1: Separate write addresses are specified for top and bottom data.
6 to 4	SC0_RES_DS_WR_MD[2:0]	0	R/W	Frame Buffer Writing Mode for Image Processing 0: Normal 1: Horizontal mirroring 2 to 7: Setting prohibited

Bit	Bit Name	Initial Value	R/W	Description
3, 2	SC0_RES_MD [1:0]	0	R/W	Frame Buffer Video-Signal Writing Format 0: YCbCr422 (16 bits) 1: RGB565 (16 bits) 2: RGB888 (24 (32) bits) 3: YCbCr444 (24 (32) bits)
1	SC0_RES_ LOOP	0	R/W	Frame Buffer Write Mode Select 0: Frame mode 1: Line mode (read as ring buffer)
0	Bit0	0	R/W	**CAUTION** The initial value “0” of this bit must be changed to “1”.

Note: SC0_RES_LOOP is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.
SC0_RES_TB_ADD_MOD, SC0_RES_DS_WR_MD[2:0], and SC0_RES_MD[1:0] are updated when the SC0_SCL1_VEN_A and SC0_SCL1_VEN_B bits in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) are 1.
SC0_RES_WRSWA[2:0] is updated when the SC0_SCL1_UPDATE_A bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

38.4.3.30 Write Address Register 1T (SC0_SCL1_WR2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC0_RES_BASE[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_BASE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SC0_RES_ BASE [31:0]	0	R/W	Frame Buffer Base Address Sets the start address of the frame buffer to store the frame data for the top field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 0. The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

38.4.3.31 Write Address Register 2T (SC0_SCL1_WR3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	SC0_RES_LN_OFF[14:0]														
Initial value:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SC0_RES_FLM_NUM[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 16	SC0_RES_LN_OFF[14:0]	2048	R/W	Frame Buffer Line Offset Address Sets the line offset address for calculating the line start address for the top field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 0. Line 0: SC0_RES_BASE[31:0] Line 1: SC0_RES_BASE[31:0] + SC0_RES_LN_OFF[14:0] × 1 : Line n: SC0_RES_BASE[31:0] + SC0_RES_LN_OFF[14:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	SC0_RES_FLM_NUM[9:0]	1	R/W	Number of Frames of Buffer to be Written to Sets the number of frames for the top field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 0 Number of frames defined by SC0_RES_FLM_NUM[9:0] + 1 are used.

Note: This register is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

38.4.3.32 Write Address Register 3T (SC0_SCL1_WR4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	SC0_RES_FLM_OFF[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_FLM_OFF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	SC0_RES_FLM_OFF[22:0]	524288	R/W	Frame Buffer Frame Offset Address Sets the frame offset address for calculating the start address of each frame for the top field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1 or that for the top and bottom fields when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 0. Buffer 0: SC0_RES_BASE[31:0] Buffer 1: SC0_RES_BASE[31:0] + SC0_RES_FLM_OFF[22:0] × 1 : Buffer n: SC0_RES_BASE[31:0] + SC0_RES_FLM_OFF[22:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

38.4.3.33 Frame Sub-Sampling Register (SC0_SCL1_WR5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	SC0_RES_INTER	—	—	SC0_RES_FS_RATE[1:0]		—	—	—	SC0_RES_FLD_SEL	—	—	—	SC0_RES_WENB
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	SC0_RES_INTER	1	R/W	Field Operating Mode Select 0: Progressive 1: Interlace
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	SC0_RES_FS_RATE[1:0]	0	R/W	Writing Rate Sets the frame buffer writing rate to the vertical frequency of the input signal. 0: 1/1 an input signal (The SC0_RES_FLD_SEL setting is invalid.) 1: 1/2 an input signal 2: 1/4 an input signal 3: 1/8 an input signal
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_RES_FLD_SEL	0	R/W	Write Field Select 0: Top field 1: Bottom field
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_WENB	0	R/W	Frame Buffer Write Enable After making the setting to enable writing, writing starts from the second frame. 0: Frame buffer writing is disabled. 1: Frame buffer writing is enabled.

Note: SC0_RES_INTER, SC0_RES_FS_RATE[1:0], and SC0_RES_FLD_SEL are updated when the SC0_SCL1_VEN_A bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1. SC0_RES_WENB is updated when the SC0_SCL1_VEN_A and SC0_SCL1_VEN_B bits in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) are 1.

38.4.3.34 Bit Reduction Register (SC0_SCL1_WR6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_DTH_ON	—	—	—	SC0_RES_BITDEC_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC0_RES_DTH_ON	0	R/W	Dither Correction On/Off 0: Off (rounded off) 1: On (2 × 2 dither pattern)
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC0_RES_BITDEC_ON	0	R/W	Bit Reduction On/Off 0: Off 1: On

Note: This register is updated when the SC0_SCL1_VEN_A bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

38.4.3.35 Write Detection Register (SC0_SCL1_WR7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SC0_RES_OVERFLOW
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SC0_RES_FLM_CNT[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	SC0_RES_OVERFLOW	0	R	Line Buffer Overflow Detect 1: Line buffer has overflowed. 0: Line buffer has not overflowed.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	SC0_RES_FLM_CNT[9:0]	0	R	Frame Number Before Frame Being Accessed Frame number before the frame being accessed in the top field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1 or that in the top or bottom field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 0.

38.4.3.36 Write Address Register 1B (SC0_SCL1_WR8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC0_RES_BASE_B[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_BASE_B[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	SC0_RES_BASE_B [31:0]	0	R/W	Frame Buffer Base Address for Bottom Sets the start address of the frame buffer to store the frame data for the bottom field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1. The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

38.4.3.37 Write Address Register 2B (SC0_SCL1_WR9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	SC0_RES_LN_OFF_B[14:0]														
Initial value:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SC0_RES_FLM_NUM_B[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 16	SC0_RES_LN_OFF_B [14:0]	2048	R/W	Frame Buffer Line Offset Address for Bottom Sets the line offset address for calculating the line start address for the bottom field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1. Line 0: SC0_RES_BASE_B[31:0] Line 1: SC0_RES_BASE_B[31:0] + SC0_RES_LN_OFF_B[14:0] × 1 : Line n: SC0_RES_BASE_B[31:0] + SC0_RES_LN_OFF_B[14:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	SC0_RES_FLM_NUM_B [9:0]	1	R/W	Number of Frames of Buffer to be Written to for Bottom Field Number of frames defined by SC0_RES_FLM_NUM_B[9:0] + 1 are used when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1.

Note: This register is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

38.4.3.38 Write Address Register 3B (SC0_SCL1_WR10)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	SC0_RES_FLM_OFF_B[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC0_RES_FLM_OFF_B[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	SC0_RES_FLM_OFF_B[22:0]	524288	R/W	Frame Buffer Frame Offset Address for Bottom Sets the frame offset address for calculating the start address of each frame for the bottom field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1. Buffer 0: SC0_RES_BASE_B[31:0] Buffer 1: SC0_RES_BASE_B[31:0] + SC0_RES_FLM_OFF_B[22:0] × 1 : Buffer n: SC0_RES_BASE_B[31:0] + SC0_RES_FLM_OFF_B[22:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when the SC0_SCL1_VEN_B bit in the SC0_SCL1 register update control register (SC0_SCL1_UPDATE) is 1.

38.4.3.39 Write Detection Register B (SC0_SCL1_WR11)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	SC0_RES_FLM_CNT_B[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	SC0_RES_FLM_CNT_B[9:0]	0	R	Frame Number Before Frame Being Accessed in Bottom Field Frame number before the frame being accessed in the bottom field when SC0_SCL1_WR1.SC0_RES_TB_ADD_MOD = 1.

38.4.3.40 Graphics 0 Register Update Control Register (GR0_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GR0_UPDATE	—	—	—	GR0_P_VEN	—	—	—	GR0_IBUS_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR0_UPDATE	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR0_P_VEN	0	R/WC1	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR0_IBUS_VEN	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.4.3.41 Frame Buffer Read Control Register (Graphics 0) (GR0_FLM_RD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR0_R_ENB
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR0_R_ENB	0	R/W	Frame Buffer Read Enable 0: Frame buffer reading is disabled. 1: Frame buffer reading is enabled.

Note: This register is updated when the GR0_IBUS_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

38.4.3.42 Frame Buffer Control Register 1 (Graphics 0) (GR0_FLM1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_FLD_SEL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR0_LN_OFF_DIR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR0_FLM_SEL[1:0]		—	—	—	—	—	—	—	Caution *
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

CAUTION

The initial value “0” of bit 0 must be changed to “1”.

Bit	Bit Name	Initial Value	R/W	Description
31	GR0_FLD_SEL	0	R/W	Enables or disables top or bottom field selection for the next frame buffer to be read. 0: Field selection is disabled. 1: Top or bottom field can be selected.
30 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR0_LN_OFF_DIR	0	R/W	Selects the line offset address direction of the frame buffer. 0: Increments the address by the line offset address. 1: Decrements the address by the line offset address.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	GR0_FLM_SEL[1:0]	0	R/W	Selects a frame buffer address setting signal. 0: Links to scaling-down process. (This setting is prohibited when separate write addresses are specified for the top and bottom fields; that is, SC0_RES_TB_ADD_MOD = 1 in SC0_SCL1_WR1.) 1: Selects GR0_FLM_NUM[9:0]. 2: prohibited 3: prohibited
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	Bit0	0	R/W	**CAUTION** The initial value “0” of this bit must be changed to “1”.

Note: GR0_FLD_SEL is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.
GR0_LN_OFF_DIR is updated when the GR0_IBUS_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.
GR0_FLM_SEL[1:0] is updated when the GR0_P_VEN and GR0_IBUS_VEN bits in the graphics 0 register update control register (GR0_UPDATE) are 1.

38.4.3.43 Frame Buffer Control Register 2 (Graphics 0) (GR0_FLM2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_BASE[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_BASE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	GR0_BASE [31:0]	0	R/W	Frame Buffer Base Address Sets the start address of the frame buffer where frame data is to be stored. The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when the GR0_IBUS_VEN and GR0_P_VEN bits in the graphics 0 register update control register (GR0_UPDATE) are 1.

38.4.3.44 Frame Buffer Control Register 3 (Graphics 0) (GR0_FLM3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_FLD_NXT	GR0_LN_OFF[14:0]														
Initial value:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR0_FLM_NUM[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	GR0_FLD_NXT	0	R/W	Top or Bottom Field Selection for Next Frame Buffer 0: Bottom 1: Top
30 to 16	GR0_LN_ OFF[14:0]	2048	R/W	Frame Buffer Line Offset Address Sets the line offset address for calculating the start address of each line. Line 0: GR0_BASE[31:0] Line 1: GR0_BASE[31:0] + GR0_LN_OFF[14:0] × 1 : Line n: GR0_BASE[31:0] + GR0_LN_OFF[14:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GR0_FLM_ NUM[9:0]	1	R/W	Frame Number of Frame Buffer Manually set the frame number when GR0_FLM_SEL[1:0] = 1.

Note: GR0_FLD_NXT is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.
GR0_LN_OFF[14:0] and GR0_FLM_NUM[9:0] are updated when the GR0_IBUS_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

38.4.3.45 Frame Buffer Control Register 4 (Graphics 0) (GR0_FLM4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	GR0_FLM_OFF[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_FLM_OFF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	GR0_FLM_OFF[22:0]	524288	R/W	Frame Buffer Frame Offset Address Specifies the frame offset address used for calculating the start address of each frame buffer when more than one buffer is used. Buffer 0: GR0_BASE[31:0] Buffer 1: GR0_BASE[31:0] + GR0_FLM_OFF[22:0] × 1 : Buffer n: GR0_BASE[31:0] + GR0_FLM_OFF[22:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when the GR0_IBUS_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

38.4.3.46 Frame Buffer Control Register 5 (Graphics 0) (GR0_FLM5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR0_FLM_LNUM[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR0_FLM_LOOP[10:0]										
Initial value:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR0_FLM_LNUM[10:0]	0	R/W	Sets number of lines in a frame Number of lines is (GR0_FLM_LNUM[10:0] + 1).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR0_FLM_LOOP[10:0]	1023	R/W	Number of lines when reading the addresses repeatedly by returning to the start address after reaching the end address. (GR0_FLM_LOOP[10:0] + 1) lines are read.

Note: This register is updated when the GR0_IBUS_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

38.4.3.47 Frame Buffer Control Register 6 (Graphics 0) (GR0_FLM6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_FORMAT[3:0]				—	GR0_HW[10:0]										
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_YCC_SWAP[2:0]			GR0_RDSWA[2:0]			—	GR0_CNV444_MD	—	—	GR0_STA_POS[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	GR0_FORMAT [3:0]	8	R/W	Sets the format of the frame buffer read signal. 0: RGB565 1: RGB888 2: αRGB1555 3: αRGB4444 4: αRGB8888 5: CLUT8 6: CLUT4 7: CLUT1 8: YCbCr422 9: YCbCr444 10: RGBα5551 11: RGBα8888 12 to 15: Setting prohibited
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 16	GR0_HW [10:0]	0	R/W	Sets the width of the horizontal valid period. The width is (GR0_HW[10:0] + 1) pixels. Note: The set value should be equal to or more than two.
15 to 13	GR0_YCC_SWAP[2:0]	0	R/W	Controls swapping of data read from buffer in the YCbCr422 format. 0: Cb/Y0/Cr/Y1 1: Y0/Cb/Y1/Cr 2: Cr/Y0/Cb/Y1 3: Y0/Cr/Y1/Cb 4: Y1/Cr/Y0/Cb 5: Cr/Y1/Cb/Y0 6: Y1/Cb/Y0/Cr 7: Cb/Y1/Cr/Y0

Bit	Bit Name	Initial Value	R/W	Description
12 to 10	GR0_ RDSWA [2:0]	0	R/W	8-Bit, 16-Bit, or 32-Bit Swap Setting These bits control swapping in frame buffer reading as follows. Bit 0 0: Swapped in 8-bit units. 1: Not swapped in 8-bit units. Bit 1 0: Swapped in 16-bit units. 1: Not swapped in 16-bit units. Bit 2 0: Swapped in 32-bit units. 1: Not swapped in 32-bit units. According to the setting of these bits, data is swapped as follows. Each number in parentheses ((1) to (8)) indicates 8-bit data. 000: (1) (2) (3) (4) (5) (6) (7) (8) [Not swapped] 001: (2) (1) (4) (3) (6) (5) (8) (7) [Swapped in 8-bit units] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Swapped in 16-bit units] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Swapped in 16-bit units + 8-bit units] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Swapped in 32-bit units] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Swapped in 32-bit units + 8-bit units] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Swapped in 32-bit units + 16-bit units] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Swapped in 32-bit units + 16-bit units + 8-bit units]
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	GR0_ CNV444_ MD	0	R/W	Sets the interpolation mode for YCbCr422 to YCbCr444 conversion. 0: Hold interpolation 1: Average interpolation
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	GR0_STA_ POS[5:0]	0	R/W	Sets the amount of data to be skipped through. Specifically data amount equal to the amount indicated by GR0_STA_POS[5:0] is skipped from the start of the line.

Note: GR0_YCC_SWAP[2:0], GR0_CNV444_MD, and GR0_STA_POS[5:0] are updated when GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.
GR0_ RDSWA is updated when the GR0_UPDATE bit in the graphics 0 register update control register (GR0_UPDATE) is 1.
GR0_FORMAT[3:0] and GR0_HW[10:0] are updated when GR0_IBUS_VEN and GR0_P_VEN bits in the graphics 0 register update control register (GR0_UPDATE) are 1.

38.4.3.48 Alpha Blending Control Register 1 (Graphics 0) (GR0_AB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	GR0_GRC_DISP_ON	—	—	GR0_DISP_SEL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR0_GRC_DISP_ON	0	R/W	Turns on/off frame-line display of the graphics image area. 0: Frame-line display off 1: Frame-line display on
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	GR0_DISP_SEL[1:0]	0	R/W	Selects the graphics display mode. 0: Background color display (GR0_BASE[31:0]) 1: Lower-layer graphics display When displaying video image or enlarged graphics, select this setting. 2: Current graphics display When displaying graphics, select this setting. 3: Blended display of lower-layer graphics and current graphics* Note:* Select this setting whenever chroma-key processing is to proceed. Since only current graphics are to be displayed by chroma-key processing, set the α values for both pixels to be subject to chroma-keying and pixels not to be subject to chroma-keying to 255.

Note: This register is updated when GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

NOTE

Alpha blending and Chroma-Key features are not supported for Graphic Layer 0 (GR0).

38.4.3.49 Alpha Blending Control Register 2 (Graphics 0) (GR0_AB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR0_GRC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR0_GRC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR0_GRC_VS[10:0]	0	R/W	Vertical Start Position of Graphics Image Area. **NOTE** The set value should be four or more (lines). GR0_GRC_VS[10:0] + GR0_GRC_VW[10:0] should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR0_GRC_VW[10:0]	0	R/W	Vertical Width of Graphics Image Area. **NOTE** GR0_GRC_VS[10:0] + GR0_GRC_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

38.4.3.50 Alpha Blending Control Register 3 (Graphics 0) (GR0_AB3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR0_GRC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR0_GRC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR0_GRC_HS[10:0]	0	R/W	Horizontal Start Position of Graphics Image Area. Note: The set value should be 16 or more (clock cycles). GR0_GRC_HS[10:0] + GR0_GRC_HW[10:0] should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR0_GRC_HW[10:0]	0	R/W	Horizontal Width of Graphics Image Area. Note: For displaying an image with 1- or 2-pixel horizontal width, set GR0_HW[10:0] to 2 and GR0_GRC_HW[10:0] to 1 (1 pixel) or 2 (2 pixels). **NOTE** GR0_GRC_HS[10:0] + GR0_GRC_HW[10:0] should be equal to or less than 2015 (clock cycles).

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

38.4.3.51 Alpha Blending Control Register 7 (Graphics 0) (GR0_AB7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR0_CK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR0_CK_ON	0	R/W	CLUT-Index/RGB-Index Chroma-Key Processing On/Off 0: Off 1: On

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

NOTE

Alpha blending and Chroma-Key features are not supported for Graphic Layer 0 (GR0).

38.4.3.52 Alpha Blending Control Register 8 (Graphics 0) (GR0_AB8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_CK_KCLUT[7:0]								GR0_CK_KG[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_CK_KB[7:0]								GR0_CK_KR[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR0_CK_ KCLUT[7:0]	0	R/W	CLUT Signal for CLUT-Index Chroma-Key Processing CLUT: Unsigned 8 bits (0 to 255 [LSB])
23 to 16	GR0_CK_ KG[7:0]	0	R/W	G Signal for RGB-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR0_CK_ KB[7:0]	0	R/W	B Signal for RGB-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR0_CK_ KR[7:0]	0	R/W	R Signal for RGB-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

NOTE

Alpha blending and Chroma-Key features are not supported for Graphic Layer 0 (GR0).

38.4.3.53 Alpha Blending Control Register 9 (Graphics 0) (GR0_AB9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_CK_A[7:0]								GR0_CK_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_CK_B[7:0]								GR0_CK_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR0_CK_A [7:0]	0	R/W	Replaced Alpha Signal after RGB-Index Chroma-Key Processing α: Unsigned 8 bits (0 to 255 [LSB]) Note: These bits should always be set to 255 to display the current graphics only.
23 to 16	GR0_CK_G [7:0]	0	R/W	Replaced G Signal after RGB-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR0_CK_B [7:0]	0	R/W	Replaced B Signal after RGB-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR0_CK_R [7:0]	0	R/W	Replaced R Signal after RGB-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

38.4.3.54 Alpha Blending Control Register 10 (Graphics 0) (GR0_AB10)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_A0[7:0]								GR0_G0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_B0[7:0]								GR0_R0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR0_A0 [7:0]	0	R/W	CLUT0 α0 Signal Replaced with α signal when in the CLUT0 format and CLUT0 = 0. Replaced with α signal when in the αRGB1555/RGBα5551 format and α = 0. Note: These bits should always be set to 255 to display the current graphics only.
23 to 16	GR0_G0 [7:0]	0	R/W	CLUT0 G0 Signal Replaced with G signal when in the CLUT0 format and CLUT0 = 0.
15 to 8	GR0_B0 [7:0]	0	R/W	CLUT0 B0 Signal Replaced with B signal when in the CLUT0 format and CLUT0 = 0.
7 to 0	GR0_R0 [7:0]	0	R/W	CLUT0 R0 Signal Replaced with R signal when in the CLUT0 format and CLUT0 = 0.

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

38.4.3.55 Alpha Blending Control Register 11 (Graphics 0) (GR0_AB11)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR0_A1[7:0]								GR0_G1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_B1[7:0]								GR0_R1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR0_A1 [7:0]	0	R/W	CLUT1 α1 Signal Replaced with α signal when in the CLUT1 format and CLUT1 = 1. Replaced with α signal when in the αRGB1555/RGBα5551 format and α = 1. Note: These bits should always be set to 255 to display the current graphics only.
23 to 16	GR0_G1 [7:0]	0	R/W	CLUT1 G1 Signal Replaced with G signal when in the CLUT1 format and CLUT1 = 1.
15 to 8	GR0_B1 [7:0]	0	R/W	CLUT1 B1 Signal Replaced with B signal when in the CLUT1 format and CLUT1 = 1.
7 to 0	GR0_R1 [7:0]	0	R/W	CLUT1 R1 Signal Replaced with R signal when in the CLUT1 format and CLUT1 = 1.

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

38.4.3.56 Background Color Control Register (Graphics 0) (GR0_BASE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR0_BASE_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR0_BASE_B[7:0]								GR0_BASE_R[7:0]							
Initial value:	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR0_BASE_G[7:0]	0	R/W	Background Color G/Y Signal G/Y: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR0_BASE_B[7:0]	128	R/W	Background Color B/Cb Signal B: Unsigned 8 bits (0 to 255 [LSB]) Cb: 8 bits; 128 offset binary; unsigned (0 to 255 [LSB])
7 to 0	GR0_BASE_R[7:0]	128	R/W	Background Color R/Cr Signal R: Unsigned 8 bits (0 to 255 [LSB]) Cr: 8 bits; 128 offset binary; unsigned (0 to 255 [LSB])

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

38.4.3.57 CLUT Table Control Register (Graphics 0) (GR0_CLUT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR0_CLT_SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR0_CLT_SEL	0	R/W	CLUT Table Select Signal 0: Selects CLUT table 0. Referring to the CLUT table 0 value to expand to αRGB8888 The CPU side can read-access or write-access to the CLUT table 1. 1: Selects CLUT table 1. Referring to the CLUT table 1 value to expand to αRGB8888 The CPU side can read-access or write-access to the CLUT table 0.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: This register is updated when the GR0_P_VEN bit in the graphics 0 register update control register (GR0_UPDATE) is 1.

NOTE

Before switching CLUT table by GR_CLT_SEL, CPU must wait until the read value from the last written address is same as the last written value.

Refer to Section 38.6.1.15, CLUT Table for detail.

38.4.3.58 SCL0 Register Update Control Register (SC1_SCL0_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SC1_SCL0_VEN_D	SC1_SCL0_VEN_C	—	—	—	SC1_SCL0_UPDATE	—	—	—	SC1_SCL0_VEN_B	—	—	—	SC1_SCL0_VEN_A
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/WC1	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	SC1_SCL0_VEN_D	0	R/WC1	Scaling-Up Control and Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
12	SC1_SCL0_VEN_C	0	R/WC1	Scaling-Down Control and Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SC1_SCL0_UPDATE	0	R/WC1	SYNC Control Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC1_SCL0_VEN_B	0	R/WC1	Synchronization Control and Scaling-up Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC1_SCL0_VEN_A	0	R/WC1	Scaling-Down Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.4.3.59 Mask Control Register (SC1_SCL0_FRC1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC1_RES_VMASK[15:0]															
Initial value:	0	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SC1_RES_VMASK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SC1_RES_VMASK [15:0]	2800	R/W	Repeated Vsync Signal Masking Period Sets the repeated Vsync signal masking period beginning at a Vsync signal in terms of 128 pixel-clock periods. Masking period [usec] = SC1_RES_VMASK[15:0] × 128 ÷ pixel clock frequency [MHz]
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC1_RES_ VMASK_ON	1	R/W	Repeated Vsync Signal Masking Control 0: Repeated Vsync signal masking control is disabled. 1: Repeated Vsync signal masking control is enabled.

Note: This register is updated when the SC1_SCL0_UPDATE bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.60 Missing Vsync Compensation Control Register (SC1_SCL0_FRC2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SC1_RES_VLACK[15:0]															
Initial value:	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SC1_RES_VLACK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	SC1_RES_ VLACK [15:0]	3600	R/W	Missing-Sync Compensating Pulse Output Wait Time Sets the wait time before outputting a missing-sync compensating pulse after a Vsync signal. Wait time [usec] = SC1_RES_VLACK[15:0] × 128 ÷ pixel clock frequency [MHz]
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC1_RES_ VLACK_ON	1	R/W	Missing Vsync Signal Compensation 0: Compensation of missing Vsync signals is disabled. 1: Compensation of missing Vsync signals is enabled.

Note: This register is updated when the SC1_SCL0_UPDATE bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.61 Output Sync Select Register (SC1_SCL0_FRC3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SC1_RES_VS_IN_SEL	—	—	—	—	—	—	—	SC1_RES_VS_SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SC1_RES_VS_IN_SEL	0	R/W	Horizontal and Vertical Sync Signal Output and Full-Screen Enable Signal Select This setting is ignored when cascaded connection is enabled (GR1_AB1.GR1_CUS_CON_ON = 1). 0: Horizontal and vertical sync signal output and full-screen enable signal from scaler 1 1: Horizontal and vertical sync signal output and full-screen enable signal from scaler 0
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC1_RES_VS_SEL	1	R/W	Vsync Signal Output Select 0: Externally input Vsync signal 1: Internally generated free-running Vsync signal

Note: This register is updated when the SC1_SCL0_UPDATE bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.62 Free-Running Period Control Register (SC1_SCL0_FRC4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC1_RES_FV[10:0]										
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC1_RES_FH[10:0]										
Initial value:	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC1_RES_FV[10:0]	524	R/W	Free-Running Vsync Period Setting Free-running Vsync period = (SC1_RES_FV[10:0] + 1) × horizontal period [usec]
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC1_RES_FH[10:0]	799	R/W	Hsync Period Setting Hsync period [usec] = (SC1_RES_FH[10:0] +1) ÷ pixel clock frequency [MHz]

Note: This register is updated when the SC1_SCL0_UPDATE bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.63 Output Delay Control Register (SC1_SCL0_FRC5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SC1_RES_FLD_DLY_SEL	SC1_RES_VSDLY[7:0]							
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	SC1_RES_FLD_DLY_SEL	1	R/W	Field Determination Signal Delay Control 0: No delay 1: Delay of one vertical cycle
7 to 0	SC1_RES_VSDLY[7:0]	1	R/W	Vsync Signal Delay Control Adjusts the Vsync signal delay in the output Hsync period units. Vsync signal delay [usec]: SC1_RES_VSDLY[7:0] × output Hsync period [usec]

Note: This register is updated when the SC1_SCL0_VEN_B bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.64 Full-Screen Vertical Size Register (SC1_SCL0_FRC6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC1_RES_F_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC1_RES_F_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC1_RES_F_VS[10:0]	35	R/W	Vertical Enable Signal Start Position for Full Screen. (VSYNC + V backporch lines) Note: The set value should be four or more (lines). SC1_RES_F_VS[10:0] + SC1_RES_F_VW[10:0] should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC1_RES_F_VW[10:0]	480	R/W	Vertical Enable Signal Width for Full Screen (lines) Note: SC1_RES_F_VS[10:0] + SC1_RES_F_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when the SC1_SCL0_VEN_B bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.65 Full-Screen Horizontal Size Register (SC1_SCL0_FRC7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC1_RES_F_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC1_RES_F_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC1_RES_F_HS[10:0]	144	R/W	Horizontal Enable Signal Start Position for Full Screen. (HSYNC+H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). SC1_RES_F_HS[10:0] + SC1_RES_F_HW[10:0] should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC1_RES_F_HW[10:0]	640	R/W	Horizontal Enable Signal Width for Full Screen (pixel-clock cycles) **Notes:** 1. SC1_RES_F_HS[10:0] + SC1_RES_F_HW[10:0] should be equal to or less than 2015 (clock cycles). 2. In the case of video output format is Serial RGB(OUT_SET.OUT_FORMAT[[1:0] = 3), it is nessesary to add 2 to value in SC1_RES_F_HW.

Note: This register is updated when the SC1_SCL0_VEN_B bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.66 Vsync Detection Register (SC1_SCL0_FRC9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC1_RES_QVLOCK	—	—	—	SC1_RES_QVLACK
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC1_RES_QVLOCK	0	R	Locked Vsync Signal Detection Flag 1: Repeated and missing Vsync signal input has not been detected for four or more vertical periods. 0: Repeated or missing Vsync signal input has been detected.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC1_RES_QVLACK	0*1	R	Missing Vsync Signal Detection Flag 1: Missing Vsync signal input has been detected. 0: Missing Vsync signal input has not been detected.

Note 1. SC1_RES_QVLACK changes immediately from 0 to 1, if the video output clock is enabled. It changes back to 0 after detection of the first VSync. So interrupt on missing Vsync shall be enabled only from the second Vsync onwards.

38.4.3.67 Scaling-Down Control Register (SC1_SCL0_DS1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC1_RES_DS_V_ON	—	—	—	SC1_RES_DS_H_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC1_RES_DS_V_ON	1	R/W	Vertical Scale Down On/Off 0: Off 1: Setting prohibited
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC1_RES_DS_H_ON	1	R/W	Horizontal Scale Down On/Off 0: Off 1: Setting prohibited

Note: This register is updated when the SC1_SCL0_VEN_A bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

NOTE

The down-scaler of scaler block 1 is not available.Screen Synthesis

The initial value “1” of the SC1_RES_DS_V_ON and SC1_RES_DS_H_ON bits must be changed to “0” when up-scaler is used at scaler block 1.

38.4.3.68 Vertical Scaling Register (SC1_SCL0_DS6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC1_RES_V_RATIO[15:0]															
Initial value:	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	SC1_RES_V_RATIO [15:0]	2044	R/W	Vertical Scale Up/Down Ratio ([15:12]: Integer part, [11:0]: Decimal part) For scale down: round (SC1_RES_VW ÷ SC1_RES_OUT_VW × 4096) For scale up: round (SC1_RES_IN_VW[10:0] ÷ SC1_RES_P_VW[10:0] × 4096) SC1_RES_V_RATIO[15:0] < 4096: Scale up SC1_RES_V_RATIO[15:0] = 4096: 100% scale up SC1_RES_V_RATIO[15:0] > 4096: Scale down

Note: These bits updated when the SC1_SCL0_VEN_A and SC1_SCL0_VEN_B bits in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) are 1. Accordingly, even a scaled-up graphics display requires both an input Vsync and output Vsync signal.

38.4.3.69 Scaling-Up Control Register (SC1_SCL0_US1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC1_RES_US_V_ON	—	—	—	SC1_RES_US_H_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC1_RES_US_V_ON	1	R/W	Vertical Scale Up On/Off 0: Off 1: On **Note:** For D1L2(H) devices this bit must be set to 0.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC1_RES_US_H_ON	1	R/W	Horizontal Scale Up On/Off 0: Off 1: On **Note:** For D1L2(H) devices this bit must be set to 0.

Note: This register is updated when the SC1_SCL0_VEN_B bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.70 Output Image Vertical Size Register (SC1_SCL0_US2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC1_RES_P_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC1_RES_P_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC1_RES_P_VS[10:0]	35	R/W	Vertical Enable Signal Start Position for Output Image (VSYNC + V backporch lines) Note: The set value should be four or more (lines). SC1_RES_P_VS[10:0] + SC1_RES_P_VW[10:0] should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC1_RES_P_VW[10:0]	480	R/W	Vertical Enable Signal Width for Output Image (lines) Note: SC1_RES_P_VS[10:0] + SC1_RES_P_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when the SC1_SCL0_VEN_B bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.71 Output Image Horizontal Size Register (SC1_SCL0_US3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC1_RES_P_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC1_RES_P_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC1_RES_P_HS[10:0]	144	R/W	Horizontal Enable Signal Start Position for Output Image (HSYNC+H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). SC1_RES_P_HS[10:0] + SC1_RES_P_HW[10:0] should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC1_RES_P_HW[10:0]	640	R/W	Horizontal Enable Signal Width for Output Image (pixel-clock cycles) Note: SC1_RES_P_HS[10:0] + SC1_RES_P_HW[10:0] should be equal to or less than 2015 (clock cycles).

Note: This register is updated when the SC1_SCL0_VEN_B bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.72 Scaling-Up Control Block Input Size Register (SC1_SCL0_US4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	SC1_RES_IN_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SC1_RES_IN_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	SC1_RES_IN_VW[10:0]	240	R/W	Number of Valid Lines in Vertical Direction Input to Scaling-down Control Block (lines)
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	SC1_RES_IN_HW[10:0]	640	R/W	Number of Valid Horizontal Pixels Input to Scaling-down Control Block (pixel-clock cycles)

Note: This register is updated when the SC1_SCL0_VEN_B and SC1_SCL0_VEN_D bits in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) are 1.

38.4.3.73 Horizontal Scale Up Register (SC1_SCL0_US5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC1_RES_US_H_RATIO[15:0]															
Initial value:	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 0	SC1_RES_US_H_RATIO [15:0]	9224	R/W	Horizontal Scale Up Ratio ([15:12]: Integer part, [11:0]: Decimal part) round (SC1_RES_IN_HW[10:0] ÷ SC1_RES_P_HW[10:0] × 4096) SC1_RES_US_H_RATIO[15:0] < 4096: Scale up SC1_RES_US_H_RATIO[15:0] = 4096: 100% scale up SC1_RES_US_H_RATIO[15:0] > 4096: Setting prohibited

Note: This register is updated when the SC1_SCL0_VEN_B bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.74 Horizontal Scale Up Initial Phase Register (SC1_SCL0_US6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	SC1_RES_US_H_INTERPOTYP	SC1_RES_US_HT_INIPHASE[11:0]											
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	SC1_RES_US_HB_INIPHASE[11:0]											
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	SC1_RES_US_H_INTERPOTYP	1	R/W	Horizontal Interpolation Mode Select 0: Hold interpolation 1: Linear interpolation
27 to 16	SC1_RES_US_HT_INIPHASE[11:0]	0	R/W	Horizontal Interpolation Start Phase for Top Field 0 to 4095 (0 to approx. 1.0)
15 to 12	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
11 to 0	SC1_RES_US_HB_INIPHASE[11:0]	0	R/W	Horizontal Interpolation Start Phase for Bottom Field 0 to 4095 (0 to approx. 1.0)

Note: This register is updated when the SC1_SCL0_VEN_B bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.75 Trimming Register (SC1_SCL0_US7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC1_RES_HCUT[7:0]								SC1_RES_VCUT[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 8	SC1_RES_HCUT[7:0]	0	R/W	Horizontal Amount of Cut-off Post-Scaling Image (Right and Left Parts) Sets the number of pixel-clock cycles.
7 to 0	SC1_RES_VCUT[7:0]	0	R/W	Vertical Amount of Cut-off Post-Scaling Image (Upper and Lower Parts) Sets the number of lines.

Note: This register is updated when the SC1_SCL0_VEN_B bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.76 Frame Buffer Read Select Register (SC1_SCL0_US8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	SC1_RES_IBUS_SYNC_SEL	—	—	—	SC1_RES_DISP_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	SC1_RES_IBUS_SYNC_SEL	0	R/W	Sync Signal Select for Frame Buffer Read Block 0: Sync signals from the scaling-up control block 1: Sync signals from the graphics processing block
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	SC1_RES_DISP_ON	0	R/W	Post-Scaling Image Frame Display On/Off 0: Frame display off 1: Frame display on

Note: SC1_RES_IBUS_SYNC_SEL is updated when the SC1_SCL0_VEN_B and SC1_SCL0_VEN_D bits in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1. SC1_RES_DISP_ON is updated when the SC1_SCL0_VEN_B bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.77 Background Color Register (SC1_SCL0_OVR1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	SC1_RES_BK_COL_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SC1_RES_BK_COL_G[7:0]								SC1_RES_BK_COL_B[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	SC1_RES_BK_COL_R[7:0]	128	R/W	Background Color Setting R/Cr Signal R: 8 bits; unsigned (0 to 255 [LSB]) Cr:8 bits; 128 offset binary; unsigned (0 to 255 [LSB])
15 to 8	SC1_RES_BK_COL_G[7:0]	0	R/W	Background Color Setting G/Y Signal G/Y: 8 bits; unsigned (0 to 255 [LSB])
7 to 0	SC1_RES_BK_COL_B[7:0]	128	R/W	Background Color Setting B/Cb Signal B: 8 bits; unsigned (0 to 255 [LSB]) Cb:8 bits; 128 offset binary; unsigned (0 to 255 [LSB])

Note: This register is updated when the SC1_SCL0_VEN_B bit in the SC1_SCL0 register update control register (SC1_SCL0_UPDATE) is 1.

38.4.3.78 Graphics 1 Register Update Control Register (GR1_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GR1_UPDATE	—	—	—	GR1_P_VEN	—	—	—	GR1_IBUS_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR1_UPDATE	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR1_P_VEN	0	R/WC1	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR1_IBUS_VEN	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.4.3.79 Frame Buffer Read Control Register (Graphics 1) (GR1_FLM_RD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR1_R_ENB
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR1_R_ENB	0	R/W	Frame Buffer Read Enable 0: Frame buffer reading is disabled. 1: Frame buffer reading is enabled.

Note: This register is updated when the GR1_IBUS_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.80 Frame Buffer Control Register 1 (Graphics 1) (GR1_FLM1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR1_FLD_SEL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR1_LN_OFF_DIR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR1_FLM_SEL[1:0]		—	—	—	—	—	—	—	Caution *
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

CAUTION

The initial value “0” of bit 0 must be changed to “1”.

Bit	Bit Name	Initial Value	R/W	Description
31	GR1_FLD_SEL	0	R/W	Enables or disables top or bottom field selection for the next frame buffer to be read. 0: Field selection is disabled. 1: Top or bottom field can be selected.
30 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR1_LN_OFF_DIR	0	R/W	Selects the line offset address direction of the frame buffer. 0: Increments the address by the line offset address. 1: Decrements the address by the line offset address.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	GR1_FLM_SEL[1:0]	0	R/W	Selects a frame buffer address setting signal. 1: Selects GR1_FLM_NUM[9:0]. Others: Setting prohibited.
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	Bit0	0	R/W	**CAUTION** The initial value “0” of this bit must be changed to “1”.

Note: GR1_FLD_SEL is updated when the GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.
GR1_LN_OFF_DIR is updated when the GR1_IBUS_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.
GR1_FLM_SEL[1:0] is updated when the GR1_P_VEN and GR1_IBUS_VEN bits in the graphics 1 register update control register (GR1_UPDATE) are 1.

38.4.3.81 Frame Buffer Control Register 2 (Graphics 1) (GR1_FLM2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR1_BASE[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR1_BASE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	GR1_BASE [31:0]	0	R/W	Frame Buffer Base Address Sets the start address of the frame buffer where frame data is to be stored. The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when the GR1_IBUS_VEN and GR1_P_VEN bits in the graphics 1 register update control register (GR1_UPDATE) are 1.

38.4.3.82 Frame Buffer Control Register 3 (Graphics 1) (GR1_FLM3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR1_FLD_NXT	GR1_LN_OFF[14:0]														
Initial value:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR1_FLM_NUM[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	GR1_FLD_NXT	0	R/W	Top or Bottom Field Selection for Next Frame Buffer 0: Bottom 1: Top
30 to 16	GR1_LN_ OFF[14:0]	2048	R/W	Frame Buffer Line Offset Address Sets the line offset address for calculating the start address of each line. Line 0: GR1_BASE[31:0] Line 1: GR1_BASE[31:0] + GR1_LN_OFF[14:0] × 1 : Line n: GR1_BASE[31:0] + GR1_LN_OFF[14:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GR1_FLM_ NUM[9:0]	1	R/W	Frame Number of Frame Buffer Manually set the frame number when GR1_FLM_SEL[1:0] = 1.

Note: GR1_FLD_NXT is updated when the GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.
GR1_LN_OFF[14:0] and GR1_FLM_NUM[9:0] are updated when the GR1_IBUS_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.83 Frame Buffer Control Register 4 (Graphics 1) (GR1_FLM4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	GR1_FLM_OFF[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR1_FLM_OFF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	GR1_FLM_OFF[22:0]	524288	R/W	Frame Buffer Frame Offset Address Specifies the frame offset address used for calculating the start address of each frame buffer when more than one buffer is used. Buffer 0: GR1_BASE[31:0] Buffer 1: GR1_BASE[31:0] + GR1_FLM_OFF[22:0] × 1 : Buffer n: GR1_BASE[31:0] + GR1_FLM_OFF[22:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when the GR1_IBUS_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.84 Frame Buffer Control Register 5 (Graphics 1) (GR1_FLM5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR1_FLM_LNUM[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR1_FLM_LOOP[10:0]										
Initial value:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR1_FLM_LNUM[10:0]	0	R/W	Sets number of lines in a frame Number of lines is (GR1_FLM_LNUM[10:0] + 1).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR1_FLM_LOOP[10:0]	1023	R/W	Number of lines when reading the addresses repeatedly by returning to the start address after reaching the end address. (GR1_FLM_LOOP[10:0] + 1) lines are read.

Note: This register is updated when the GR1_IBUS_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.85 Frame Buffer Control Register 6 (Graphics 1) (GR1_FLM6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR1_FORMAT[3:0]				—	GR1_HW[10:0]										
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR1_YCC_SWAP[2:0]			GR1_RDSWA[2:0]			—	GR1_CNV444_MD	—	—	GR1_STA_POS[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	GR1_FORMAT [3:0]	8	R/W	Sets the format of the frame buffer read signal. 0: RGB565 1: RGB888 2: αRGB1555 3: αRGB4444 4: αRGB8888 5: CLUT8 6: CLUT4 7: CLUT1 8: YCbCr422 9: YCbCr444 10: RGBα5551 11: RGBα8888 12 to 15: Setting prohibited
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 16	GR1_HW [10:0]	0	R/W	Sets the width of the horizontal valid period. The width is (GR1_HW[10:0] + 1) pixels. Note: The set value should be equal to or more than two.
15 to 13	GR1_YCC_SWAP[2:0]	0	R/W	Controls swapping of data read from buffer in the YCbCr422 format. 0: Cb/Y0/Cr/Y1 1: Y0/Cb/Y1/Cr 2: Cr/Y0/Cb/Y1 3: Y0/Cr/Y1/Cb 4: Y1/Cr/Y0/Cb 5: Cr/Y1/Cb/Y0 6: Y1/Cb/Y0/Cr 7: Cb/Y1/Cr/Y0

Bit	Bit Name	Initial Value	R/W	Description
12 to 10	GR1_ RDSWA [2:0]	0	R/W	8-Bit, 16-Bit, or 32-Bit Swap Setting These bits control swapping in frame buffer reading as follows. Bit 0 0: Swapped in 8-bit units. 1: Not swapped in 8-bit units. Bit 1 0: Swapped in 16-bit units. 1: Not swapped in 16-bit units. Bit 2 0: Swapped in 32-bit units. 1: Not swapped in 32-bit units. According to the setting of these bits, data is swapped as follows. Each number in parentheses ((1) to (8)) indicates 8-bit data. 000: (1) (2) (3) (4) (5) (6) (7) (8) [Not swapped] 001: (2) (1) (4) (3) (6) (5) (8) (7) [Swapped in 8-bit units] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Swapped in 16-bit units] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Swapped in 16-bit units + 8-bit units] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Swapped in 32-bit units] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Swapped in 32-bit units + 8-bit units] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Swapped in 32-bit units + 16-bit units] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Swapped in 32-bit units + 16-bit units + 8-bit units]
9	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
8	GR1_ CNV444_ MD	0	R/W	Sets the interpolation mode for YCbCr422 to YCbCr444 conversion. 0: Hold interpolation 1: Average interpolation
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	GR1_STA_ POS[5:0]	0	R/W	Sets the amount of data to be skipped through. Specifically data amount equal to the amount indicated by GR1_STA_POS[5:0] is skipped from the start of the line.

Note: GR1_YCC_SWAP[2:0], GR1_CNV444, and GR1_STA_POS[5:0] are updated when GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.
GR1_ RDSWA is updated when the GR1_UPDATE bit in the graphics 1 register update control register (GR1_UPDATE) is 1.
GR1_FORMAT[3:0] and GR1_HW[10:0] are updated when GR1_IBUS_VEN and GR1_P_VEN bits in the graphics 1 register update control register (GR1_UPDATE) are 1.

38.4.3.86 Alpha Blending Control Register 1 (Graphics 1) (GR1_AB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	GR1_CUS_CON_ON	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR1_ARC_MUL	GR1_ACALC_MD	—	GR1_ARC_ON	—	—	—	GR1_ARC_DISP_ON	—	—	—	GR1_GRC_DISP_ON	—	—	GR1_DISP_SEL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	GR1_CUS_CON_ON	0	R/W	Enables or disables cascaded connection. 0: Cascaded connection is disabled. 1: Cascaded connection is enabled.
27 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	GR1_ARC_MUL	0	R/W	Enables or disables multiplication with the current α for alpha blending in rectangular area. 0: Disabled 1: Enabled
14	GR1_ACALC_MD	0	R/W	Enables or disables pre-multiplication for alpha blending in pixel units. 0: Disabled 1: Enabled
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	GR1_ARC_ON	0	R/W	Enables or disables alpha blending in rectangular area. 0: Disabled 1: Enabled
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR1_ARC_DISP_ON	0	R/W	Turns on/off frame-line display of the image area for alpha blending in rectangular area. 0: Frame-line display is turned off. 1: Frame-line display is turned on.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR1_GRC_DISP_ON	0	R/W	Turns on/off frame-line display of the graphics image area. 0: Frame-line display off 1: Frame-line display on
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	GR1_DISP_SEL[1:0]	0	R/W	Selects the graphics display mode. 0: Background color display (GR1_BASE[31:0]) 1: Lower-layer graphics display When displaying video image or enlarged graphics, select this setting. 2: Current graphics display When displaying graphics, select this setting. 3: Blended display of lower-layer graphics and current graphics* Note:* When cascaded connection is disabled (GR1_CUS_CON_ON = 0), select this setting whenever chroma-key processing is to proceed. Since only current graphics are to be displayed by chroma-key processing, set the α values for both pixels to be subject to chroma-keying and pixels not to be subject to chroma-keying to 255.

Note: GR1_CUS_CON_ON is updated when GR1_UPDATE bit in the graphics 1 register update control register (GR1_UPDATE) is 1. The other bits are updated when GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.87 Alpha Blending Control Register 2 (Graphics 1) (GR1_AB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR1_GRC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR1_GRC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR1_GRC_VS[10:0]	0	R/W	Vertical Start Position of Graphics Image Area. Note: The set value should be four or more (lines). GR1_GRC_VS[10:0] + GR1_GRC_VW[10:0] should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR1_GRC_VW[10:0]	0	R/W	Vertical Width of Graphics Image Area. **NOTE** GR1_GRC_VS[10:0] + GR1_GRC_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when the GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.88 Alpha Blending Control Register 3 (Graphics 1) (GR1_AB3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR1_GRC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR1_GRC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR1_GRC_HS[10:0]	0	R/W	Horizontal Start Position of Graphics Image Area. Note: The set value should be 16 or more (clock cycles). GR1_GRC_HS[10:0] + GR1_GRC_HW[10:0] should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR1_GRC_HW[10:0]	0	R/W	Horizontal Width of Graphics Image Area. Note: For displaying an image with 1- or 2-pixel horizontal width, set GR1_HW[10:0] to 2 and GR1_GRC_HW[10:0] to 1 (1 pixel) or 2 (2 pixels). **NOTE** GR1_GRC_HS[10:0] + GR1_GRC_HW[10:0] should be equal to or less than 2015 (clock cycles).

Note: This register is updated when the GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.89 Alpha Blending Control Register 4 (Graphics 1) (GR1_AB4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR1_ARC_VS[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR1_ARC_VW[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR1_ARC_VS[10:0]	0	R/W	Sets the vertical start position of the valid image area for alpha blending in a rectangular area.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR1_ARC_VW[10:0]	0	R/W	Sets the vertical width of the valid image area for alpha blending in a rectangular area.

Note: This register is updated when GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.90 Alpha Blending Control Register 5 (Graphics 1) (GR1_AB5)

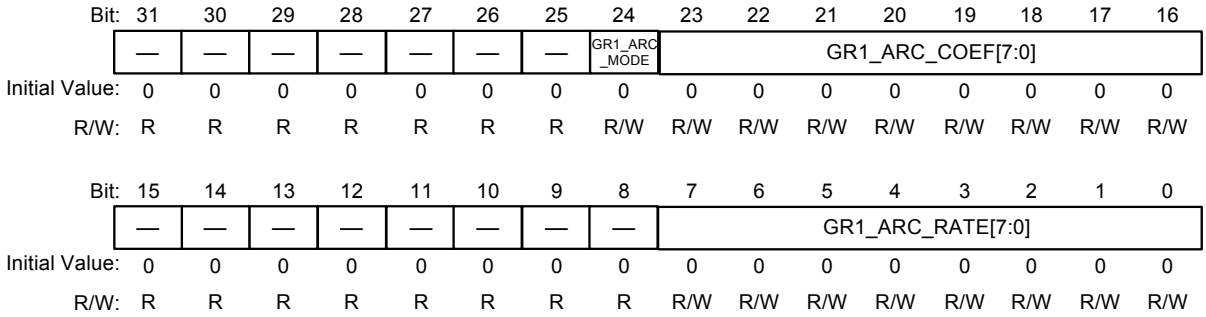
Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR1_ARC_HS[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR1_ARC_HW[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR1_ARC_HS[10:0]	0	R/W	Sets the horizontal start position of the valid image area for alpha blending in a rectangular area.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR1_ARC_HW[10:0]	0	R/W	Sets the horizontal width of the valid image area for alpha blending in a rectangular area.

Note: This register is updated when GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

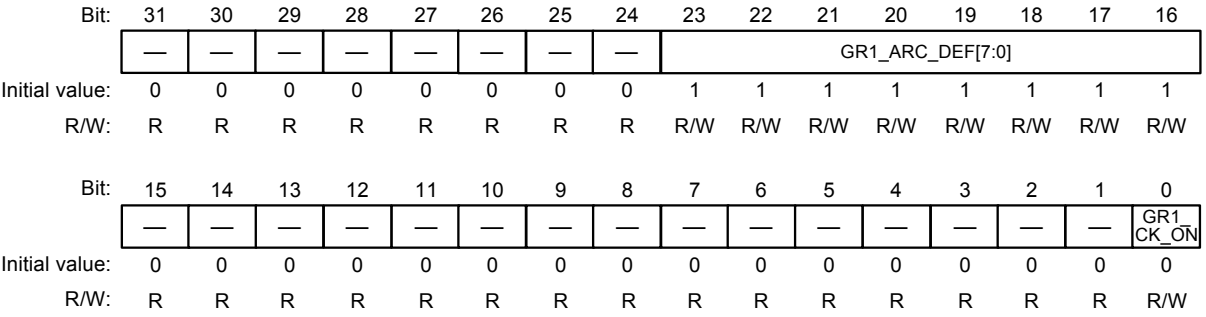
38.4.3.91 Alpha Blending Control Register 6 (Graphics 1) (GR1_AB6)



Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	GR1_ARC_MODE	0	R/W	Alpha Blending Mode in Rectangular Area 0: Addition 1: Subtraction
23 to 16	GR1_ARC_COEF[7:0]	0	R/W	Alpha Coefficient for Alpha Blending in Rectangular Area (0 to 255) [7:0]: Variation (absolute value)
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	GR1_ARC_RATE[7:0]	0	R/W	Frame Rate for Alpha Blending in Rectangular Area

Note: This register is updated when GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.92 Alpha Blending Control Register 7 (Graphics 1) (GR1_AB7)



Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR1_ARC_DEF[7:0]	255	R/W	Initial Alpha Value for Alpha Blending in Rectangular Area
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR1_CK_ON	0	R/W	CLUT-Index/RGB-Index Chroma-Key Processing On/Off 0: Off 1: On

Note: This register is updated when the GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.93 Alpha Blending Control Register 8 (Graphics 1) (GR1_AB8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR1_CK_KCLUT[7:0]								GR1_CK_KG[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR1_CK_KB[7:0]								GR1_CK_KR[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR1_CK_KCLUT[7:0]	0	R/W	CLUT Signal for CLUT-Index Chroma-Key Processing CLUT: Unsigned 8 bits (0 to 255 [LSB])
23 to 16	GR1_CK_KG[7:0]	0	R/W	G Signal for RGB-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR1_CK_KB[7:0]	0	R/W	B Signal for RGB-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR1_CK_KR[7:0]	0	R/W	R Signal for RGB-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when the GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.94 Alpha Blending Control Register 9 (Graphics 1) (GR1_AB9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR1_CK_A[7:0]								GR1_CK_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR1_CK_B[7:0]								GR1_CK_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR1_CK_A[7:0]	0	R/W	Replaced Alpha Signal after RGB-Index Chroma-Key Processing α: Unsigned 8 bits (0 to 255 [LSB]) Note: These bits should always be set to 255 to display the current graphics only when cascaded connection is disabled.
23 to 16	GR1_CK_G[7:0]	0	R/W	Replaced G Signal after RGB-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR1_CK_B[7:0]	0	R/W	Replaced B Signal after RGB-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR1_CK_R[7:0]	0	R/W	Replaced R Signal after RGB-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when the GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.95 Alpha Blending Control Register 10 (Graphics 1) (GR1_AB10)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR1_A0[7:0]								GR1_G0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR1_B0[7:0]								GR1_R0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR1_A0 [7:0]	0	R/W	CLUT0 α0 Signal Replaced with α signal when in the CLUT0 format and CLUT0 = 0. Replaced with α signal when in the αRGB1555/RGBα5551 format and α = 0. Note: These bits should always be set to 255 to display the current graphics only when cascaded connection is disabled.
23 to 16	GR1_G0 [7:0]	0	R/W	CLUT0 G0 Signal Replaced with G signal when in the CLUT0 format and CLUT0 = 0.
15 to 8	GR1_B0 [7:0]	0	R/W	CLUT0 B0 Signal Replaced with B signal when in the CLUT0 format and CLUT0 = 0.
7 to 0	GR1_R0 [7:0]	0	R/W	CLUT0 R0 Signal Replaced with R signal when in the CLUT0 format and CLUT0 = 0.

Note: This register is updated when the GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.96 Alpha Blending Control Register 11 (Graphics 1) (GR1_AB11)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR1_A1[7:0]								GR1_G1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR1_B1[7:0]								GR1_R1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR1_A1 [7:0]	0	R/W	CLUT1 α1 Signal Replaced with α signal when in the CLUT1 format and CLUT1 = 1. Replaced with α signal when in the αRGB1555/RGBα5551 format and α = 1. Note: These bits should always be set to 255 to display the current graphics only when cascaded connection is disabled.
23 to 16	GR1_G1 [7:0]	0	R/W	CLUT1 G1 Signal Replaced with G signal when in the CLUT1 format and CLUT1 = 1.
15 to 8	GR1_B1 [7:0]	0	R/W	CLUT1 B1 Signal Replaced with B signal when in the CLUT1 format and CLUT1 = 1.
7 to 0	GR1_R1 [7:0]	0	R/W	CLUT1 R1 Signal Replaced with R signal when in the CLUT1 format and CLUT1 = 1.

Note: This register is updated when the GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.97 Background Color Control Register (Graphics 1) (GR1_BASE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR1_BASE_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR1_BASE_B[7:0]								GR1_BASE_R[7:0]							
Initial value:	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR1_BASE_G[7:0]	0	R/W	Background Color G/Y Signal G/Y: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR1_BASE_B[7:0]	128	R/W	Background Color B/Cb Signal B: Unsigned 8 bits (0 to 255 [LSB]) Cb: 8 bits; 128 offset binary; unsigned (0 to 255 [LSB])
7 to 0	GR1_BASE_R[7:0]	128	R/W	Background Color R/Cr Signal R: Unsigned 8 bits (0 to 255 [LSB]) Cr: 8 bits; 128 offset binary; unsigned (0 to 255 [LSB])

Note: This register is updated when the GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

38.4.3.98 CLUT Table Control Register (Graphics 1) (GR1_CLUT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR1_CLT_SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR1_CLT_SEL	0	R/W	CLUT Table Select Signal 0: Selects CLUT table 0. Referring to the CLUT table 0 value to expand to αRGB8888 The CPU side can read-access or write-access to the CLUT table 1. 1: Selects CLUT table 1. Referring to the CLUT table 1 value to expand to αRGB8888 The CPU side can read-access or write-access to the CLUT table 0.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: This register is updated when the GR1_P_VEN bit in the graphics 1 register update control register (GR1_UPDATE) is 1.

NOTE

Before switching CLUT table by GR_CLT_SEL, CPU must wait until the read value from the last written address is same as the last written value.

Refer to Section 38.6.1.15, CLUT Table for detail.

38.4.3.99 Status Monitor Register (Graphics 1) (GR1_MON)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR1_ ARC_ST
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR1_ARC_ST	0	R	Status Flag for Alpha Blending in Rectangular Area 0: Addition or subtraction has been completed. (α value is 0 or 255) 1: Addition or subtraction is in progress.

38.4.4 Usage Method

38.4.4.1 Scaling Setting Example for Graphics Display

(1) Angle of View for Graphics Display

This section describes an example of setting the signals of the input and output angles of view shown in **Table 38.88**.

Table 38.88 Input and Output Angles of View for Graphics Display

Graphics Size	Output Signal	Graphics Signal Format
640 × 480	640 × 480	RGB888

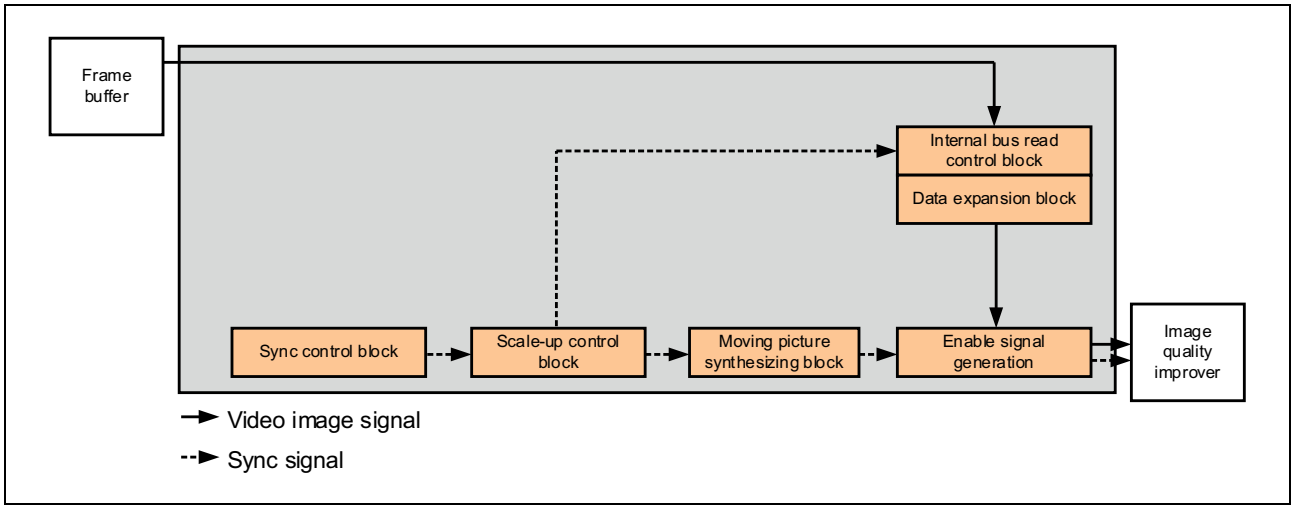


Figure 38.40 Signal Paths for Graphics Display

(2) Setting Frame Buffer Access Area

In the frame buffer in which graphics data is stored, graphics data needs to be expanded in the area of 640 × 480 pixels or larger.

Here, the frame buffer area in which graphics data is expanded is assumed to be 640 × 480 pixels.

Since the frame buffer is accessed in 64-bit units, RGB888 (32 bits) is accessed in 2-pixel units.

The line offset address values to be set are:

GR_LN_OFF[14:0] = 640 × 4 = 2560

The frame offset address values to be set are:

GR_FLM_OFF[22:0] = GR_LN_OFF[14:0] × 480 = 1228800

(3) Register Setting Example

Table 38.89 Register Setting Example for Graphics Display

Register Name	Bit Name	Settings	Remarks
Synchronization Control			
SC_SCL0_FRC3	SC_RES_VS_SEL	1	Free-running Vsync selected (when the appropriate input signal is available, an external sync can also be selected)
SC_SCL0_FRC4	SC_RES_FV[10:0]	524	Vertical period width of output signal (period width = set value + 1)
SC_SCL0_FRC4	SC_RES_FH[10:0]	799	Horizontal period width of output signal (period width = set value + 1)
Size of Angle of View			
SC_SCL0_FRC6	SC_RES_F_VS[10:0]	35	Vertical valid start position of full screen
SC_SCL0_FRC6	SC_RES_F_VW[10:0]	480	Vertical valid width of full screen
SC_SCL0_FRC7	SC_RES_F_HS[10:0]	144	Horizontal valid start position of full screen
SC_SCL0_FRC7	SC_RES_F_HW[10:0]	640	Horizontal valid width of full screen
GR_AB2	GR_GRC_VS[10:0]	35	Vertical valid start position of graphics output
GR_AB2	GR_GRC_VW[10:0]	480	Vertical valid width of graphics output
GR_AB3	GR_GRC_HS[10:0]	144	Horizontal valid start position of graphics output
GR_AB3	GR_GRC_HW[10:0]	640	Horizontal valid width of graphics output
Frame Buffer Read Setting			
GR_FLM1	GR_FLM_SEL[1:0]	1	Frame number setting with register
GR_FLM3	GR_FLM_NUM[9:0]	0	Frame number of frame buffer (0 in setting example)
GR_FLM5	GR_FLM_LNUM[10:0]	479	Number of graphics lines (number of lines = set value + 1)
GR_FLM6	GR_HW[10:0]	639	Horizontal valid width of graphics (valid width = set value + 1)
GR_FLM2	GR_BASE[31:0]	0	Conforming to graphics expansion setting (0 in setting example)
GR_FLM3	GR_LN_OFF[14:0]	2560	Conforming to graphics expansion setting
Frame Buffer Read Setting			
GR_FLM4	GR_FLM_OFF[22:0]	1228800	Conforming to graphics expansion setting
GR_FLM6	GR_FORMAT[3:0]	1	Frame buffer read format RGB888
GR_FLM_RD	GR_R_ENB	1	Frame buffer read enabled
Scaling-up Selection			
SC_SCL0_US8	SC_RES_IBUS_SYNC_SEL	1	Graphics output displayed
GR_AB1	GR_DISP_SEL[1:0]	2	Graphics display selected

38.4.4.2 Scaling Setting Example for Scaled-up Graphics Display

(1) Angles of View for Input and Output

This section describes an example of setting the signals of the input and output angles of view shown in **Table 38.90**.

Table 38.90 Input and Output Angles of View for Scaled-up Graphics Display

Graphics Size	Output Signal	Graphics Signal Format
640 × 480	800 × 600	RGB565

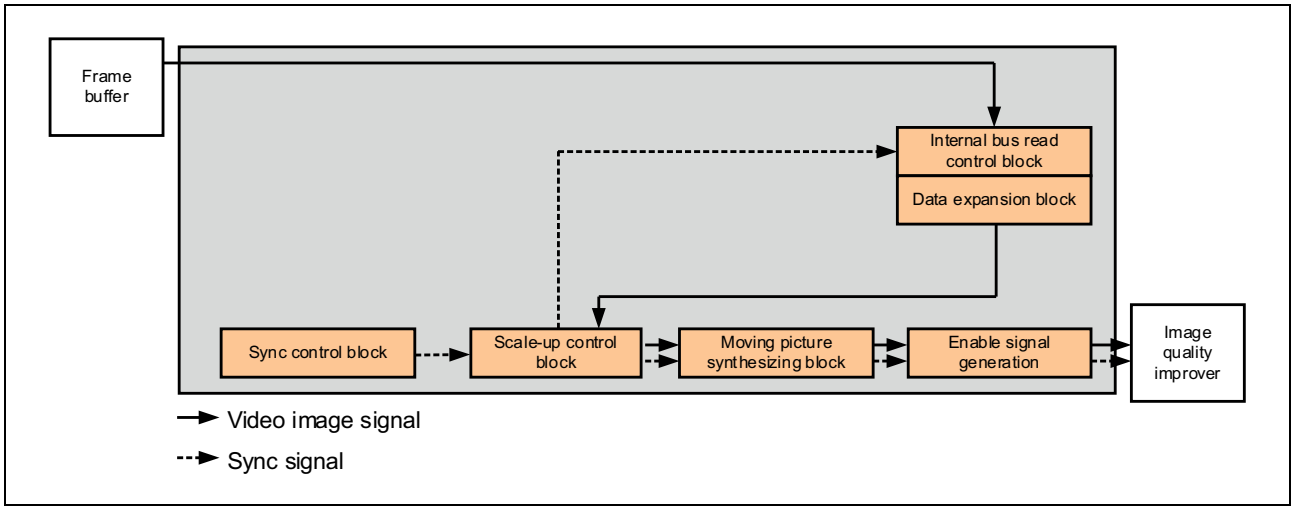


Figure 38.41 Signal Paths for Scaled-up Graphics Display

(2) Horizontal Scaling (Horizontal Scale Up, Scaling Filter: 2-Tap Linear)

The scaling rate for folding can be calculated as shown below.

RATIO_org = round (640 ÷ 800 × 4096) = 3277

σ = (3277 × (800 − 1) − (640 − 1) × 4096) ÷ (800 − 1) = 1.23

Horizontal scaling ratio = round (3277 − (1.23)) = 3276

(3) Vertical Scaling (Vertical Scale Up, Scaling Filter: 2-Tap Linear)

The scaling rate for folding can be calculated as shown below.

RATIO_org = round (480 ÷ 600 × 4096) = 3277

σ = (3277 × (600 − 1) − (480 − 1) × 4096) ÷ (600 − 1) = 1.57

Vertical scaling ratio = round (3277 − (1.57)) = 3275

(4) Setting Frame Buffer Access Area

In the frame buffer in which graphics data is stored, graphics data needs to be expanded in the area of 640 × 480 pixels or larger.

Here, the frame buffer area in which graphics data is expanded is assumed to be 640 × 480 pixels.

Since the frame buffer is accessed in 64-bit units, RGB565 (16 bits) is accessed in 4-pixel units.

The line offset address values to be set are:

GR_LN_OFF[14:0] = 640 × 2 = 1280

The frame offset address values to be set are:

GR_FLM_OFF[22:0] = GR_LN_OFF[14:0] × 480 = 614400

(5) Register Setting Example

Table 38.91 Register Setting Example for Scaled-up Graphics Display

Register Name	Bit Name	Settings	Remarks
Synchronization Control			
SC_SCL0_FRC3	SC_RES_VS_SEL	1	Free-running Vsync selected (when the appropriate input signal is available, an external sync can also be selected)
SC_SCL0_FRC4	SC_RES_FV[10:0]	668	Vertical period width of output signal (period width = set value + 1)
SC_SCL0_FRC4	SC_RES_FH[10:0]	1040	Horizontal period width of output signal (period width = set value + 1)
Size of Angle of View			
SC_SCL0_FRC6	SC_RES_F_VS[10:0]	27	Vertical valid start position of full screen
SC_SCL0_FRC6	SC_RES_F_VW[10:0]	600	Vertical valid width of full screen
SC_SCL0_FRC7	SC_RES_F_HS[10:0]	216	Horizontal valid start position of full screen
SC_SCL0_FRC7	SC_RES_F_HW[10:0]	800	Horizontal valid width of full screen
SC_SCL0_US2	SC_RES_P_VS[10:0]	27	Vertical valid start position of image output
SC_SCL0_US2	SC_RES_P_VW[10:0]	600	Vertical valid width of image output
SC_SCL0_US3	SC_RES_P_HS[10:0]	216	Horizontal valid start position of image output
SC_SCL0_US3	SC_RES_P_HW[10:0]	800	Horizontal valid width of image output
Scaling Setting			
SC_SCL0_US5	SC_RES_US_H_RATIO[15:0]	3276	Horizontal scaling-up because SC_RES_US_H_RATIO is smaller than 4096
SC_SCL0_DS6	SC_RES_V_RATIO[15:0]	3275	Vertical scaling-up because SC_RES_V_RATIO is smaller than 4096
SC_SCL0_US1	SC_RES_US_H_ON	1	Horizontal scaling-up on
SC_SCL0_US1	SC_RES_US_V_ON	1	Vertical scaling-up on
SC_SCL0_US4	SC_RES_IN_VW[10:0]	480	Vertical width of frame buffer read
SC_SCL0_US4	SC_RES_IN_HW[10:0]	640	Horizontal width of frame buffer read
Frame Buffer Read Setting			
GR_FLM1	GR_FLM_SEL[1:0]	1	Frame number setting with register
GR_FLM3	GR_FLM_NUM[9:0]	0	Frame number of frame buffer (0 in setting example)
GR_FLM2	GR_BASE[31:0]	0	Conforming to graphics expansion setting (0 in setting example)
GR_FLM3	GR_LN_OFF[14:0]	1280	Conforming to graphics expansion setting
GR_FLM4	GR_FLM_OFF[22:0]	614400	Conforming to graphics expansion setting
GR_FLM6	GR_FORMAT[3:0]	0	Frame buffer read format RGB565
GR_FLM_RD	GR_R_ENB	1	Frame buffer read enabled
Scaling-up Selection			
SC_SCL0_US8	SC_RES_IBUS_SYNC_SEL	0	Scaled-up video signal displayed
GR_AB1	GR_DISP_SEL[1:0]	1	Scaling display selected

38.5 Image Quality Improver

38.5.1 Image Quality Improver Functions

38.5.1.1 Overview of Functions

The image quality improver subjects scaled YCbCr signals to GBR conversion by using a color matrix.

The image quality improver does not act on RGB signals.

Figure 38.42 is a functional block diagram of the image quality improver. Image quality improvers 0 and 1 are connected to scalers 0 and 1, respectively.

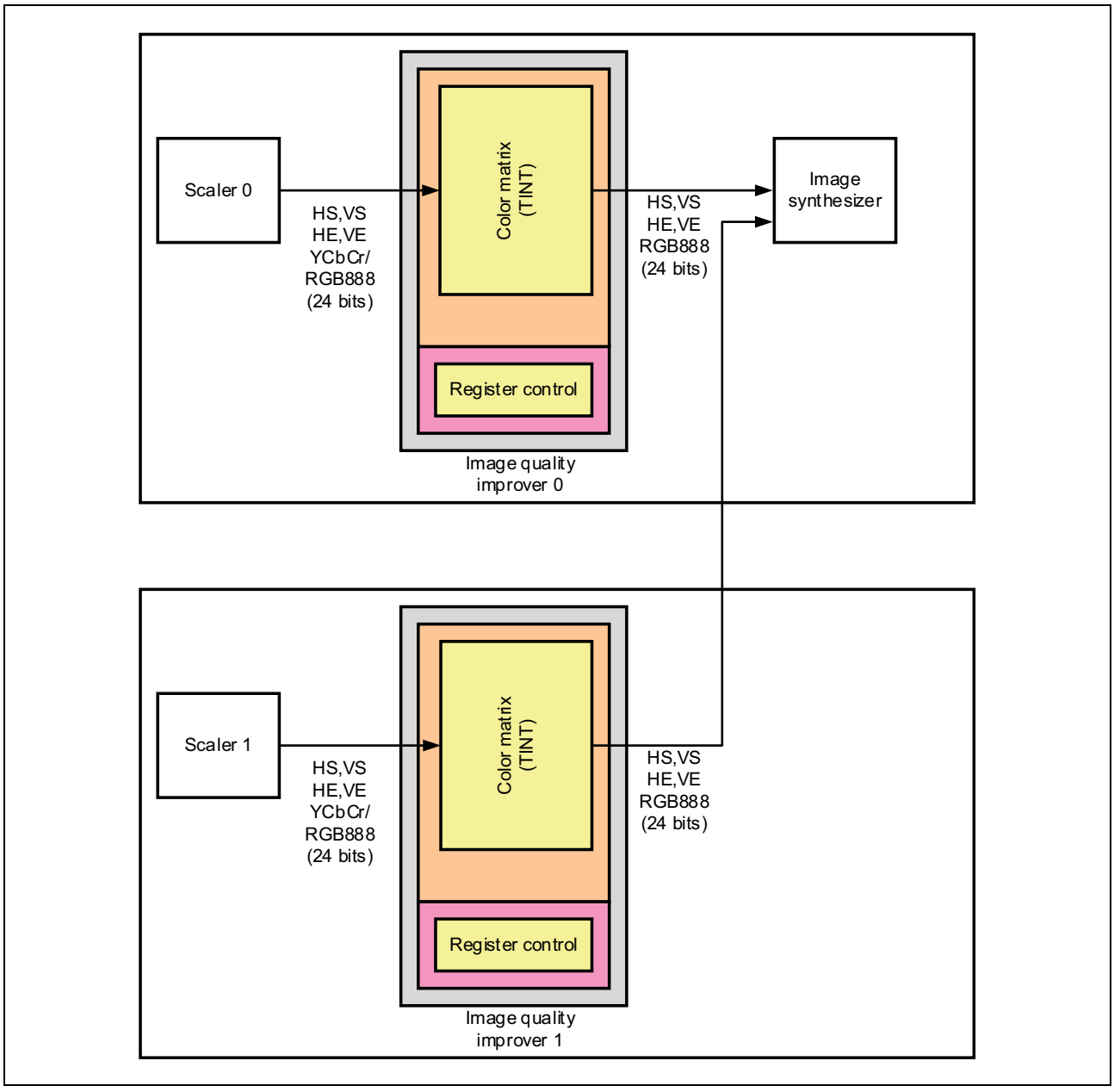


Figure 38.42 Functional Block Diagram of Image Quality Improver

38.5.1.2 Register Update Control

The control register for image quality improver controls the update timing entirely by vertical synchronous signals.

The vertical synchronous signal launched after the update control register is set to 1 is reflected in various registers, following which the update control register is automatically cleared to 0.

Note that there are two image quality improvers and registers for each improver can be identified by the number in the register name like ADJ0_xxxx and ADJ1_xxxx, respectively. In the sections except for Register Description, however, the number is omitted like ADJ_xxxx for convenience sake.

Table 38.92 Register Update Control

Register Name	Bit Name	Initial Value	Description
ADJ_UPDATE	ADJ_VEN	0	Image Quality Improver Register Update 0: Register is not updated. 1: Register is updated by launch of vertical synchronous signal.

38.5.1.3 Color Matrix

Color matrix is performed by adjusting the offset of each input signal and nine-axis gain. This allows YCbCr to GBR conversion.

(1) GBR to GBR Conversion

YGIN_A = YGIN + ADJ_MTX_YG − 128

CBBIN_A = CBBIN + ADJ_MTX_B − 128

CRRIN_A = CRRIN + ADJ_MTX_R − 128

YGOUT = (ADJ_MTX_GG × YGIN_A + ADJ_MTX_GB × CBBIN_A + ADJ_MTX_GR × CRRIN_A) ÷ 256

CBBOUT = (ADJ_MTX_BG × YGIN_A + ADJ_MTX_BB × CBBIN_A + ADJ_MTX_BR × CRRIN_A) ÷ 256

CRROUT = (ADJ_MTX_RG × YGIN_A + ADJ_MTX_RB × CBBIN_A + ADJ_MTX_RR × CRRIN_A) ÷ 256

(2) YCbCr to GBR Conversion

YGIN_A = YGIN + ADJ_MTX_YG − 128

CBBIN_A = CBBIN − 128

CRRIN_A = CRRIN − 128

YGOUT = (ADJ_MTX_GG × YGIN_A + ADJ_MTX_GB × CBBIN_A + ADJ_MTX_GR × CRRIN_A) ÷ 256

CBBOUT = (ADJ_MTX_BG × YGIN_A + ADJ_MTX_BB × CBBIN_A + ADJ_MTX_BR × CRRIN_A) ÷ 256

CRROUT = (ADJ_MTX_RG × YGIN_A + ADJ_MTX_RB × CBBIN_A + ADJ_MTX_RR × CRRIN_A) ÷ 256

Table 38.93 Matrix Coefficients (Standard Values) of SMPTE 293M

	YGIN		CBBIN		CRRIN	
	Coefficient	Bit Setting	Coefficient	Bit Setting	Coefficient	Bit Setting
YGOUT	1.000	ADJ_MTX_GG = 256	–0.344	ADJ_MTX_GB = 1960	–0.714	ADJ_MTX_GR = 1865
CBBOUT	1.000	ADJ_MTX_BG = 256	1.772	ADJ_MTX_BB = 454	0.000	ADJ_MTX_BR = 0
CRROUT	1.000	ADJ_MTX_RG = 256	0.000	ADJ_MTX_RB = 0	1.402	ADJ_MTX_RR = 359

Table 38.94 Color Matrix Control

Register Name	Bit Name	Initial Value	Description
ADJ_MTX_MODE	ADJ_MTX_MD[1:0]	2	Operating Mode 0: GBR → GBR 1: Setting prohibited 2: YCbCr → GBR 3: Setting prohibited
ADJ_MTX_YG_ADJ0	ADJ_MTX_YG[7:0]	128	Y/G Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128(0) to 255 (+127) [LSB])
ADJ_MTX_CBB_ADJ0	ADJ_MTX_B[7:0]	128	B Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
ADJ_MTX_CRR_ADJ0	ADJ_MTX_R[7:0]	128	R Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
ADJ_MTX_YG_ADJ0	ADJ_MTX_GG[10:0]	256	Gain Adjustment of Y/G Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_YG_ADJ1	ADJ_MTX_GB[10:0]	1960	Gain Adjustment of Cb/B Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_YG_ADJ1	ADJ_MTX_GR[10:0]	1865	Gain Adjustment of Cr/R Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_CBB_ADJ0	ADJ_MTX_BG[10:0]	256	Gain Adjustment of Y/G Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_CBB_ADJ1	ADJ_MTX_BB[10:0]	454	Gain adjustment of Cb/B signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_CBB_ADJ1	ADJ_MTX_BR[10:0]	0	Gain Adjustment of Cr/R Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_CRR_ADJ0	ADJ_MTX_RG[10:0]	256	Gain Adjustment of Y/G Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_CRR_ADJ1	ADJ_MTX_RB[10:0]	0	Gain Adjustment of Cb/B Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
ADJ_MTX_CRR_ADJ1	ADJ_MTX_RR[10:0]	359	Gain Adjustment of Cr/R Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

38.5.2 Register Description

Table 38.95 and **Table 38.96** show the register configurations.

[Symbols used in Register Description]

Initial value: Register value after a reset

—: Undefined value

R/W: Readable/writable. The written value can be read.

R/WC0: Read and write. Bit is initialized if 0 is written, and ignored if 1 is written.

R/WC1: Read and write. Bit is initialized if 1 is written, and ignored if 0 is written.

R: Read-only. The write value should always be 0.

—/W: Write-only. Read value is undefined.

Table 38.95 Image Quality Improver Register Configuration (Channel 0)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Register update control register in image quality improver (image quality improver 0)	ADJ0_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1680	32
VDCE0	Matrix mode register in image quality improver (image quality improver 0)	ADJ0_MTX_MODE	R/W	H'0000 0002	<VDCE0_base> + 16B4	32
VDCE0	Matrix YG control register 0 in image quality improver (image quality improver 0)	ADJ0_MTX_YG_ADJ0	R/W	H'0080 0100	<VDCE0_base> + 16B8	32
VDCE0	Matrix YG control register 1 in image quality improver (image quality improver 0)	ADJ0_MTX_YG_ADJ1	R/W	H'07A8 0749	<VDCE0_base> + 16BC	32
VDCE0	Matrix CBB control register 0 in image quality improver (image quality improver 0)	ADJ0_MTX_CBB_ADJ0	R/W	H'0080 0100	<VDCE0_base> + 16C0	32
VDCE0	Matrix CBB control register 1 in image quality improver (image quality improver 0)	ADJ0_MTX_CBB_ADJ1	R/W	H'01C6 0000	<VDCE0_base> + 16C4	32
VDCE0	Matrix CRR control register 0 in image quality improver (image quality improver 0)	ADJ0_MTX_CRR_ADJ0	R/W	H'0080 0100	<VDCE0_base> + 16C8	32
VDCE0	Matrix CRR control register 1 in image quality improver (image quality improver 0)	ADJ0_MTX_CRR_ADJ1	R/W	H'0000 0167	<VDCE0_base> + 16CC	32
VDCE0	Register update control register in image quality improver (image quality improver 1)	ADJ1_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1D80	32
VDCE0	Matrix mode register in image quality improver (image quality improver 1)	ADJ1_MTX_MODE	R/W	H'0000 0002	<VDCE0_base> + 1DB4	32
VDCE0	Matrix YG control register 0 in image quality improver (image quality improver 1)	ADJ1_MTX_YG_ADJ0	R/W	H'0080 0100	<VDCE0_base> + 1DB8	32
VDCE0	Matrix YG control register 1 in image quality improver (image quality improver 1)	ADJ1_MTX_YG_ADJ1	R/W	H'07A8 0749	<VDCE0_base> + 1DBC	32
VDCE0	Matrix CBB control register 0 in image quality improver (image quality improver 1)	ADJ1_MTX_CBB_ADJ0	R/W	H'0080 0100	<VDCE0_base> + 1DC0	32
VDCE0	Matrix CBB control register 1 in image quality improver (image quality improver 1)	ADJ1_MTX_CBB_ADJ1	R/W	H'01C6 0000	<VDCE0_base> + 1DC4	32
VDCE0	Matrix CRR control register 0 in image quality improver (image quality improver 1)	ADJ1_MTX_CRR_ADJ0	R/W	H'0080 0100	<VDCE0_base> + 1DC8	32
VDCE0	Matrix CRR control register 1 in image quality improver (image quality improver 1)	ADJ1_MTX_CRR_ADJ1	R/W	H'0000 0167	<VDCE0_base> + 1DCC	32

Table 38.96 Image Quality Improver Register Configuration (Channel 1)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Register update control register in image quality improver (image quality improver 0)	ADJ0_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1680	32
VDCE1	Matrix mode register in image quality improver (image quality improver 0)	ADJ0_MTX_MODE	R/W	H'0000 0002	<VDCE1_base> + 16B4	32
VDCE1	Matrix YG control register 0 in image quality improver (image quality improver 0)	ADJ0_MTX_YG_ADJ0	R/W	H'0080 0100	<VDCE1_base> + 16B8	32
VDCE1	Matrix YG control register 1 in image quality improver (image quality improver 0)	ADJ0_MTX_YG_ADJ1	R/W	H'07A8 0749	<VDCE1_base> + 16BC	32
VDCE1	Matrix CBB control register 0 in image quality improver (image quality improver 0)	ADJ0_MTX_CBB_ADJ0	R/W	H'0080 0100	<VDCE1_base> + 16C0	32
VDCE1	Matrix CBB control register 1 in image quality improver (image quality improver 0)	ADJ0_MTX_CBB_ADJ1	R/W	H'01C6 0000	<VDCE1_base> + 16C4	32
VDCE1	Matrix CRR control register 0 in image quality improver (image quality improver 0)	ADJ0_MTX_CRR_ADJ0	R/W	H'0080 0100	<VDCE1_base> + 16C8	32
VDCE1	Matrix CRR control register 1 in image quality improver (image quality improver 0)	ADJ0_MTX_CRR_ADJ1	R/W	H'0000 0167	<VDCE1_base> + 16CC	32
VDCE1	Register update control register in image quality improver (image quality improver 1)	ADJ1_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1D80	32
VDCE1	Matrix mode register in image quality improver (image quality improver 1)	ADJ1_MTX_MODE	R/W	H'0000 0002	<VDCE1_base> + 1DB4	32
VDCE1	Matrix YG control register 0 in image quality improver (image quality improver 1)	ADJ1_MTX_YG_ADJ0	R/W	H'0080 0100	<VDCE1_base> + 1DB8	32
VDCE1	Matrix YG control register 1 in image quality improver (image quality improver 1)	ADJ1_MTX_YG_ADJ1	R/W	H'07A8 0749	<VDCE1_base> + 1DBC	32
VDCE1	Matrix CBB control register 0 in image quality improver (image quality improver 1)	ADJ1_MTX_CBB_ADJ0	R/W	H'0080 0100	<VDCE1_base> + 1DC0	32
VDCE1	Matrix CBB control register 1 in image quality improver (image quality improver 1)	ADJ1_MTX_CBB_ADJ1	R/W	H'01C6 0000	<VDCE1_base> + 1DC4	32
VDCE1	Matrix CRR control register 0 in image quality improver (image quality improver 1)	ADJ1_MTX_CRR_ADJ0	R/W	H'0080 0100	<VDCE1_base> + 1DC8	32
VDCE1	Matrix CRR control register 1 in image quality improver (image quality improver 1)	ADJ1_MTX_CRR_ADJ1	R/W	H'0000 0167	<VDCE1_base> + 1DCC	32

NOTE

Register access sizes other than defined in the table above are not supported.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

38.5.2.1 Register Update Control Register in Image Quality Improver (ADJ0_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADJ0_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ADJ0_VEN	0	R/WC1	Image Quality Improver Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.5.2.2 Matrix Mode Register in Image Quality Improver (ADJ0_MTX_MODE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADJ0_MTX_MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	ADJ0_MTX_MD[1:0]	2	R/W	Operating Mode 0: GBR → GBR 1: Setting prohibited 2: YCbCr → GBR 3: Setting prohibited

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

38.5.2.3 Matrix YG Control Register 0 in Image Quality Improver (ADJ0_MTX_YG_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADJ0_MTX_YG[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ0_MTX_GG[10:0]										
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	ADJ0_MTX_YG[7:0]	128	R/W	Y/G Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ0_MTX_GG[10:0]	256	R/W	Gain Adjustment of Y/G Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

38.5.2.4 Matrix YG Control Register 1 in Image Quality Improver (ADJ0_MTX_YG_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ADJ0_MTX_GB[10:0]										
Initial value:	0	0	0	0	0	1	1	1	1	0	1	0	1	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ0_MTX_GR[10:0]										
Initial value:	0	0	0	0	0	1	1	1	0	1	0	0	1	0	0	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	ADJ0_MTX_GB[10:0]	1960	R/W	Gain Adjustment of Cb/B Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ0_MTX_GR[10:0]	1865	R/W	Gain Adjustment of Cr/R Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

38.5.2.5 Matrix CBB Control Register 0 in Image Quality Improver (ADJ0_MTX_CBB_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADJ0_MTX_B[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ0_MTX_BG[10:0]										
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	ADJ0_MTX_B [7:0]	128	R/W	B Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ0_MTX_BG [10:0]	256	R/W	Gain Adjustment of Y/G Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

38.5.2.6 Matrix CBB Control Register 1 in Image Quality Improver (ADJ0_MTX_CBB_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ADJ0_MTX_BB[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ0_MTX_BR[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	ADJ0_MTX_BB [10:0]	454	R/W	Gain Adjustment of Cb/B Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ0_MTX_BR [10:0]	0	R/W	Gain Adjustment of Cr/R Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

38.5.2.7 Matrix CRR Control Register 0 in Image Quality Improver (ADJ0_MTX_CRR_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADJ0_MTX_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ0_MTX_RG[10:0]										
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	ADJ0_MTX_R [7:0]	128	R/W	R Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ0_MTX_RG [10:0]	256	R/W	Gain Adjustment of Y/G Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

38.5.2.8 Matrix CRR Control Register 1 in Image Quality Improver (ADJ0_MTX_CRR_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ADJ0_MTX_RB[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ0_MTX_RR[10:0]										
Initial value:	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	ADJ0_MTX_RB [10:0]	0	R/W	Gain Adjustment of Cb/B Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ0_MTX_RR [10:0]	359	R/W	Gain Adjustment of Cr/R Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ0_VEN in ADJ0_UPDATE is 1.

38.5.2.9 Register Update Control Register in Image Quality Improver (ADJ1_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADJ1_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	ADJ1_VEN	0	R/WC1	Image Quality Improver Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.5.2.10 Matrix Mode Register in Image Quality Improver (ADJ1_MTX_MODE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADJ1_MTX_MD[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	ADJ1_MTX_MD[1:0]	2	R/W	Operating Mode 0: GBR → GBR 1: Setting prohibited 2: YCbCr → GBR 3: Setting prohibited

Note: This register is updated when ADJ1_VEN in ADJ1_UPDATE is 1.

38.5.2.11 Matrix YG Control Register 0 in Image Quality Improver (ADJ1_MTX_YG_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADJ1_MTX_YG[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ1_MTX_GG[10:0]										
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	ADJ1_MTX_YG[7:0]	128	R/W	Y/G Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ1_MTX_GG[10:0]	256	R/W	Gain Adjustment of Y/G Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ1_VEN in ADJ1_UPDATE is 1.

38.5.2.12 Matrix YG Control Register 1 in Image Quality Improver (ADJ1_MTX_YG_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ADJ1_MTX_GB[10:0]										
Initial value:	0	0	0	0	0	1	1	1	1	0	1	0	1	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ1_MTX_GR[10:0]										
Initial value:	0	0	0	0	0	1	1	1	0	1	0	0	1	0	0	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	ADJ1_MTX_GB[10:0]	1960	R/W	Gain Adjustment of Cb/B Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ1_MTX_GR[10:0]	1865	R/W	Gain Adjustment of Cr/R Signal of G Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ1_VEN in ADJ1_UPDATE is 1.

38.5.2.13 Matrix CBB Control Register 0 in Image Quality Improver (ADJ1_MTX_CBB_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADJ1_MTX_B[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ1_MTX_BG[10:0]										
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	ADJ1_MTX_B [7:0]	128	R/W	B Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ1_MTX_BG [10:0]	256	R/W	Gain Adjustment of Y/G Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ1_VEN in ADJ1_UPDATE is 1.

38.5.2.14 Matrix CBB Control Register 1 in Image Quality Improver (ADJ1_MTX_CBB_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ADJ1_MTX_BB[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ1_MTX_BR[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	ADJ1_MTX_BB [10:0]	454	R/W	Gain Adjustment of Cb/B Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ1_MTX_BR [10:0]	0	R/W	Gain Adjustment of Cr/R Signal of B Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ1_VEN in ADJ1_UPDATE is 1.

38.5.2.15 Matrix CRR Control Register 0 in Image Quality Improver (ADJ1_MTX_CRR_ADJ0)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADJ1_MTX_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ1_MTX_RG[10:0]										
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	ADJ1_MTX_R [7:0]	128	R/W	R Signal Offset (DC) Adjustment Unsigned (0 (-128) to 128 (0) to 255 (+127) [LSB])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ1_MTX_RG [10:0]	256	R/W	Gain Adjustment of Y/G Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ1_VEN in ADJ1_UPDATE is 1.

38.5.2.16 Matrix CRR Control Register 1 in Image Quality Improver (ADJ1_MTX_CRR_ADJ1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	ADJ1_MTX_RB[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	ADJ1_MTX_RR[10:0]										
Initial value:	0	0	0	0	0	0	0	1	0	1	1	0	0	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	ADJ1_MTX_RB [10:0]	0	R/W	Gain Adjustment of Cb/B Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	ADJ1_MTX_RR [10:0]	359	R/W	Gain Adjustment of Cr/R Signal of R Signal Output Signed (complement of 2) (-1024 to +1023 [LSB], 256 [LSB] = 1.0 [times])

Note: This register is updated when ADJ1_VEN in ADJ1_UPDATE is 1.

38.5.3 Usage Method

38.5.3.1 Setting Method for Color Matrix Data Conversion

GBR signals are assumed to be input to the circuit subsequent to the image quality improver; therefore, the output from the color matrix circuit should be in the GBR format.

Table 38.97 shows an example of GBR conversion setting.

Table 38.97 Recommended Setting Values for Matrix Conversion

		GBR to GBR Conversion	YCBCR to GBR Conversion
Register Name	**Bit Name**	**Recommended Values**	**Recommended Values**
ADJ_MTX_MODE	ADJ_MTX_MD[1:0]	0	2
ADJ_MTX_YG_ADJ0	ADJ_MTX_YG[7:0]	128	128
ADJ_MTX_CBB_ADJ0	ADJ_MTX_B[7:0]	128	128
ADJ_MTX_CRR_ADJ0	ADJ_MTX_R[7:0]	128	128
ADJ_MTX_YG_ADJ0	ADJ_MTX_GG[10:0]	256	256
ADJ_MTX_YG_ADJ1	ADJ_MTX_GB[10:0]	0	1960
ADJ_MTX_YG_ADJ1	ADJ_MTX_GR[10:0]	0	1865
ADJ_MTX_CBB_ADJ0	ADJ_MTX_BG[10:0]	0	256
ADJ_MTX_CBB_ADJ1	ADJ_MTX_BB[10:0]	256	454
ADJ_MTX_CBB_ADJ1	ADJ_MTX_BR[10:0]	0	0
ADJ_MTX_CRR_ADJ0	ADJ_MTX_RG[10:0]	0	256
ADJ_MTX_CRR_ADJ1	ADJ_MTX_RB[10:0]	0	0
ADJ_MTX_CRR_ADJ1	ADJ_MTX_RR[10:0]	256	359

Note: ADJ_VEN in ADJ_UPDATE should be set to 1 after setting the registers.

38.6 Image Synthesizer

38.6.1 Image Synthesizer Functions

38.6.1.1 Overview of Functions

The image synthesizer reads graphics data from the frame buffer and displays the synthesized image on the screen.

Two video planes + two graphics planes, one video plane + three graphics planes, or four graphics planes can be selected for synthesis.

RGB565, RGB888, αRGB1555, αRGB4444, αRGB8888, RGBα5551, RGBα8888, CLUT8, CLUT4, CLUT1, YCbCr422 (for the graphics 0 and 1 processes), and YCbCr444 (for the graphics 0 and 1 processes) formats can be used for graphics data, and RGB565, RGB888, YCbCr422, and YCbCr444 formats for video data.

On each of the graphics planes, background color, lower-layer graphics, current graphics, or blended image (for the graphics 1, 2, and 3 processes and VIN synthesizer) of lower-layer graphics and current graphics can be displayed.

In the VIN synthesizer, graphics data is not read and images (for the graphics 0 and 1 processes) are blended.

The functional block diagram of the image synthesizer is shown below.

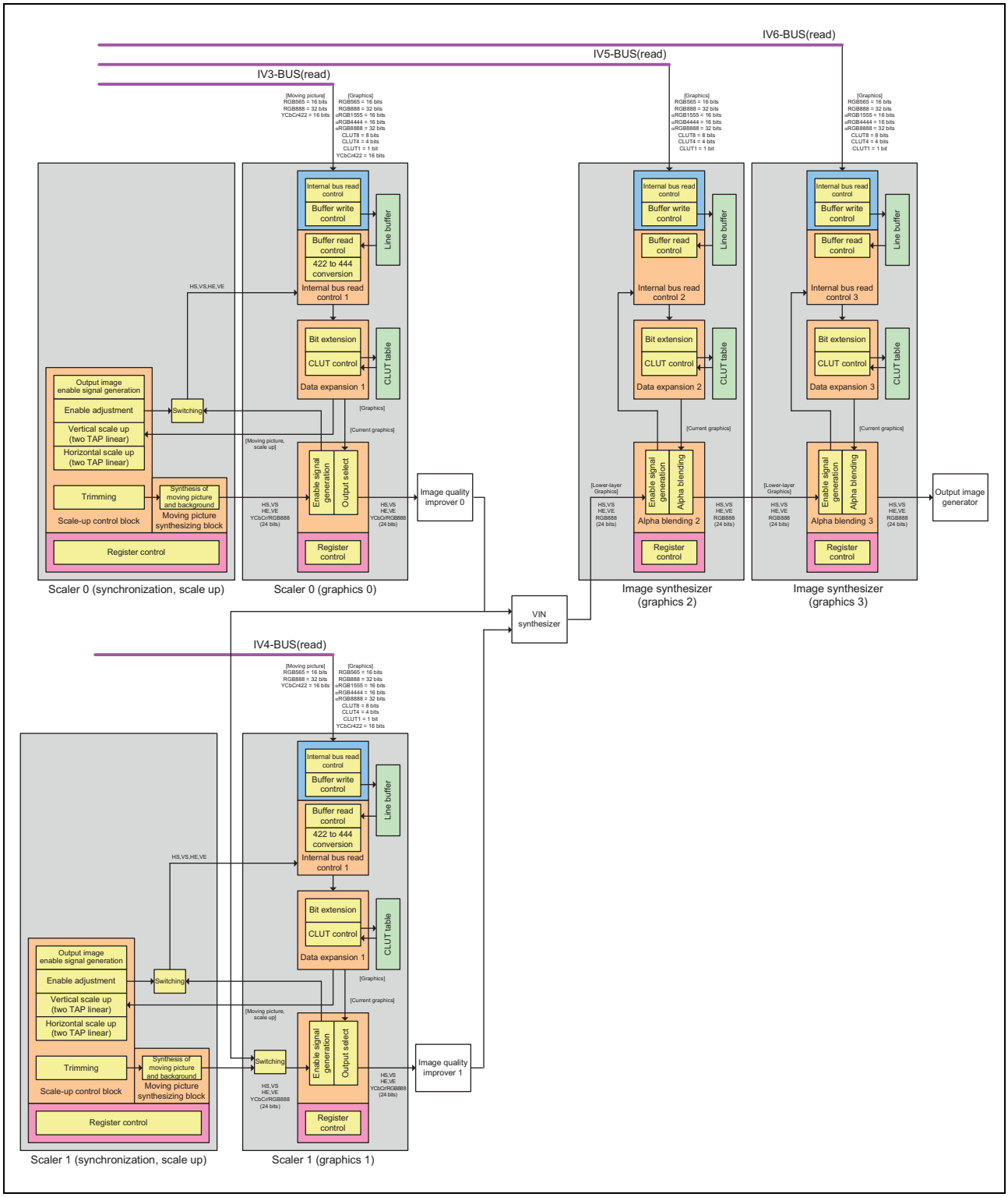


Figure 38.43 Functional Block Diagram of Image Synthesizer

38.6.1.2 Graphics Data Read Control

Graphics data read can be controlled for the five processes: the graphics 0 process in the scaler 0, the graphics 1 process in the scaler 1, the graphics 2 and 3 processes in the image synthesizer, and the graphics OIR process in the output image generator.

The register bits of each process can be identified by the number in the register name like GR0_xxxx, GR1_xxxx, GR2_xxxx, GR3_xxxx, and GR_OIR_xxxx, respectively. In the sections except for Register Descriptions, however, the number is omitted like GR_xxxx for convenience sake.

In the VIN synthesizer, graphics data read is not controlled and the blending register GR_VIN_xxxx is present. In this manual, the name is omitted like GR_xxxx for convenience sake. The synthesizer does not have the read control register (GR_FLM).

(1) Updating Registers

The Vsync signal is used to control the update timing of the registers for graphics display and frame buffer read control, except for some of the registers.

After 1 is set to the bits in the update control register, the contents of the relevant registers are actually modified at the rising edge of the Vsync signal, when the update control register is automatically cleared to 0.

Table 38.98 Register Update Control

Register Name	Bit Name	Initial Value	Description
GR_UPDATE	GR_UPDATE	0	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated.
GR_UPDATE	GR_P_VEN	0	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GR_UPDATE	GR_IBUS_VEN	0	Frame Buffer Read Register Update*1 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

Note 1. This bit is not supported for the VIN synthesizer.

(2) Frame Buffer Control Mode

More than one frame of data is read from the frame buffer.

For graphics data, set the GR_FLM_SEL[1:0] bits to 1, and set the specific display frame number with the GR_FLM_NUM[9:0] bits. For video data, select a mode with the GR_FLM_SEL[1:0] bits depending on the writing process used; the quantity of the frames used for video data is set in the writing process block.

Table 38.99 Frame Buffer Control Mode

Register Name	Bit Name	Initial Value	Description
GR_FLM1	GR_FLM_SEL[1:0]	0	Frame Buffer Address Setting Signal Select 0: Control linked to scaling-down process, or frame 0 selected. *1 1: Register GR_FLM_NUM selected. 2: Control linked to distortion correction, or frame 0 selected. *2 3: Control linked to pointer buffer, or setting prohibited. *3
GR_FLM3	GR_FLM_NUM[9:0]	0	Frame Number of Frame Buffer Manually set the frame number when GR_FLM_SEL = 1.

Note 1. For the graphics 0 process, frame buffer control links to the scaling-down process. For the graphics 2 and 3 processes, frame 0 is selected.

Note 2. For the graphics OIR process, frame buffer control links to distortion correction. For the graphics 1, 2 and 3

processes, frame 0 is selected.

Note 3. This configuration is prohibit for graphics 2 and 3 .

(3) Frame Buffer Read Control

The following bit enables or disables read access to the frame buffer.

Table 38.100 Frame Buffer Read Control

Register Name	Bit Name	Initial Value	Description
GR_FLM_RD	GR_R_ENB	0	Frame Buffer Read Enable 0: Disables read access to the frame buffer. 1: Enables read access to the frame buffer.

(4) Frame Buffer Size

The following bits set the size of the frame buffer to be read.

The numbers of horizontal pixels and of lines in the vertical direction are set with the GR_HW[10:0] and GR_FLM_LNUM[10:0] bits, respectively.

Table 38.101 Frame Buffer Size

Register Name	Bit Name	Initial Value	Description
GR_FLM6	GR_HW[10:0]	0	Sets the width of the horizontal valid period. The width is (GR_HW + 1) pixels. Note: Set to 2 or greater.
GR_FLM5	GR_FLM_LNUM[10:0]	0	Sets the number of lines in a frame The number of lines is (GR_FLM_LNUM + 1).

(5) Calculating Addresses in Frame Buffer

The data area in the frame buffer is defined using the addresses specified by GR_BASE[31:0], GR_LN_OFF[14:0], and GR_FLM_OFF[22:0] bits and the display frame number.

The GR_LN_OFF[14:0] and GR_FLM_OFF[22:0] bits should be set in units of 128 bytes (the lower seven bits should be fixed to 000 0000).

The GR_BASE[31:0] bits should be set in units of 128 bytes to set the display data start position (the lower seven bits should be fixed).

(6) Restriction

For any buffer that is read by any VDCE layer (GR0-GR3 and OIR), following care must be taken. In addition to the memory area occupied by the frame buffer, the next 256 bytes of memory must be readable by the same VDCE layer. The layer will neither use or modify the data in this area, so it may be used by any other VDCE layer, VOWE Drawing List, GPU Drawing List or other macro.

The memory area may overlap into other address regions given that they are allowed to be read. This excludes reserved areas as well as unavailable areas (e.g. SDRA, if external SDRAM is not connected).

Recommendations:

If SDRAM is not available, place GPU working memory or VOWE command list at the end of VRAM. Remaining VRAM can be used without restrictions.

If SDRAM is available, place GPU working memory or VOWE command list at the end of SDRAM. Remaining VRAM and SDRAM can be used without restrictions.

Example:

SDRAM is available from 4000 0000_H to 5FFF $FFFF_H$. We have a buffer of 320 x 240, 4 BPP, Stride: 256 px using 5 0000_H bytes of memory.

If you place it into SDRAM area, do not use 5FFB 0000_H - 5FFF $FFFF_H$, as the next 256 bytes (6000 0000_H - 6000 $00FF_H$) are reserved addresses.

Use 5FFA $FF00_H$ - 5FFF $FF00_H$ or any smaller start address down to the start of VRAM area.

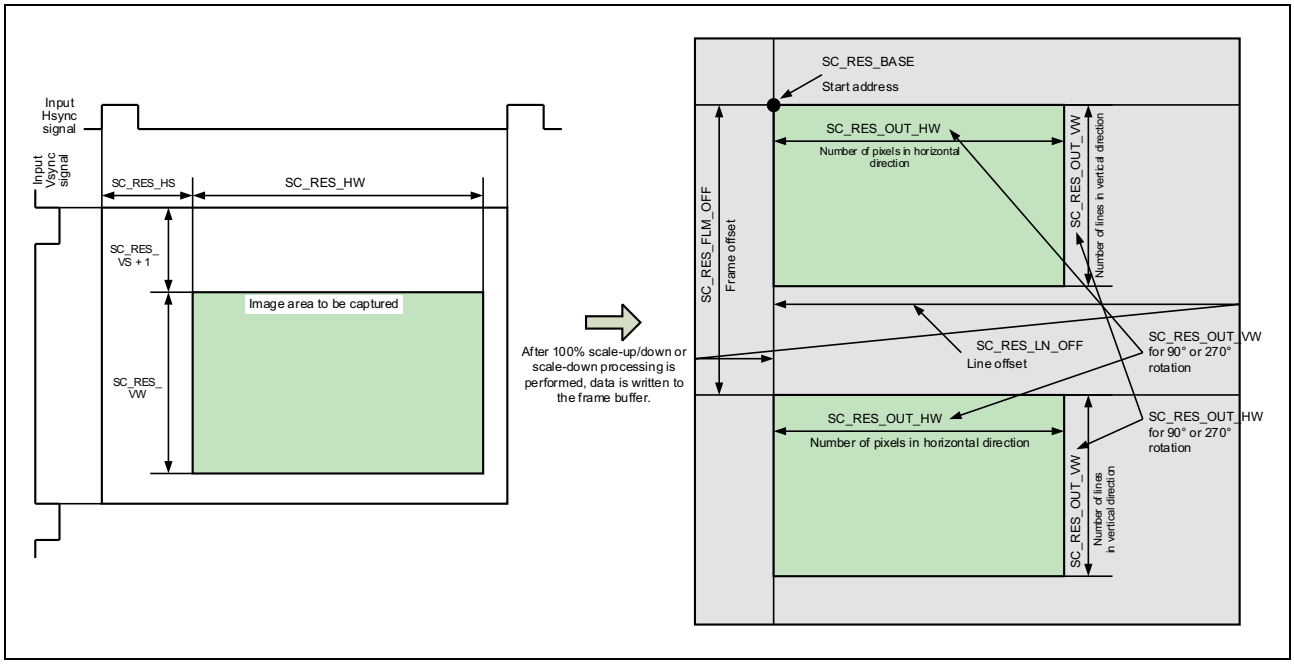


Figure 38.44 Data Arrangement in Frame Buffer

Table 38.102 Calculation of Addresses in Frame Buffer

Register Name	Bit Name	Initial Value	Description
GR_FLM2	GR_BASE[31:0]	0	Frame Buffer Base Address Sets the start address of the frame buffer where frame data is to be stored. The lower seven bits should be fixed to 000 0000.
GR_FLM3	GR_LN_OFF[14:0]	0	Frame Buffer Line Offset Address Sets the line offset address for calculating the start address of each line. Line 0:GR_BASE Line 1:GR_BASE + GR_LN_OFF × 1 : Line n: GR_BASE + GR_LN_OFF × n The lower seven bits should be fixed to 000 0000.
GR_FLM4	GR_FLM_OFF[22:0]	0	Frame Buffer Frame Offset Address (lower) Specifies the frame offset address used for calculating the start address of each frame buffer when more than one buffer is used. Buffer 0: GR_BASE Buffer 1: GR_BASE + GR_FLM_OFF × 1 : Buffer n: GR_BASE + GR_FLM_OFF × n The lower seven bits should be fixed to 000 0000.

(7) Setting Frame Buffer Size Smaller than One Frame

Frame buffer size can be set in one-line units.

When the number of lines set with the GR_FLM_LOOP[10:0] bits is smaller than the value of the GR_FLM_LNUM[10:0] bits, data is again read from the start address of the frame buffer after the number of lines set with the (GR_FLM_LOOP[10:0] + 1) bits have been read.

Table 38.103 Setting of Frame Buffer Size Smaller than One Frame

Register Name	Bit Name	Initial Value	Description
GR_FLM5	GR_FLM_LOOP[10:0]	1023	Number of lines when reading the addresses repeatedly by returning to the start address after reaching the end address. The number of lines is (GR_FLM_LOOP + 1).

(8) Line Offset Control for Frame Buffer

The following bit sets the line offset address direction of the frame buffer.

Table 38.104 Line Offset Address Direction Control for Frame Buffer

Register Name	Bit Name	Initial Value	Description
GR_FLM1	GR_LN_OFF_DIR	0	Selects the line offset address direction of the frame buffer. 0:Increments the address by the line offset address. 1:Decrements the address by the line offset address.

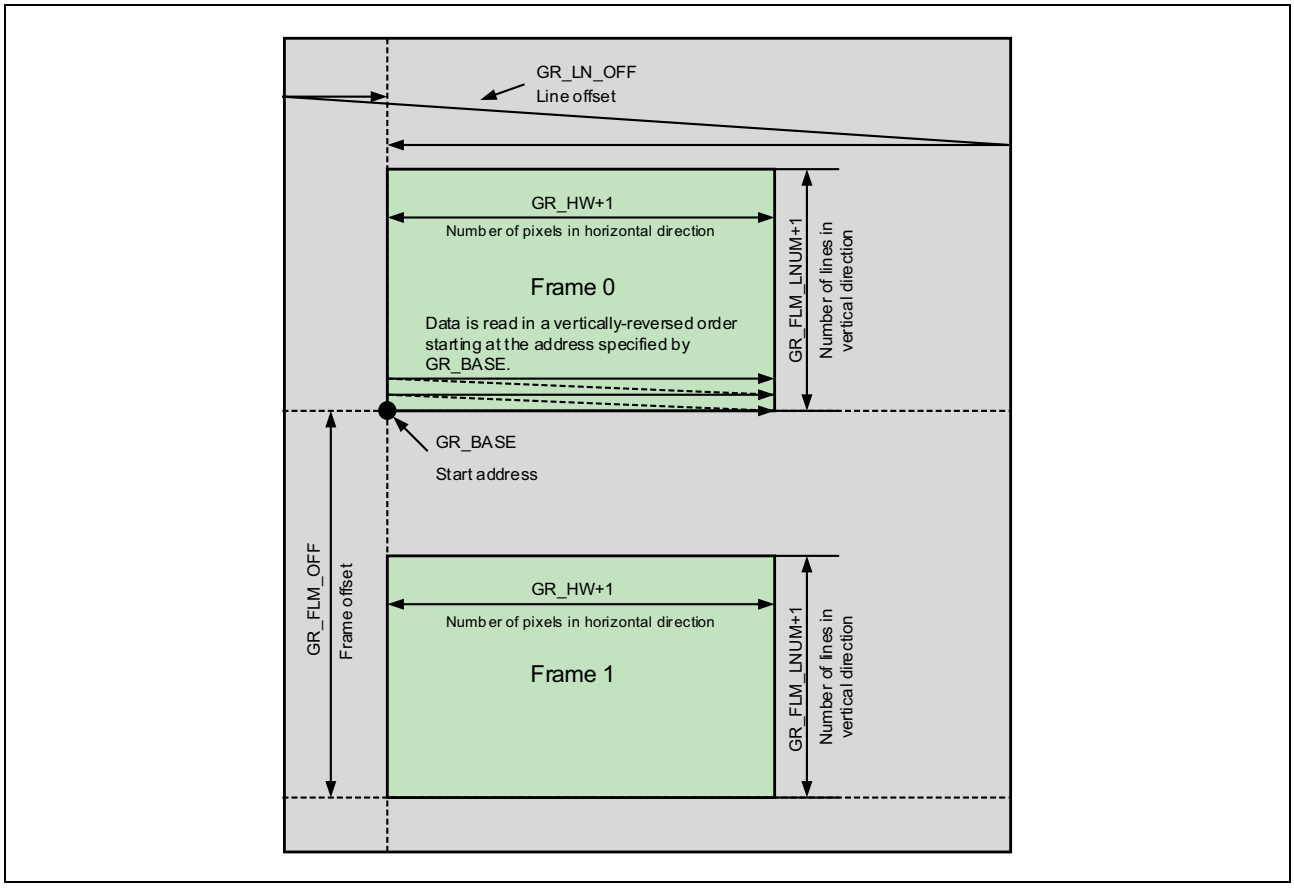


Figure 38.45 Data Arrangement with Line Offset and Decrement Control

(9) Selecting Format of Frame Buffer Read Signal

Signal formats RGB565, RGB888, αRGB1555, αRGB4444, αRGB8888, RGBα5551, RGBα8888, CLUT8, CLUT4 and CLUT1 are supported for the graphics 0, 1, 2, 3, and OIR processes. The YCbCr422 and YCbCr444 formats are also supported for the graphics 0 and 1 processes.

The GR_FORMAT[3:0] bits select a signal format.

Table 38.105 Format Selection for Frame Buffer Read Signal

Register Name	Bit Name	Initial Value	Description
GR_FLM6	GR_FORMAT[3:0]	0	Sets the format of the frame buffer read signal. 0: RGB565 1: RGB888 2: αRGB1555 3: αRGB4444 4: αRGB8888 5: CLUT8 6: CLUT4 7: CLUT1 8: YCbCr422 or setting prohibited *1 9: YCbCr444 or setting prohibited *1 10: RGBα5551 11: RGBα8888 12 to 15: Setting prohibited

Note 1. Setting this value selects YCbCr422 and YCbCr444 for the graphics 0 and 1 processes, and is prohibited for the graphics 2, 3, and OIR processes.

(10) Endian Control

In the frame buffer, data is handled in 64-bit units, and endian of the data to be read can be controlled by setting the GR_RDSWA[2:0] bits. Bit 0 of these bits indicates whether 8-bit data is swapped. Bit 1 indicates whether 16-bit data is swapped. Bit 2 indicates whether 32-bit data is swapped. In the YCbCr422 format, data can be arranged with the GR_YCC_SWAP[2:0] bits.

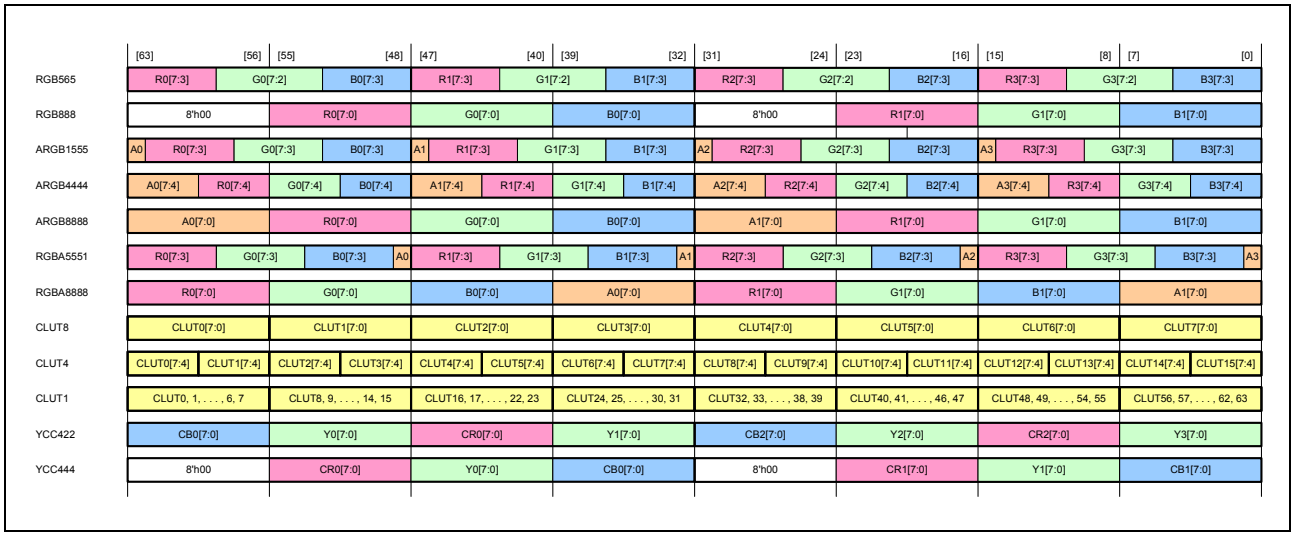


Figure 38.46 Data Arrangement with Endian Control Disabled (GR_RDSWA = 000)

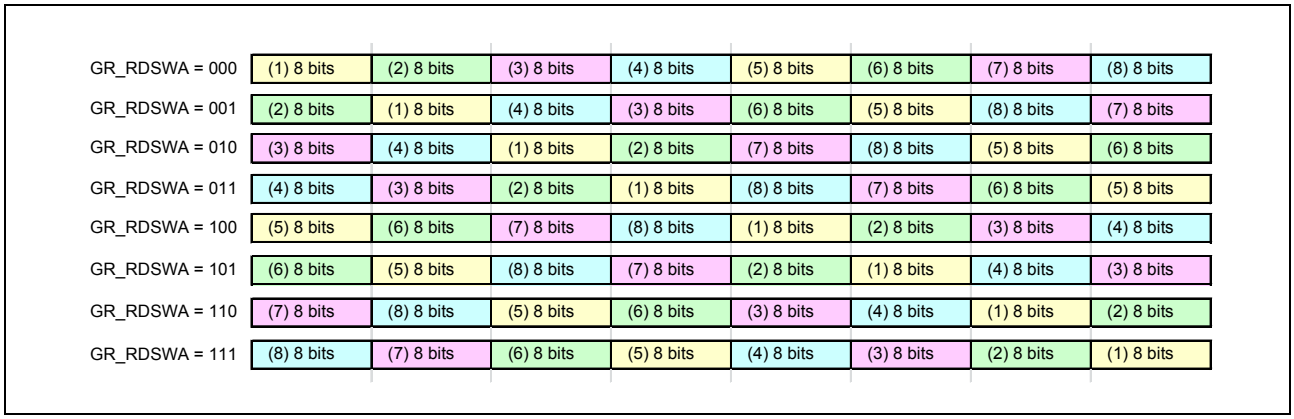


Figure 38.47 Data Arrangement with Endian Control Enabled

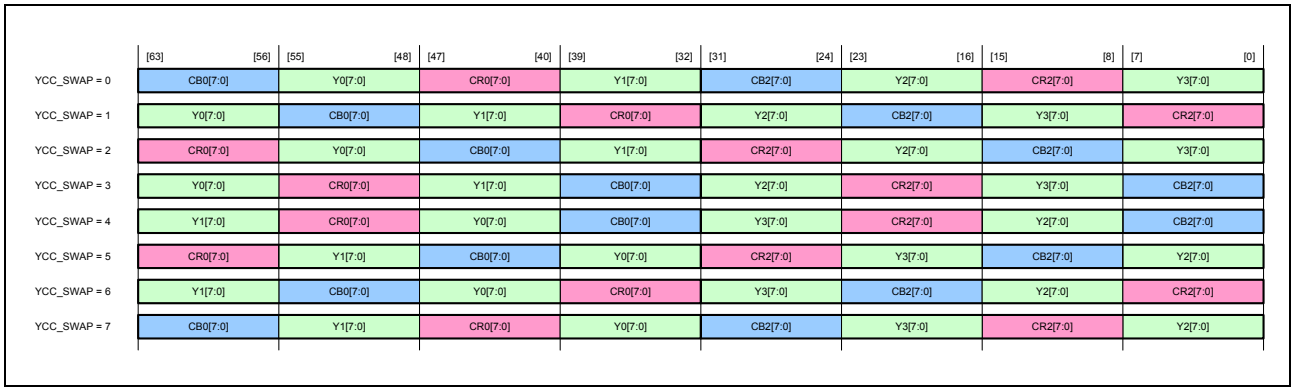


Figure 38.48 YCbCr422 Data Arrangement with Swapping Enabled

Table 38.106 Endian Control

Register Name	Bit Name	Initial Value	Description
GR_FLM6	GR_RDSWA[2:0]	0	Sets 8-, 16-, and 32-bit swap. These three bits specify the method for swapping the bits of frame buffer read data as follows. Bit 0 0: 8 bits are not swapped. 1: 8 bits are swapped. Bit 1 0: 16 bits are not swapped. 1: 16 bits are swapped. Bit 2 0: 32 bits are not swapped. 1: 32 bits are swapped. When eight bits are put together, they are swapped as follows. Each of (1) to (8) indicates eight-bit data. 000: (1) (2) (3) (4) (5) (6) (7) (8) [No swap] 001: (2) (1) (4) (3) (6) (5) (8) (7) [8-bit swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [16-bit swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [16-bit swap + 8-bit swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [32-bit swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [32-bit swap + 8-bit swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [32-bit swap + 16-bit swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [32-bit swap + 16-bit swap + 8-bit swap]
GR_FLM6	GR_YCC_SWAP [2:0]	0	Controls swapping of data read from buffer in the YCbCr422 format. *1 0: Cb/Y0/Cr/Y1 1: Y0/Cb/Y1/Cr 2: Cr/Y0/Cb/Y1 3: Y0/Cr/Y1/Cb 4: Y1/Cr/Y0/Cb 5: Cr/Y1/Cb/Y0 6: Y1/Cb/Y0/Cr 7: Cb/Y1/Cr/Y0

Note 1. These bits are supported for the graphics 0 and 1 processes only.

(11) Display Start Pixel Setting for Read Data

When a horizontal offset is applied to display the image data in the frame buffer, the display start pixel is set with the GR_BASE[31:0] and GR_STA_POS[5:0] bits. Calculation of the values for the GR_BASE[31:0] and GR_STA_POS[5:0] bits depends on the signal format. The display start pixel can be calculated with the formulas in the table below, where H_OFF is a horizontal offset from the display start pixel.

Table 38.107 Calculation of Display Start Position for Various Signal Formats

Signal Format of Video/Graphics	Number of Bits per Pixel	Calculation Formula *1
RGB888 αRGB8888, RGBα8888 YCbCr422*2 YCbCr444*3	32	GR_BASE[31:7] += int (H_OFF ÷ 2) * BPP GR_STA_POS[5:0] = mod (H_OFF ÷ 2)
RGB565 αRGB1555, RGBα5551 αRGB4444	16	GR_BASE[31:7] += int (H_OFF ÷ 4) * BPP GR_STA_POS[5:0] = mod (H_OFF ÷ 4)
CLUT8	8	GR_BASE[31:7] += int (H_OFF ÷ 8) * BPP GR_STA_POS[5:0] = mod (H_OFF ÷ 8)
CLUT4	4	GR_BASE[31:7] += int (H_OFF ÷ 16) * BPP GR_STA_POS[5:0] = mod (H_OFF ÷ 16)
CLUT1	1	GR_BASE[31:7] += int (H_OFF ÷ 64) * BPP GR_STA_POS[5:0] = mod (H_OFF ÷ 64)

Note 1. The functions int() and mod() output a quotient and a remainder, respectively.
Note 2. The YCbCr422 format is not supported for the graphics 2, 3, and OIR processes.
In the YCbCr422 format, 32 bits are used for two pixels (Cb, Y0, Cr, and Y1 components). Therefore, the start position is controlled in units of 32 bits.
Note 3. The YCbCr444 format is not supported for the graphics 2, 3, and OIR processes.

Table 38.108 Setting of Display Start Pixel of Read Data

Register Name	Bit Name	Initial Value	Description
GR_FLM6	GR_STA_POS[5:0]	0	Sets the amount of pixels to be skipped through. Specifically pixels amount equal to the amount indicated by GR_STA_POS is skipped from the start of the line.
GR_FLM2	GR_BASE[31:0]	0	Frame Buffer Base Address Sets the start address of the frame buffer where frame data is to be stored. The lower seven bits should be fixed to 000 0000.

(12) YCbCr422 to YCbCr444 Conversion

Data format for the graphics 0 and 1 processes are converted from YCbCr422 to YCbCr444.

This function is not supported for the graphics 2, 3, and OIR processes.

Table 38.109 YCbCr422 to YCbCr444 Conversion

Register Name	Bit Name	Initial Value	Description
GR_FLM6	GR_CNV444_MD	0	Sets the interpolation mode for YCbCr422 to YCbCr444 conversion. *1 0: Hold interpolation 1: Average interpolation

Note 1. This register is not provided for the graphics 2, 3, and OIR processes, for which the YCbCr422 format is not supported.

(13) Bit Extension

When the value of the GR_FORMAT[3:0] bits is 0 to 3, the RGB565, RGB888, αRGB1555, and αRGB4444 formats are converted to the αRGB8888 format. When the value of the GR_FORMAT[3:0] bits is 10, the RGBα5551 format is converted to the RGBα8888 format. The RGBα5551 to RGBα8888 format conversion is omitted because it differs from the αRGB5551 to αRGB8888 format conversion only in the position of α.

- RGB565 to αRGB8888 Format Conversion

  After conversion, α[7:0] is fixed to 255.

  After conversion, R[7:0] = R[4:0] × 263 ÷ 32 (round off to an integer), approximation of #R[4:0] × 255 ÷ 31

  After conversion G[7:0] = G[5:0] × 259 ÷ 64 (round off to an integer), approximation of #G[5:0] × 255 ÷ 63

  After conversion, B[7:0] = B[4:0] × 263 ÷ 32 (round off to an integer), approximation of #B[4:0] × 255 ÷ 31

- RGB888 to αRGB8888 Format Conversion

  After conversion, α[7:0] is fixed to 255.

- αRGB1555 to αRGB8888 Format Conversion

  After conversion, α[7:0] is GR_A1 when α input is 1, and GR_A0 when 0.

After conversion, R[7:0] = R[4:0] × 263 ÷ 32 (round off to an integer), approximation of #R[4:0] × 255 ÷ 31

After conversion, G[7:0] = G[4:0] × 263 ÷ 32 (round off to an integer), approximation of #G[4:0] × 255 ÷ 31

After conversion, B[7:0] = B[4:0] × 263 ÷ 32 (round off to an integer), approximation of #B[4:0] × 255 ÷ 31

- αRGB4444 to αRGB8888 Format Conversion

  After conversion, α[7:0] = α[3:0] × 17

  After conversion, R[7:0] = R[3:0] × 17

  After conversion, G[7:0] = G[3:0] × 17

  After conversion, B[7:0] = B[3:0] × 17

(14) Buffer Underflow Processing

When data read from the frame buffer cannot be completed due to bus-traffic related problems, an underflow interrupt signal is output.

38.6.1.3 Setting Graphics Display Area

The graphics display area is set with the GR_GRC_HS[10:0], GR_GRC_HW[10:0], GR_GRC_VS[10:0], and GR_GRC_VW[10:0] bits based on the rising edges of the Hsync and Vsync signals.

Figure 38.49 shows the graphics display area.

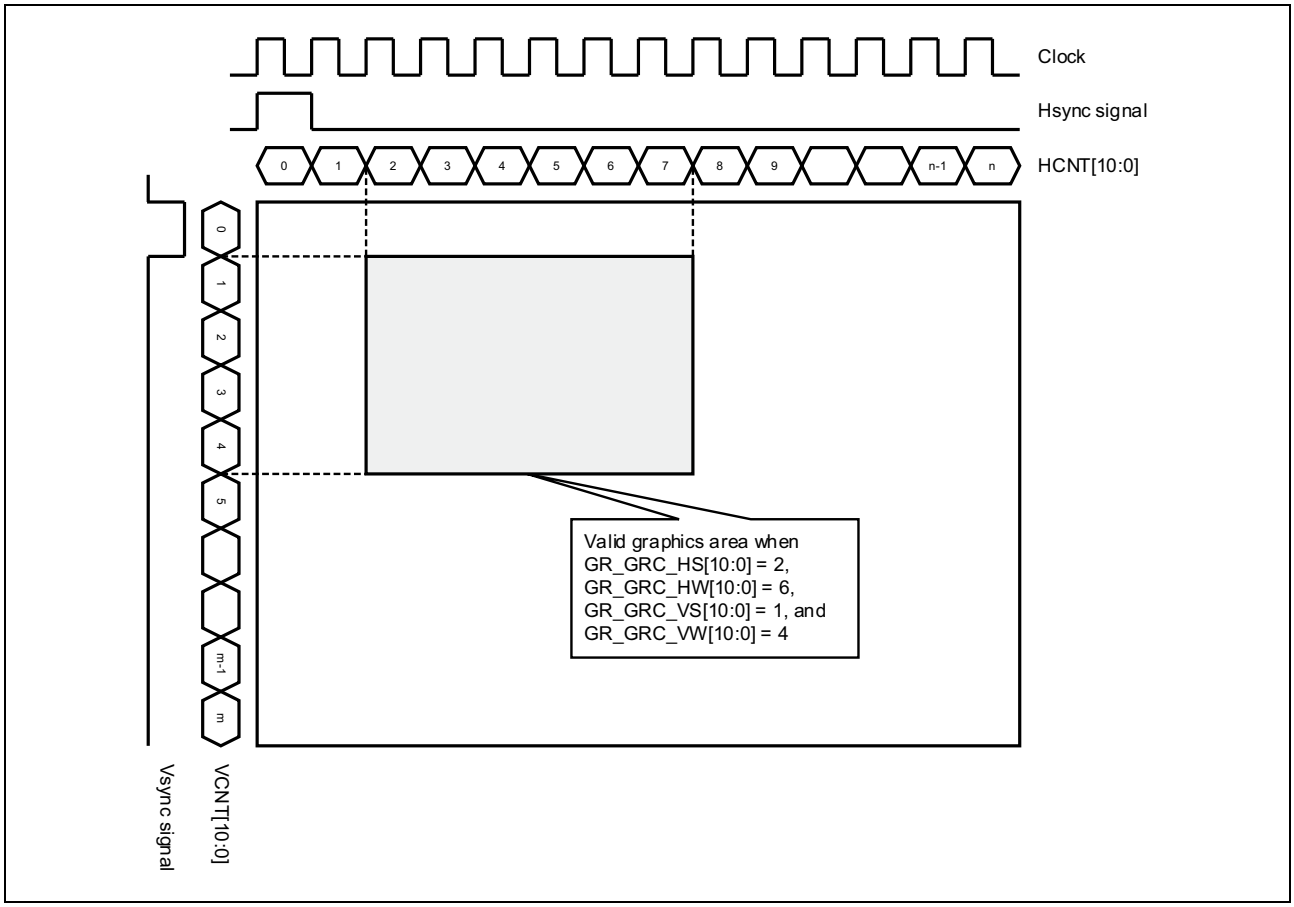


Figure 38.49 Graphics Display Area

The frame line of the graphics area can be displayed by setting the GR_GRC_DISP_ON bit to 1.

Table 38.110 Graphics Image Area Setting

Register Name	Bit Name	Initial Value	Description
GR_AB3	GR_GRC_HS[10:0]	0	Sets the horizontal start position of the graphics image area. Note: Set to 16 or greater clocks and the result of GR_GRC_HS + GR_GRC_HW should be smaller than or equal to 2015 clocks.
GR_AB3	GR_GRC_HW[10:0]	0	Sets the horizontal width of the graphics image area. Note: For displaying an image with 1- or 2-pixel horizontal width, set GR_HW to 2 and GR_GRC_HW to 1 (1-pixel) or 2 (2-pixel).
GR_AB2	GR_GRC_VS[10:0]	0	Sets the vertical start position of the graphics image area. Note: Set to 4 or greater lines and the result of GR_GRC_VS + GR_GRC_VW should be smaller than or equal to 2039 lines.
GR_AB2	GR_GRC_VW[10:0]	0	Sets the vertical width of the graphics image area.
GR_AB1	GR_GRC_DISP_ON	0	Turns on/off frame-line display of the graphics image area. 0: Frame-line display off 1: Frame-line display on

38.6.1.4 Interrupt Generation at Specified Line

An interrupt signal can be generated at the line specified with the GR_LINE[10:0] bits.

Table 38.111 Interrupt Generation at Specified Line

Register Name	Bit Name	Initial Value	Description*1
GR_CLUT_INT	GR_LINE[10:0]	0	Line Interrupt Set When the number of lines matches the value of the GR_LINE bits, an interrupt signal is output. This function is supported for the graphics 3 and OIR processes only. This function supported for the graphics 3 process is enabled even when the graphics 3 process is not used. This function supported for the graphics OIR process is enabled only when the output image generator is enabled.

Note 1. This function is supported for the graphics 3 and OIR processes only; these bits are not supported for the graphics 0, 1, and 2 processes.

38.6.1.5 Formats of Frame Buffer Read Signals and Corresponding Alpha Blending Types

Setting the GR_FORMAT[3:0] bits selects the format of the signal read from the frame buffer.

Table 38.112 shows the signal formats and the corresponding alpha blending types. The priority of the alpha value is: alpha blending in rectangular area > chroma-key processing > alpha blending in pixel units.

Table 38.112 Formats of Frame Buffer Read Signal and Corresponding Alpha Blending Types

GR_FORMAT[3:0]	Signal Format	Alpha Blending in Rectangular Area	RGB-Index Chroma-Key Processing	CLUT-Index Chroma-Key Processing	Alpha Blending in Pixel Units
0	RGB565	Supported	Supported	Not supported	Not supported *2
1	RGB888	Supported	Supported	Not supported	Not supported *2
2	αRGB1555	Supported	Supported *1*3	Not supported	Supported *3
3	αRGB4444	Supported	Supported *1	Not supported	Supported
4	αRGB8888	Supported	Supported	Not supported	Supported
5	CLUT8	Supported	Not supported	Supported	Supported
6	CLUT4	Supported	Not supported	Supported	Supported
7	CLUT1	Supported *4	Not supported	Supported *4	Supported *4
8	YCbCr422	Not supported *5	Not supported *5	Not supported *5	Not supported *5
9	YCbCr444	Not supported *5	Not supported *5	Not supported *5	Not supported *5
10	RGBα5551	Supported	Supported *1*3	Not supported	Supported *3
11	RGBα8888	Supported	Supported	Not supported	Supported

Note 1. When each color component of the RGB signal read from the frame buffer is not 8 bits, it is converted to 8 bits by calculation in RGB-index chroma-key processing. (See Section (13), Bit Extension.)

Note 2. Since α value is 255, the current graphics is always displayed.

Note 3. α value for data read from the frame buffer is specified with one bit. This one-bit signal selects one of the two registers, each of which holds an 8-bit α value.

Note 4. CLUT value for the frame buffer signal is specified with one bit. This one-bit signal selects one of the two registers, each of which holds the α, G, B, and R values (8 bits for each value). The CLUT table is not referenced.

Note 5. YCbCr422 and YCbCr444 are supported for the graphics 0 and 1 processes, but any type of blending and chroma-key processing cannot be used.

38.6.1.6 Display Selection

The GR_DISP_SEL[1:0] bits select the graphics to be displayed from the background color, the lower-layer graphics, the current graphics, and the blended image of the lower-layer graphics and the current graphics. For blending, alpha blending in a rectangular area, multiplication with current alpha at alpha blending in a rectangular area, RGB-index chroma-key processing, CLUT-index chroma-key processing, alpha blending in one-pixel units, or premultiplication at alpha blending in one-pixel units can be selected. Only alpha blending in a rectangular area can be selected in the VIN synthesizer.

Table 38.113 shows the settings for various display types.

Table 38.113 Settings for Various Display Types

GR_DISP_SEL [1:0]	GR_ARC_ON	GR_CK_ON	GR_ARC_MUL	GR_ACALC_MD	Processing for Graphics Area	Processing for the Area outside the Graphics Area
0	—	—	—	—	Background color	Background color
1	—	—	—	—	Lower-layer graphics	Lower-layer graphics
2	—	—	—	—	Current graphics	Background color
3	1	—	0	0	Alpha blending in a rectangular area*1	Lower-layer graphics
3	1	—	0	1	Setting prohibited	
3	1	—	1	0	Multiplication with current alpha at alpha blending in a rectangular area*2	Lower-layer graphics
3	1	—	1	1	Multiplication with current alpha at alpha blending in a rectangular area with alpha premultiplied*2	Lower-layer graphics
3	0	1	—	—	RGB-index or CLUT-index chroma-key processing*3	Lower-layer graphics
3	0	0	—	0	Alpha blending in one-pixel units*2	Lower-layer graphics
3	0	0	—	1	Premultiplication at alpha blending in one-pixel units*2	Lower-layer graphics

Note 1. The alpha blending in a rectangular area is not supported for the graphics 0 process in the scaler and the graphics OIR process. When this processing is selected in the VIN synthesizer, cascaded connection cannot be selected in the scaler.

Note 2. The multiplication with current alpha at alpha blending in a rectangular area and alpha blending function in one-pixel units are supported for the graphics 1, 2, and 3 processes only.

Note 3. The RGB-index or CLUT-index chroma-key processing is not supported for the VIN synthesizer and OIR.

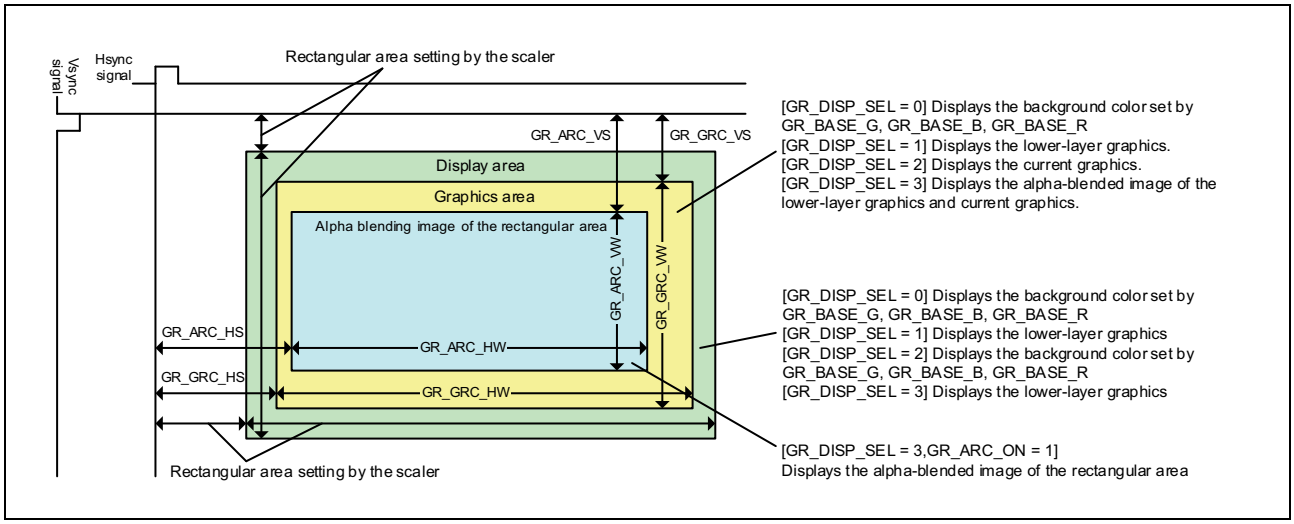


Figure 38.50 Graphic Display Types

Figure 38.51 shows the graphics planes displayed when the GR_DISP_SEL bits are set to 3.

For correspondence between the lower-layer graphics and the current graphics, see **Figure 38.43**.

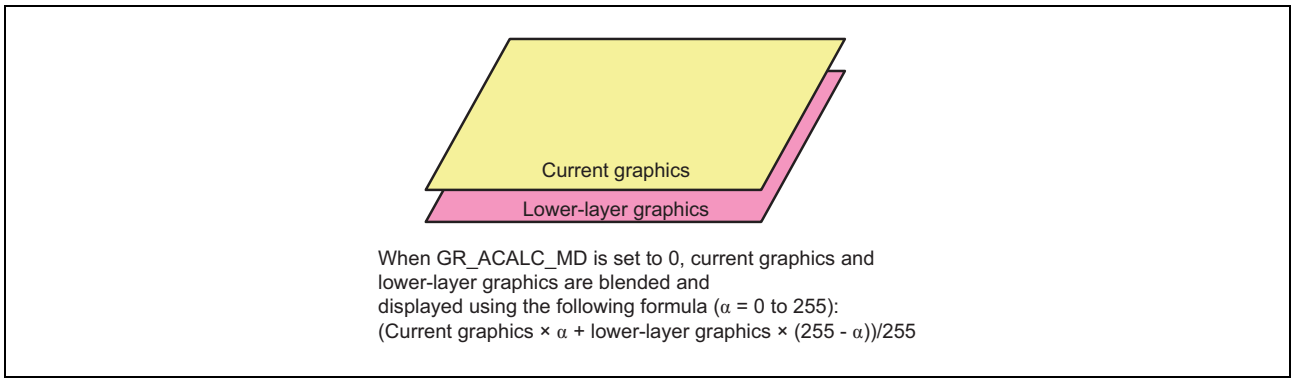


Figure 38.51 Graphics Planes with GR_DISP_SEL Set to 3

Table 38.114 Alpha Blending Setting (1/2)

Register Name	Bit Name	Initial Value	Description
GR_AB1	GR_DISP_SEL [1:0]	0	Selects the graphics display mode. 0: Background color display 1: Lower-layer graphics display 2: Current graphics display 3: Blended display of lower-layer graphics and current graphics*1 Note: The outside of a layer is only transparent if 3 is selected. If 2 is selected, the color outside of the layer is the color contained in GR_BASE register.
GR_AB1	GR_ARC_ON	0	Turns on/off alpha blending in a rectangular area.*2 0: Off 1: On

Table 38.114 Alpha Blending Setting (2/2)

Register Name	Bit Name	Initial Value	Description
GR_AB1	GR_ARC_MUL	0	Turns on/off multiplication processing with current alpha at alpha blending in a rectangular area*3 0: Off 1: On
GR_AB1	GR_ACALC_MD	0	Turns on/off premultiplication processing at alpha blending in one-pixel units*3 0: Off 1: On
GR_AB7	GR_CK_ON	0	Turns on/off CLUT-index/RGB-index chroma-key processing. *4 0: Off 1: On

Note 1. The graphics 0 and OIR processes support the chroma-key processing only. When performing chroma-key processing, set the α value for converting the pixels to be subjected to chroma-key processing, and the α value of the pixels not to be subjected to the chroma-key processing to 255 to display the current graphics only. The VIN synthesizer supports display processing with alpha blending in a rectangular area only.

Note 2. This function is supported only for the graphics 1, 2, and 3 processes and the VIN synthesizer. This bit is not provided for the graphics 0 and OIR processes.

Note 3. This function is supported only for the graphics 1, 2, and 3 processes. This bit is not provided for the graphics 0 and OIR processes and the VIN synthesizer and OIR.

Note 4. This bit is not provided for the VIN synthesizer.

38.6.1.7 Background Color Display Processing

The color set with the GR_BASE_G[7:0], GR_BASE_B[7:0], and GR_BASE_R[7:0] bits is displayed.

G output = GR_BASE_G

B output = GR_BASE_B

R output = GR_BASE_R

Table 38.115 Background Color Setting

Register Name	Bit Name	Initial Value	Description
GR_BASE	GR_BASE_G[7:0]	0	Background color G signal G: 8 bits; unsigned (0 to 255 [LSB])
GR_BASE	GR_BASE_B[7:0]	0	Background color B signal B: 8 bits; unsigned (0 to 255 [LSB])
GR_BASE	GR_BASE_R[7:0]	0	Background color R signal R: 8 bits; unsigned (0 to 255 [LSB])

38.6.1.8 Lower-Layer Graphics Display Processing

The lower-layer graphics are displayed as follows:

G output = G input of lower-layer graphics

B output = B input of lower-layer graphics

R output = R input of lower-layer graphics

38.6.1.9 Current Graphics Display Processing

The current graphics are displayed as follows:

G output = G input of current graphics

B output = B input of current graphics

R output = R input of current graphics

38.6.1.10 Display with Alpha Blending in a Rectangular Area

The rectangular area subjected to alpha blending is set with the GR_ARC_HS[10:0], GR_ARC_HW[10:0], GR_ARC_VS[10:0], and GR_ARC_VW[10:0] bits based on the rising edges of the Hsync and Vsync signals. This function is not supported for the graphics 0 and OIR processes.

Figure 38.52 shows the rectangular area setting for alpha blending.

NOTE

Alpha blending and Chroma-Key features are not supported for Graphic Layer 0 (GR0).

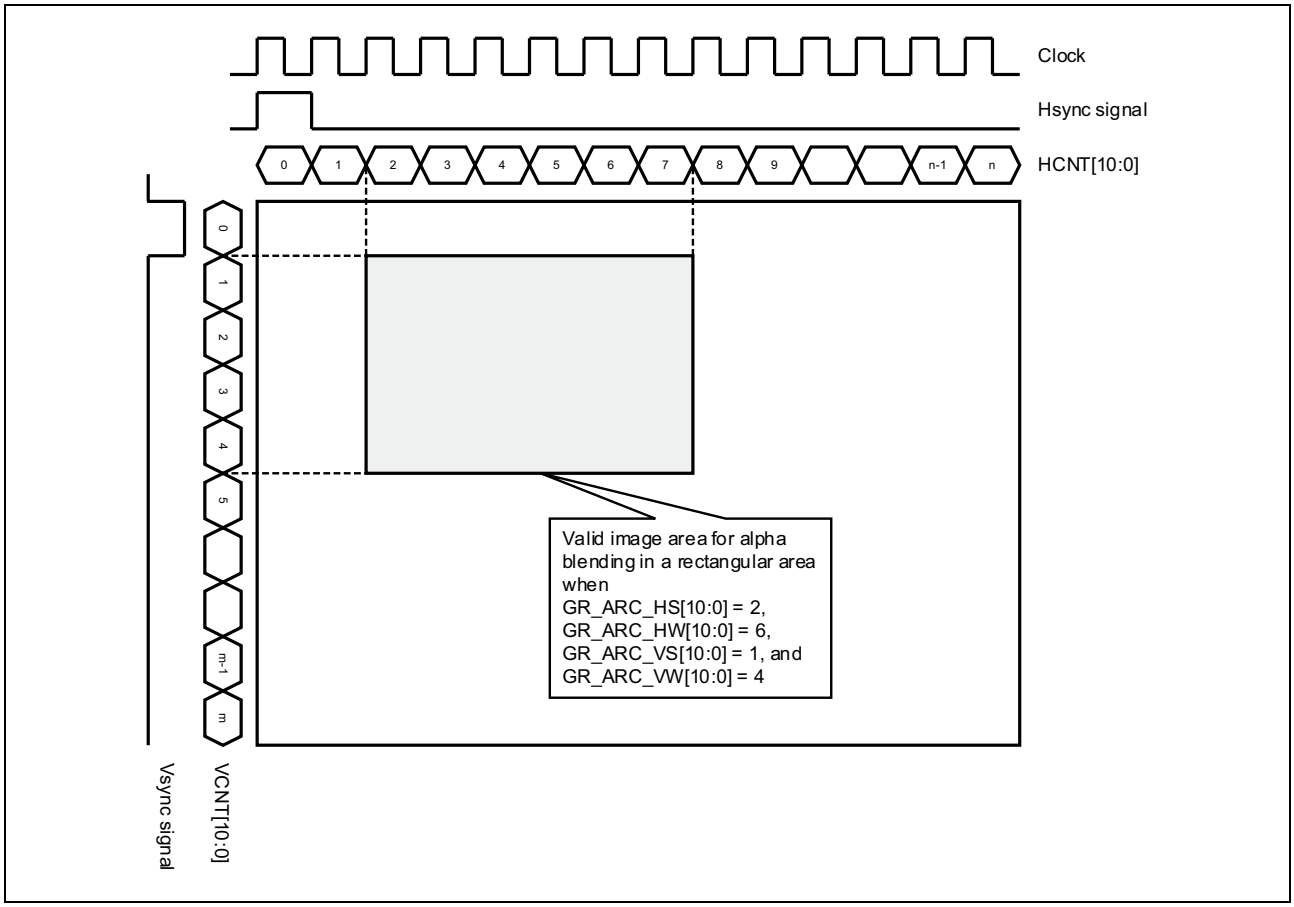


Figure 38.52 Rectangular Area Setting for Alpha Blending

The frame line of graphics area can be displayed by setting the GR_ARC_DISP_ON bit to 1.

Table 38.116 Setting of Rectangular Area for Alpha Blending (1/2)

Register Name	Bit Name	Initial Value	Description
GR_AB5	GR_ARC_HS[10:0]	0	Sets the horizontal start position of the valid image area for alpha blending in a rectangular area.
GR_AB5	GR_ARC_HW[10:0]	0	Sets the horizontal width of the valid image area for alpha blending in a rectangular area.

Table 38.116 Setting of Rectangular Area for Alpha Blending (2/2)

Register Name	Bit Name	Initial Value	Description
GR_AB4	GR_ARC_VS[10:0]	0	Sets the vertical start position of the valid image area for alpha blending in a rectangular area.
GR_AB4	GR_ARC_VW[10:0]	0	Sets the vertical width of the valid image area for alpha blending in a rectangular area.
GR_AB1	GR_ARC_DISP_ON	0	Turns on/off frame-line display of the valid image area for alpha blending in a rectangular area. 0: Frame-line display off 1: Frame-line display on

In alpha blending in a rectangular area, the current graphics are faded in or out by setting the fade-in or fade-out coefficients with the GR_ARC_DEF[7:0], GR_ARC_MODE, GR_ARC_COEF[7:0], and GR_ARC_RATE[7:0] bits.

First, the value of the GR_ARC_DEF[7:0] bits is assigned to the α value.

Then, each time the Vsync signal rises for the number of times set with the GR_ARC_RATE[7:0] bits + 1, the value of the GR_ARC_COEF[7:0] bit is added to or subtracted from the α value.

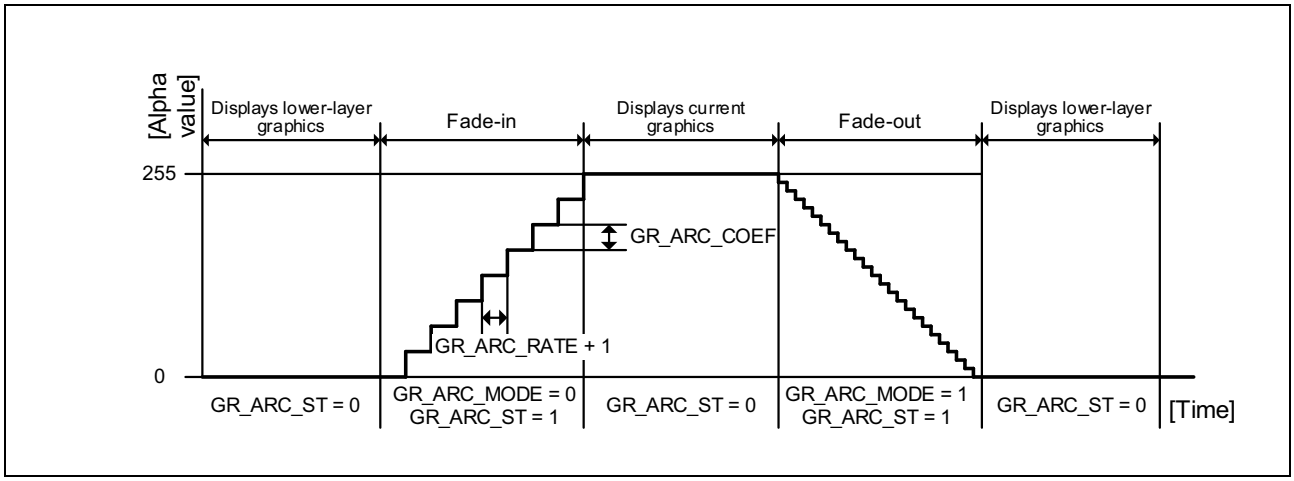


Figure 38.53 Fade In and Fade Out

Table 38.117 Setting for Alpha Blending in a Rectangular Area

Register Name	Bit Name	Initial Value	Description
GR_AB7	GR_ARC_DEF[7:0]	0	Sets the initial alpha value for alpha blending in a rectangular area.
GR_AB6	GR_ARC_MODE	0	Alpha Blending Mode in Rectangular Area 0: Addition 1: Subtraction
GR_AB6	GR_ARC_COEF[7:0]	0	Sets the alpha coefficient for alpha blending in a rectangular area. (0 to 255) [7:0]: Variation (absolute value)
GR_AB6	GR_ARC_RATE[7:0]	0	Sets the value obtained by subtracting 1 from the frame rate for alpha blending in a rectangular area.
GR_MON	GR_ARC_ST	—	Status Flag for Alpha Blending in Rectangular Area 0: Addition or subtraction has been completed. (α value is 0 or 255) 1: Addition or subtraction is in progress.

The values specified with the following expressions are used in the alpha blending calculation described in Section 38.6.1.14, Alpha Blending Calculation.

α value = Fade-in/out coefficient

G value = G input of the current graphics

B value = B input of the current graphics

R value = R input of the current graphics

38.6.1.11 RGB-Index Chroma-Key Processing

The pixels that satisfy all the expressions below are subjected to RGB-index chroma-key processing.

G input of the current graphics = GR_CK_KG

B input of the current graphics = GR_CK_KB

R input of the current graphics = GR_CK_KR

In RGB-index chroma-key processing, the values specified with the following expressions are used in the alpha blending calculation described in Section 38.6.1.14, Alpha Blending Calculation. This function is not supported in the VIN synthesizer.

α value = GR_CK_A

G value = GR_CK_G

B value = GR_CK_B

R value = GR_CK_R

For the pixels that are not subjected to RGB-index chroma-key processing, the values specified with the following expressions are used in the alpha blending calculation described in **Section 38.6.1.14, Alpha Blending Calculation**.

α value = α input of the current graphics

G value = G input of the current graphics

B value = B input of the current graphics

R value = R input of the current graphics

Table 38.118 Setting for RGB-Index Chroma-Key Processing

Register Name	Bit Name	Initial Value	Description
GR_AB8	GR_CK_KG[7:0]	0	G Signal for RGB-Index Chroma-Key Processing G: 8 bits; unsigned (0 to 255 [LSB])
GR_AB8	GR_CK_KB[7:0]	0	B Signal for RGB-Index Chroma-Key Processing B: 8 bits; unsigned (0 to 255 [LSB])
GR_AB8	GR_CK_KR[7:0]	0	R Signal for RGB-Index Chroma-Key Processing R: 8 bits; unsigned (0 to 255 [LSB])
GR_AB9	GR_CK_A[7:0]	0	Replaced Alpha Signal after RGB-Index Chroma-Key Processing*1 α: 8 bits; unsigned (0 to 255 [LSB])
GR_AB9	GR_CK_G[7:0]	0	Replaced G Signal after RGB-Index Chroma-Key Processing G: 8 bits; unsigned (0 to 255 [LSB])
GR_AB9	GR_CK_B[7:0]	0	Replaced B Signal after RGB-Index Chroma-Key Processing B: 8 bits; unsigned (0 to 255 [LSB])
GR_AB9	GR_CK_R[7:0]	0	Replaced R Signal after RGB-Index Chroma-Key Processing R: 8 bits; unsigned (0 to 255 [LSB])

Note 1. To use this function for the graphics 0 and OIR processes, the alpha value should be set to 255.

NOTE

Alpha blending and Chroma-Key features are not supported for Graphic Layer 0 (GR0).

38.6.1.12 CLUT-Index Chroma-Key Processing

The pixels that satisfy the expression below are subjected to CLUT-index chroma-key processing.

CLUT input of the current graphics = GR_CK_KCLUT

In CLUT-index chroma-key processing, the values specified with the following expressions are used in the alpha blending calculation described in Section 38.6.1.14, Alpha Blending Calculation. This function is not supported in the VIN synthesizer and OIR.

α value = GR_CK_A

G value = GR_CK_G

B value = GR_CK_B

R value = GR_CK_R

For the pixels that are not subjected to CLUT-index chroma-key processing, the values specified with the following expressions are used in the alpha blending calculation described in **Section 38.6.1.14, Alpha Blending Calculation**.

α value = α input of the current graphics

G value = G input of the current graphics

B value = B input of the current graphics

R value = R input of the current graphics

NOTE

Alpha blending and Chroma-Key features are not supported for Graphic Layer 0 (GR0).

38.6.1.13 Display with Alpha Blending in One-Pixel Units

In the alpha blending in one-pixel units, the values specified with the following expressions are used in the alpha blending calculation described in Section 38.6.1.14, Alpha Blending Calculation. This function is not supported in the graphics 0 and OIR processes and the VIN synthesizer.

α value = α input of the current graphics

G value = G input of the current graphics

B value = B input of the current graphics

R value = R input of the current graphics

38.6.1.14 Alpha Blending Calculation

Alpha blending of two input signals is performed using the α value as described below (rounded up if the result includes a decimal fraction).

[GR_ACALC_MD = 0]

G output = (G value × α value + G input of the lower-layer graphics × (255 − α value)) ÷ 256

B output = (B value × α value + B input of the lower-layer graphics × (255 − α value)) ÷ 256

R output = (R value × α value + R input of the lower-layer graphics × (255 − α value)) ÷ 256

[GR_ACALC_MD = 1 (premultiplication)]

G output = (G value + G input of the lower-layer graphics × (255 − α value)) ÷ 256

B output = (B value + B input of the lower-layer graphics × (255 − α value)) ÷ 256

R output = (R value + R input of the lower-layer graphics × (255 − α value)) ÷ 256

38.6.1.15 CLUT Table

When the signal format is CLUT8 or CLUT4, the format is converted to αRGB8888 based on the CLUT table. When the format is CLUT1, it is converted to αRGB8888 based on the register value.

Figure 38.54 shows data arrangement in the CLUT table.

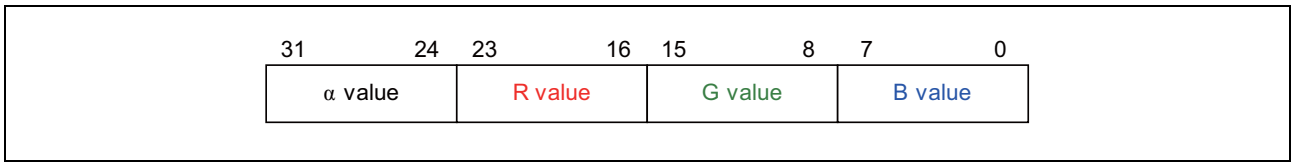


Figure 38.54 Data Arrangement in CLUT Table

For the addresses of the CLUT tables, see the register configuration described in **Section 38.6.2, Register Descriptions**.

Two CLUT tables (CLUT table 0, CLUT table 1) on the different planes are allocated to the same address and one of the tables is selected with the GR_CLT_SEL bit. This allows rewriting one CLUT table when this module refers to the other CLUT table.

Table 38.119 CLUT Table Selection

Register Name	Bit Name	Initial Value	Description
GR_CLUT	GR_CLT_SEL	0	CLUT Table Select Signal 0: Selects CLUT table 0. The format is converted to αRGB8888 based on the CLUT table 0. CLUT table 1 can be read from or written to by the CPU. 1: Selects CLUT table 1. The format is converted to αRGB8888 based on the CLUT table 1. CLUT table 0 can be read from or written to by the CPU.

NOTE

Before switching CLUT table by GR_CLT_SEL, CPU must wait until the read value from the last written address is same as the last written value.

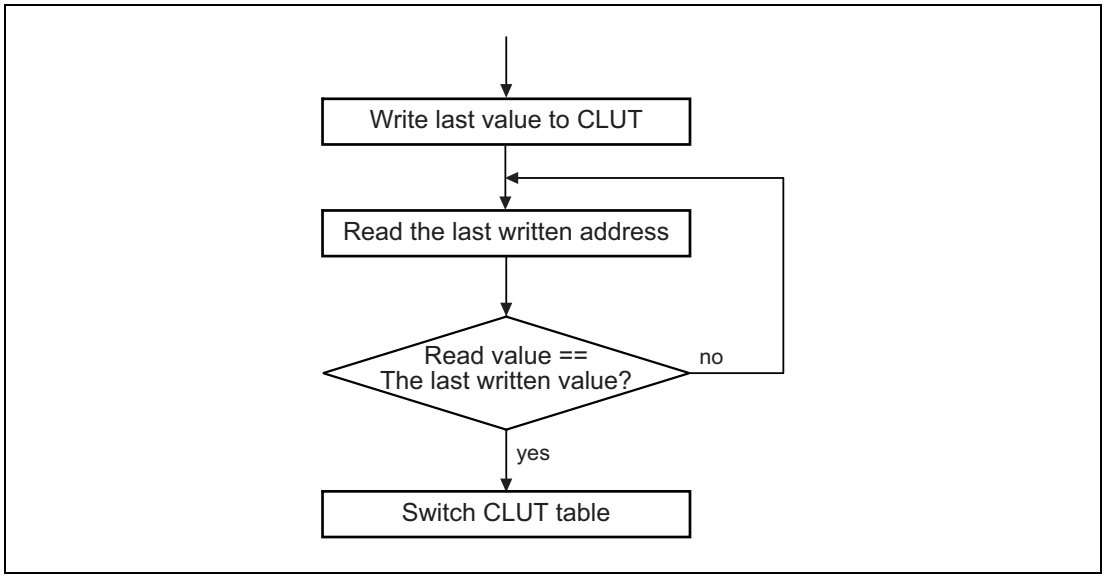


Figure 38.55 CLUT switching

38.6.1.16 Multiplication Processing with Current Alpha at Alpha Blending in Rectangular Area

In multiplication processing with current alpha at alpha blending in a rectangular area, the values specified with the following expressions are used in the alpha blending calculation described in Section 38.6.1.14, Alpha Blending Calculation.

[GR_ARC_MUL = 0]

α value = Fade-in/out coefficient

G value = G input of the current graphics

B value = B input of the current graphics

R value = R input of the current graphics

[GR_ARC_MUL = 1]

α value = Fade-in/out coefficient x α input of current graphics

G value = G input of the current graphics

B value = B input of the current graphics

R value = R input of the current graphics

38.6.1.17 Selection of Lower-Layer/Current Graphics in VIN Synthesizer

Graphics 0 and 1 are allocated to the lower-layer/current graphics in the VIN synthesizer, respectively.

Table 38.120 Selection of Lower-Layer Plane in Scaler

Register Name	Bit Name	Initial Value	Description
GR_VIN_AB1	GR_VIN_SCL_UND_SEL	0	Specifies lower-layer plane in the scaler. 0: Selects graphics 0 as lower-layer graphics and graphics 1 as current graphics. 1: Selects graphics 1 as lower-layer graphics and graphics 0 as current graphics.

38.6.2 Register Descriptions

Table 38.121 to **Table 38.128** show the register configuration.

- Symbols used in Register Description:

  Initial value: Register value after a reset

  —: Undefined value

  R/W: Readable/writable. The written value can be read.

  R/WC0: Readable/writable. Writing 0 initializes the bit. Writing 1 is ignored.

  R/WC1: Readable/writable. Writing 1 initializes the bit. Writing 0 is ignored.

  R: Read-only. The write value should always be 0.

  —/W: Write-only. The read value is undefined.

Table 38.121 shows the register configuration for the graphics 2 process of channel 0.

Table 38.122 shows the register configuration for the graphics 3 process of channel 0.

Table 38.123 shows the CLUT table configuration of channel 0.

Table 38.124 shows the register configuration for the VIN synthesizer of channel 0.

Table 38.125 shows the register configuration for the graphics 2 process of channel 1.

Table 38.126 shows the register configuration for the graphics 3 process of channel 1.

Table 38.127 shows the CLUT table configuration of channel 1.

Table 38.128 shows the register configuration for the VIN synthesizer of channel 1.

The register configuration for the graphics 0 and 1 processes is described in Section 38.4, Scaler.

Table 38.121 Register Configuration of the Image Synthesizer (Graphics 2 Process) (Channel 0) (1/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Graphics 2 register update control register	GR2_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1700	32
VDCE0	Frame buffer read control register (Graphics 2)	GR2_FLM_RD	R/W	H'0000 0000	<VDCE0_base> + 1704	32
VDCE0	Frame buffer control register 1 (Graphics 2)	GR2_FLM1	R/W	H'0000 0000	<VDCE0_base> + 1708	32
VDCE0	Frame buffer control register 2 (Graphics 2)	GR2_FLM2	R/W	H'0000 0000	<VDCE0_base> + 170C	32
VDCE0	Frame buffer control register 3 (Graphics 2)	GR2_FLM3	R/W	H'0000 0000	<VDCE0_base> + 1710	32
VDCE0	Frame buffer control register 4 (Graphics 2)	GR2_FLM4	R/W	H'0000 0000	<VDCE0_base> + 1714	32
VDCE0	Frame buffer control register 5 (Graphics 2)	GR2_FLM5	R/W	H'0000 03FF	<VDCE0_base> + 1718	32
VDCE0	Frame buffer control register 6 (Graphics 2)	GR2_FLM6	R/W	H'0000 0000	<VDCE0_base> + 171C	32
VDCE0	Alpha blending control register 1 (Graphics 2)	GR2_AB1	R/W	H'0000 0000	<VDCE0_base> + 1720	32
VDCE0	Alpha blending control register 2 (Graphics 2)	GR2_AB2	R/W	H'0000 0000	<VDCE0_base> + 1724	32

Table 38.121 Register Configuration of the Image Synthesizer (Graphics 2 Process) (Channel 0) (2/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Alpha blending control register 3 (Graphics 2)	GR2_AB3	R/W	H'0000 0000	<VDCE0_base> + 1728	32
VDCE0	Alpha blending control register 4 (Graphics 2)	GR2_AB4	R/W	H'0000 0000	<VDCE0_base> + 172C	32
VDCE0	Alpha blending control register 5 (Graphics 2)	GR2_AB5	R/W	H'0000 0000	<VDCE0_base> + 1730	32
VDCE0	Alpha blending control register 6 (Graphics 2)	GR2_AB6	R/W	H'0000 0000	<VDCE0_base> + 1734	32
VDCE0	Alpha blending control register 7 (Graphics 2)	GR2_AB7	R/W	H'00FF 0000	<VDCE0_base> + 1738	32
VDCE0	Alpha blending control register 8 (Graphics 2)	GR2_AB8	R/W	H'0000 0000	<VDCE0_base> + 173C	32
VDCE0	Alpha blending control register 9 (Graphics 2)	GR2_AB9	R/W	H'0000 0000	<VDCE0_base> + 1740	32
VDCE0	Alpha blending control register 10 (Graphics 2)	GR2_AB10	R/W	H'0000 0000	<VDCE0_base> + 1744	32
VDCE0	Alpha blending control register 11 (Graphics 2)	GR2_AB11	R/W	H'0000 0000	<VDCE0_base> + 1748	32
VDCE0	Background color control register (Graphics 2)	GR2_BASE	R/W	H'0000 0000	<VDCE0_base> + 174C	32
VDCE0	CLUT table control register (Graphics 2)	GR2_CLUT	R/W	H'0000 0000	<VDCE0_base> + 1750	32
VDCE0	Status monitor register (Graphics 2)	GR2_MON	R	H'0000 0000	<VDCE0_base> + 1754	32

Table 38.122 Register Configuration of the Image Synthesizer (Graphics 3 Process) (Channel 0) (1/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Graphics 3 register update control register	GR3_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1780	32
VDCE0	Frame buffer read control register (Graphics 3)	GR3_FLM_RD	R/W	H'0000 0000	<VDCE0_base> + 1784	32
VDCE0	Frame buffer control register 1 (Graphics 3)	GR3_FLM1	R/W	H'0000 0000	<VDCE0_base> + 1788	32
VDCE0	Frame buffer control register 2 (Graphics 3)	GR3_FLM2	R/W	H'0000 0000	<VDCE0_base> + 178C	32
VDCE0	Frame buffer control register 3 (Graphics 3)	GR3_FLM3	R/W	H'0000 0000	<VDCE0_base> + 1790	32
VDCE0	Frame buffer control register 4 (Graphics 3)	GR3_FLM4	R/W	H'0000 0000	<VDCE0_base> + 1794	32
VDCE0	Frame buffer control register 5 (Graphics 3)	GR3_FLM5	R/W	H'0000 03FF	<VDCE0_base> + 1798	32
VDCE0	Frame buffer control register 6 (Graphics 3)	GR3_FLM6	R/W	H'0000 0000	<VDCE0_base> + 179C	32
VDCE0	Alpha blending control register 1 (Graphics 3)	GR3_AB1	R/W	H'0000 0000	<VDCE0_base> + 17A0	32
VDCE0	Alpha blending control register 2 (Graphics 3)	GR3_AB2	R/W	H'0000 0000	<VDCE0_base> + 17A4	32
VDCE0	Alpha blending control register 3 (Graphics 3)	GR3_AB3	R/W	H'0000 0000	<VDCE0_base> + 17A8	32

Table 38.122 Register Configuration of the Image Synthesizer (Graphics 3 Process) (Channel 0) (2/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Alpha blending control register 4 (Graphics 3)	GR3_AB4	R/W	H'0000 0000	<VDCE0_base> + 17AC	32
VDCE0	Alpha blending control register 5 (Graphics 3)	GR3_AB5	R/W	H'0000 0000	<VDCE0_base> + 17B0	32
VDCE0	Alpha blending control register 6 (Graphics 3)	GR3_AB6	R/W	H'0000 0000	<VDCE0_base> + 17B4	32
VDCE0	Alpha blending control register 7 (Graphics 3)	GR3_AB7	R/W	H'00FF 0000	<VDCE0_base> + 17B8	32
VDCE0	Alpha blending control register 8 (Graphics 3)	GR3_AB8	R/W	H'0000 0000	<VDCE0_base> + 17BC	32
VDCE0	Alpha blending control register 9 (Graphics 3)	GR3_AB9	R/W	H'0000 0000	<VDCE0_base> + 17C0	32
VDCE0	Alpha blending control register 10 (Graphics 3)	GR3_AB10	R/W	H'0000 0000	<VDCE0_base> + 17C4	32
VDCE0	Alpha blending control register 11 (Graphics 3)	GR3_AB11	R/W	H'0000 0000	<VDCE0_base> + 17C8	32
VDCE0	Background color control register (Graphics 3)	GR3_BASE	R/W	H'0000 0000	<VDCE0_base> + 17CC	32
VDCE0	CLUT table and interrupt control register (Graphics 3)	GR3_CLUT_INT	R/W	H'0000 0000	<VDCE0_base> + 17D0	32
VDCE0	Status monitor register (Graphics 3)	GR3_MON	R	H'0000 0000	<VDCE0_base> + 17D4	32

Table 38.123 CLUT Table Configuration (Channel 0)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Graphics 0 CLUT table	GR0_CLUT_TBL	R/W	—	<VDCE0_base> + 0000 to <VDCE0_base> + 03FF	32
VDCE0	Graphics 1 CLUT table	GR1_CLUT_TBL	R/W	—	<VDCE0_base> + 0400 to <VDCE0_base> + 07FF	32
VDCE0	Graphics 2 CLUT table	GR2_CLUT_TBL	R/W	—	<VDCE0_base> + 0800 to <VDCE0_base> + 0BFF	32
VDCE0	Graphics 3 CLUT table	GR3_CLUT_TBL	R/W	—	<VDCE0_base> + 0C00 to <VDCE0_base> + 0FFF	32

Table 38.124 Register Configuration of the VIN Synthesizer (Channel 0) (1/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	VIN synthesizer register update control register	GR_VIN_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1E00	32
VDCE0	Alpha blending control register 1 (VIN synthesizer)	GR_VIN _AB1	R/W	H'0000 0000	<VDCE0_base> + 1E20	32
VDCE0	Alpha blending control register 2 (VIN synthesizer)	GR_VIN _AB2	R/W	H'0000 0000	<VDCE0_base> + 1E24	32

Table 38.124 Register Configuration of the VIN Synthesizer (Channel 0) (2/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Alpha blending control register 3 (VIN synthesizer)	GR_VIN _AB3	R/W	H'0000 0000	<VDCE0_base> + 1E28	32
VDCE0	Alpha blending control register 4 (VIN synthesizer)	GR_VIN _AB4	R/W	H'0000 0000	<VDCE0_base> + 1E2C	32
VDCE0	Alpha blending control register 5 (VIN synthesizer)	GR_VIN _AB5	R/W	H'0000 0000	<VDCE0_base> + 1E30	32
VDCE0	Alpha blending control register 6 (VIN synthesizer)	GR_VIN _AB6	R/W	H'0000 0000	<VDCE0_base> + 1E34	32
VDCE0	Alpha blending control register 7 (VIN synthesizer)	GR_VIN _AB7	R/W	H'00FF 0000	<VDCE0_base> + 1E38	32
VDCE0	Background color control register (VIN synthesizer)	GR_VIN_BASE	R/W	H'0000 0000	<VDCE0_base> + 1E4C	32
VDCE0	Status monitor register (VIN synthesizer)	GR_VIN_MON	R	H'0000 0000	<VDCE0_base> + 1E54	32

Table 38.125 Register Configuration of the Image Synthesizer (Graphics 2 Process) (Channel 1) (1/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Graphics 2 register update control register	GR2_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1700	32
VDCE1	Frame buffer read control register (Graphics 2)	GR2_FLM_RD	R/W	H'0000 0000	<VDCE1_base> + 1704	32
VDCE1	Frame buffer control register 1 (Graphics 2)	GR2_FLM1	R/W	H'0000 0000	<VDCE1_base> + 1708	32
VDCE1	Frame buffer control register 2 (Graphics 2)	GR2_FLM2	R/W	H'0000 0000	<VDCE1_base> + 170C	32
VDCE1	Frame buffer control register 3 (Graphics 2)	GR2_FLM3	R/W	H'0000 0000	<VDCE1_base> + 1710	32
VDCE1	Frame buffer control register 4 (Graphics 2)	GR2_FLM4	R/W	H'0000 0000	<VDCE1_base> + 1714	32
VDCE1	Frame buffer control register 5 (Graphics 2)	GR2_FLM5	R/W	H'0000 03FF	<VDCE1_base> + 1718	32
VDCE1	Frame buffer control register 6 (Graphics 2)	GR2_FLM6	R/W	H'0000 0000	<VDCE1_base> + 171C	32
VDCE1	Alpha blending control register 1 (Graphics 2)	GR2_AB1	R/W	H'0000 0000	<VDCE1_base> + 1720	32
VDCE1	Alpha blending control register 2 (Graphics 2)	GR2_AB2	R/W	H'0000 0000	<VDCE1_base> + 1724	32
VDCE1	Alpha blending control register 3 (Graphics 2)	GR2_AB3	R/W	H'0000 0000	<VDCE1_base> + 1728	32
VDCE1	Alpha blending control register 4 (Graphics 2)	GR2_AB4	R/W	H'0000 0000	<VDCE1_base> + 172C	32
VDCE1	Alpha blending control register 5 (Graphics 2)	GR2_AB5	R/W	H'0000 0000	<VDCE1_base> + 1730	32
VDCE1	Alpha blending control register 6 (Graphics 2)	GR2_AB6	R/W	H'0000 0000	<VDCE1_base> + 1734	32
VDCE1	Alpha blending control register 7 (Graphics 2)	GR2_AB7	R/W	H'00FF 0000	<VDCE1_base> + 1738	32
VDCE1	Alpha blending control register 8 (Graphics 2)	GR2_AB8	R/W	H'0000 0000	<VDCE1_base> + 173C	32

Table 38.125 Register Configuration of the Image Synthesizer (Graphics 2 Process) (Channel 1) (2/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Alpha blending control register 9 (Graphics 2)	GR2_AB9	R/W	H'0000 0000	<VDCE1_base> + 1740	32
VDCE1	Alpha blending control register 10 (Graphics 2)	GR2_AB10	R/W	H'0000 0000	<VDCE1_base> + 1744	32
VDCE1	Alpha blending control register 11 (Graphics 2)	GR2_AB11	R/W	H'0000 0000	<VDCE1_base> + 1748	32
VDCE1	Background color control register (Graphics 2)	GR2_BASE	R/W	H'0000 0000	<VDCE1_base> + 174C	32
VDCE1	CLUT table control register (Graphics 2)	GR2_CLUT	R/W	H'0000 0000	<VDCE1_base> + 1750	32
VDCE1	Status monitor register (Graphics 2)	GR2_MON	R	H'0000 0000	<VDCE1_base> + 1754	32

Table 38.126 Register Configuration of the Image Synthesizer (Graphics 3 Process) (Channel 1) (1/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Graphics 3 register update control register	GR3_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1780	32
VDCE1	Frame buffer read control register (Graphics 3)	GR3_FLM_RD	R/W	H'0000 0000	<VDCE1_base> + 1784	32
VDCE1	Frame buffer control register 1 (Graphics 3)	GR3_FLM1	R/W	H'0000 0000	<VDCE1_base> + 1788	32
VDCE1	Frame buffer control register 2 (Graphics 3)	GR3_FLM2	R/W	H'0000 0000	<VDCE1_base> + 178C	32
VDCE1	Frame buffer control register 3 (Graphics 3)	GR3_FLM3	R/W	H'0000 0000	<VDCE1_base> + 1790	32
VDCE1	Frame buffer control register 4 (Graphics 3)	GR3_FLM4	R/W	H'0000 0000	<VDCE1_base> + 1794	32
VDCE1	Frame buffer control register 5 (Graphics 3)	GR3_FLM5	R/W	H'0000 03FF	<VDCE1_base> + 1798	32
VDCE1	Frame buffer control register 6 (Graphics 3)	GR3_FLM6	R/W	H'0000 0000	<VDCE1_base> + 179C	32
VDCE1	Alpha blending control register 1 (Graphics 3)	GR3_AB1	R/W	H'0000 0000	<VDCE1_base> + 17A0	32
VDCE1	Alpha blending control register 2 (Graphics 3)	GR3_AB2	R/W	H'0000 0000	<VDCE1_base> + 17A4	32
VDCE1	Alpha blending control register 3 (Graphics 3)	GR3_AB3	R/W	H'0000 0000	<VDCE1_base> + 17A8	32
VDCE1	Alpha blending control register 4 (Graphics 3)	GR3_AB4	R/W	H'0000 0000	<VDCE1_base> + 17AC	32
VDCE1	Alpha blending control register 5 (Graphics 3)	GR3_AB5	R/W	H'0000 0000	<VDCE1_base> + 17B0	32
VDCE1	Alpha blending control register 6 (Graphics 3)	GR3_AB6	R/W	H'0000 0000	<VDCE1_base> + 17B4	32
VDCE1	Alpha blending control register 7 (Graphics 3)	GR3_AB7	R/W	H'00FF 0000	<VDCE1_base> + 17B8	32
VDCE1	Alpha blending control register 8 (Graphics 3)	GR3_AB8	R/W	H'0000 0000	<VDCE1_base> + 17BC	32
VDCE1	Alpha blending control register 9 (Graphics 3)	GR3_AB9	R/W	H'0000 0000	<VDCE1_base> + 17C0	32

Table 38.126 Register Configuration of the Image Synthesizer (Graphics 3 Process) (Channel 1) (2/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Alpha blending control register 10 (Graphics 3)	GR3_AB10	R/W	H'0000 0000	<VDCE1_base> + 17C4	32
VDCE1	Alpha blending control register 11 (Graphics 3)	GR3_AB11	R/W	H'0000 0000	<VDCE1_base> + 17C8	32
VDCE1	Background color control register (Graphics 3)	GR3_BASE	R/W	H'0000 0000	<VDCE1_base> + 17CC	32
VDCE1	CLUT table and interrupt control register (Graphics 3)	GR3_CLUT_INT	R/W	H'0000 0000	<VDCE1_base> + 17D0	32
VDCE1	Status monitor register (Graphics 3)	GR3_MON	R	H'0000 0000	<VDCE1_base> + 17D4	32

Table 38.127 CLUT Table Configuration (Channel 1)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Graphics 0 CLUT table	GR0_CLUT_TBL	R/W	—	<VDCE1_base> + 0000 to <VDCE1_base> + 03FF	32
VDCE1	Graphics 1 CLUT table	GR1_CLUT_TBL	R/W	—	<VDCE1_base> + 0400 to <VDCE1_base> + 07FF	32
VDCE1	Graphics 2 CLUT table	GR2_CLUT_TBL	R/W	—	<VDCE1_base> + 0800 to <VDCE1_base> + 0BFF	32
VDCE1	Graphics 3 CLUT table	GR3_CLUT_TBL	R/W	—	<VDCE1_base> + 0C00 to <VDCE1_base> + 0FFF	32

Table 38.128 Register Configuration of the VIN Synthesizer (Channel 1) (1/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	VIN synthesizer register update control register	GR_VIN_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1E00	32
VDCE1	Alpha blending control register 1 (VIN synthesizer)	GR_VIN_AB1	R/W	H'0000 0000	<VDCE1_base> + 1E20	32
VDCE1	Alpha blending control register 2 (VIN synthesizer)	GR_VIN_AB2	R/W	H'0000 0000	<VDCE1_base> + 1E24	32
VDCE1	Alpha blending control register 3 (VIN synthesizer)	GR_VIN_AB3	R/W	H'0000 0000	<VDCE1_base> + 1E28	32
VDCE1	Alpha blending control register 4 (VIN synthesizer)	GR_VIN_AB4	R/W	H'0000 0000	<VDCE1_base> + 1E2C	32
VDCE1	Alpha blending control register 5 (VIN synthesizer)	GR_VIN_AB5	R/W	H'0000 0000	<VDCE1_base> + 1E30	32
VDCE1	Alpha blending control register 6 (VIN synthesizer)	GR_VIN_AB6	R/W	H'0000 0000	<VDCE1_base> + 1E34	32

Table 38.128 Register Configuration of the VIN Synthesizer (Channel 1) (2/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Alpha blending control register 7 (VIN synthesizer)	GR_VIN_AB7	R/W	H'00FF 0000	<VDCE1_base> + 1E38	32
VDCE1	Background color control register (VIN synthesizer)	GR_VIN_BASE	R/W	H'0000 0000	<VDCE1_base> + 1E4C	32
VDCE1	Status monitor register (VIN synthesizer)	GR_VIN_MON	R	H'0000 0000	<VDCE1_base> + 1E54	32

NOTE

Register access sizes other than defined in the table above are not supported.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

38.6.2.1 Graphics 2 Register Update Control Register (GR2_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GR2_UPDATE	—	—	—	GR2_P_VEN	—	—	—	GR2_IBUS_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR2_UPDATE	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR2_P_VEN	0	R/WC1	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR2_IBUS_VEN	0	R/WC1	Frame Buffer Read Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.6.2.2 Frame Buffer Read Control Register (Graphics 2) (GR2_FLM_RD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR2_R_ENB
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR2_R_ENB	0	R/W	Frame Buffer Read Enable 0: Frame buffer reading is disabled. 1: Frame buffer reading is enabled.

Note: This register is updated when GR2_IBUS_VEN in GR2_UPDATE is 1.

38.6.2.3 Frame Buffer Control Register 1 (Graphics 2) (GR2_FLM1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR2_LN_OFF_DIR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR2_FLM_SEL[1:0]		—	—	—	—	—	—	—	Caution *
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

CAUTION

The initial value “0” of bit 0 must be changed to “1”.

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR2_LN_OFF_DIR	0	R/W	Selects the line offset address direction of the frame buffer. 0: Increments the address by the line offset address. 1: Decrements the address by the line offset address.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	GR2_FLM_SEL[1:0]	0	R/W	Selects a frame buffer address setting signal. 0: Selects frame 0. 1: Selects register GR2_FLM_NUM[9:0]. 2: Selects frame 0. 3: Setting prohibited
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	Bit0	0	R/W	**CAUTION** The initial value “0” of this bit must be changed to “1”.

Note: GR2_LN_OFF_DIR and GR2_FLM_SEL[1:0] are updated when GR2_IBUS_VEN in GR2_UPDATE is 1.

38.6.2.4 Frame Buffer Control Register 2 (Graphics 2) (GR2_FLM2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR2_BASE[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_BASE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	GR2_BASE [31:0]	0	R/W	Frame Buffer Base Address (upper) Sets the start address of the frame buffer where frame data is to be stored. The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when GR2_IBUS_VEN and GR2_P_VEN in GR2_UPDATE are 1.

NOTE

Check restriction concerning frame buffer addressing in Section 38.6.1.2, Graphics Data Read Control.

38.6.2.5 Frame Buffer Control Register 3 (Graphics 2) (GR2_FLM3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	GR2_LN_OFF[14:0]														
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR2_FLM_NUM[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 16	GR2_LN_OFF[14:0]	0	R/W	Frame Buffer Line Offset Address Sets the line offset address for calculating the start address of each line. Line 0: GR2_BASE[31:0] Line 1: GR2_BASE[31:0] + GR2_LN_OFF[14:0] × 1 : Line n: GR2_BASE[31:0] + GR2_LN_OFF[14:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GR2_FLM_NUM[9:0]	0	R/W	Frame Number of Frame Buffer Manually set the frame number when GR2_FLM_SEL[1:0] = 1.

Note: This register is updated when GR2_IBUS_VEN in GR2_UPDATE is 1.

38.6.2.6 Frame Buffer Control Register 4 (Graphics 2) (GR2_FLM4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	GR2_FLM_OFF[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_FLM_OFF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	GR2_FLM_OFF[22:0]	0	R/W	Frame Buffer Frame Offset Address (upper) Specifies the frame offset address used for calculating the start address of each frame buffer when more than one buffer is used. Buffer 0: GR2_BASE[31:0] Buffer 1: GR2_BASE[31:0] + GR2_FLM_OFF[22:0] × 1 : Buffer n: GR2_BASE[31:0] + GR2_FLM_OFF[22:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when GR2_IBUS_VEN in GR2_UPDATE is 1.

38.6.2.7 Frame Buffer Control Register 5 (Graphics 2) (GR2_FLM5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR2_FLM_LNUM[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR2_FLM_LOOP[10:0]										
Initial value:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR2_FLM_ LNUM[10:0]	0	R/W	Sets number of lines in a frame The number of lines is (GR2_FLM_LNUM[10:0] + 1).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR2_FLM_ LOOP[10:0]	1023	R/W	Number of lines when reading the addresses repeatedly by returning to the start address after reaching the end address. The number of lines is (GR2_FLM_LOOP[10:0] + 1).

Note: This register is updated when GR2_IBUS_VEN in GR2_UPDATE is 1.

38.6.2.8 Frame Buffer Control Register 6 (Graphics 2) (GR2_FLM6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR2_FORMAT[3:0]				—	GR2_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	GR2_RDSWA[2:0]			—	—	—	—	GR2_STA_POS[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	GR2_ FORMAT [3:0]	0	R/W	Sets the format of the frame buffer read signal. 0: RGB565 1: RGB888 2: αRGB1555 3: αRGB4444 4: αRGB8888 5: CLUT8 6: CLUT4 7: CLUT1 8: Setting prohibited 9: Setting prohibited 10: RGBα5551 11: RGBα8888 12 to 15: Setting prohibited
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 16	GR2_HW [10:0]	0	R/W	Sets the width of the horizontal enable period. The width is (GR2_HW[10:0] + 1) pixels. Note: Set to 2 or greater.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12 to 10	GR2_ RDSWA [2:0]	0	R/W	Sets 8-, 16-, and 32-bit swap. These three bits specify the method for swapping the bits of frame buffer read data as follows. Bit 0 0: 8 bits are not swapped. 1: 8 bits are swapped. Bit 1 0: 16 bits are not swapped. 1: 16 bits are swapped. Bit 2 0: 32 bits are not swapped. 1: 32 bits are swapped. When eight bits are put together, they are swapped as follows. Each of (1) to (8) indicates eight-bit data. 000: (1) (2) (3) (4) (5) (6) (7) (8) [No swap] 001: (2) (1) (4) (3) (6) (5) (8) (7) [8-bit swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [16-bit swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [16-bit swap + 8-bit swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [32-bit swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [32-bit swap + 8-bit swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [32-bit swap + 16-bit swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [32-bit swap + 16-bit swap + 8-bit swap]
9 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	GR2_ STA_POS [5:0]	0	R/W	Sets the amount of pixels to be skipped through. Specifically pixels amount equal to the amount indicated by GR2_STA_POS[5:0] is skipped from the start of the line.

Note: GR2_STA_POS[5:0] is updated when GR2_P_VEN in GR2_UPDATE is 1. GR2_RDSWA[2:0] is updated when GR2_UPDATE in GR2_UPDATE is 1.
GR2_FORMAT[3:0] and GR2_HW[10:0] are updated when GR2_IBUS_VEN and GR2_P_VEN in GR2_UPDATE are 1.

38.6.2.9 Alpha Blending Control Register 1 (Graphics 2) (GR2_AB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_ARC_MUL	GR2_ACALC_MD	—	GR2_ARC_ON	—	—	—	GR2_ARC_DISP_ON	—	—	—	GR2_GRC_DISP_ON	—	—	GR2_DISP_SEL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	GR2_ARC_MUL	0	R/W	Turns on/off multiplication processing with current alpha at alpha blending in a rectangular area. 0: Off 1: On
14	GR2_ACALC_MD	0	R/W	Turns on/off premultiplication processing at alpha blending in one-pixel units. 0: Off 1: On
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	GR2_ARC_ON	0	R/W	Turns on/off alpha blending in a rectangular area. 0: Off 1: On
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR2_ARC_DISP_ON	0	R/W	Turns on/off frame-line display of the image area for alpha blending in a rectangular area. 0: Frame-line display off 1: Frame-line display on
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR2_GRC_DISP_ON	0	R/W	Turns on/off frame-line display of the graphics image area. 0: Frame-line display off 1: Frame-line display on
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	GR2_DISP_SEL[1:0]	0	R/W	Selects the graphics display mode. 0: Background color display 1: Lower-layer graphics display 2: Current graphics display 3: Blended display of lower-layer graphics and current graphics

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

38.6.2.10 Alpha Blending Control Register 2 (Graphics 2) (GR2_AB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR2_GRC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR2_GRC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR2_GRC_VS[10:0]	0	R/W	Sets the vertical start position of the graphics image area. **NOTE** Set to 4 or greater lines and the result of GR2_GRC_VS[10:0] + GR2_GRC_VW[10:0] should be smaller than or equal to 2039 lines.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR2_GRC_VW[10:0]	0	R/W	Sets the vertical width of the graphics image area. **NOTE** GR0_GRC_VS[10:0] + GR0_GRC_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

38.6.2.11 Alpha Blending Control Register 3 (Graphics 2) (GR2_AB3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR2_GRC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR2_GRC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR2_GRC_HS[10:0]	0	R/W	Sets the horizontal start position of the graphics image area. **NOTE** Set to 16 or greater clocks and the result of GR2_GRC_HS[10:0] + GR2_GRC_HW[10:0] should be smaller than or equal to 2015 clocks.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR2_GRC_HW[10:0]	0	R/W	Sets the horizontal width of the graphics image area. **NOTES** 1. For displaying an image with 1- or 2-pixel horizontal width, set GR2_HW[10:0] to 2 and GR2_GRC_HW[10:0] to 1 (1-pixel) or 2 (2-pixel). 2. GR2_GRC_HS[10:0] + GR2_GRC_HW[10:0] should be smaller than or equal to 2015 clocks.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

38.6.2.12 Alpha Blending Control Register 4 (Graphics 2) (GR2_AB4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR2_ARC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR2_ARC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR2_ARC_VS[10:0]	0	R/W	Sets the vertical start position of the valid image area for alpha blending in a rectangular area.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR2_ARC_VW[10:0]	0	R/W	Sets the vertical width of the valid image area for alpha blending in a rectangular area.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

38.6.2.13 Alpha Blending Control Register 5 (Graphics 2) (GR2_AB5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR2_ARC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR2_ARC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR2_ARC_HS[10:0]	0	R/W	Sets the horizontal start position of the valid image area for alpha blending in a rectangular area.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR2_ARC_HW[10:0]	0	R/W	Sets the horizontal width of the valid image area for alpha blending in a rectangular area.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

38.6.2.14 Alpha Blending Control Register 6 (Graphics 2) (GR2_AB6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	GR2_ARC_MODE	GR2_ARC_COEF[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	GR2_ARC_RATE[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	GR2_ARC_MODE	0	R/W	Alpha Blending Mode in Rectangular Area 0: Addition 1: Subtraction
23 to 16	GR2_ARC_COEF[7:0]	0	R/W	Sets the alpha coefficient for alpha blending in a rectangular area. (0 to 255) [7:0]: Variation (absolute value)
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	GR2_ARC_RATE[7:0]	0	R/W	Sets the value obtained by subtracting 1 from the frame rate for alpha blending in a rectangular area.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

38.6.2.15 Alpha Blending Control Register 7 (Graphics 2) (GR2_AB7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR2_ARC_DEF[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR2_CK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR2_ARC_DEF[7:0]	255	R/W	Sets the initial alpha value for alpha blending in a rectangular area.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR2_CK_ON	0	R/W	Turns on/off CLUT-index/RGB-index chroma-key processing. 0: Off 1: On

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

38.6.2.16 Alpha Blending Control Register 8 (Graphics 2) (GR2_AB8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR2_CK_KCLUT[7:0]								GR2_CK_KG[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_CK_KB[7:0]								GR2_CK_KR[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR2_CK_KCLUT[7:0]	0	R/W	CLUT Signal for CLUT-Index Chroma-Key Processing CLUT: Unsigned 8 bits (0 to 255 [LSB])
23 to 16	GR2_CK_KG[7:0]	0	R/W	G Signal for RGB-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR2_CK_KB[7:0]	0	R/W	B Signal for RGB-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR2_CK_KR[7:0]	0	R/W	R Signal for RGB-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

38.6.2.17 Alpha Blending Control Register 9 (Graphics 2) (GR2_AB9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR2_CK_A[7:0]								GR2_CK_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_CK_B[7:0]								GR2_CK_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR2_CK_A [7:0]	0	R/W	Replaced Alpha Signal after RGB/CLUT-Index Chroma-Key Processing α: Unsigned 8 bits (0 to 255 [LSB])
23 to 16	GR2_CK_G [7:0]	0	R/W	Replaced G Signal after RGB/CLUT-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR2_CK_B [7:0]	0	R/W	Replaced B Signal after RGB/CLUT-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR2_CK_R [7:0]	0	R/W	Replaced R Signal after RGB/CLUT-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

38.6.2.18 Alpha Blending Control Register 10 (Graphics 2) (GR2_AB10)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR2_A0[7:0]								GR2_G0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_B0[7:0]								GR2_R0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR2_A0 [7:0]	0	R/W	CLUT0 α0 Signal Replaced with α signal when in the CLUT0 format and CLUT0= 0. Replaced with α signal when in the αRGB1555 or RGBα5551 format and α = 0.
23 to 16	GR2_G0 [7:0]	0	R/W	CLUT0 G0 Signal Replaced with G signal when in the CLUT0 format and CLUT0 = 0.
15 to 8	GR2_B0 [7:0]	0	R/W	CLUT0 B0 Signal Replaced with B signal when in the CLUT0 format and CLUT0 = 0.
7 to 0	GR2_R0 [7:0]	0	R/W	CLUT0 R0 Signal Replaced with R signal when in the CLUT0 format and CLUT0 = 0.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

38.6.2.19 Alpha Blending Control Register 11 (Graphics 2) (GR2_AB11)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR2_A1[7:0]								GR2_G1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_B1[7:0]								GR2_R1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR2_A1 [7:0]	0	R/W	CLUT1 α1 Signal Replaced with α signal when in the CLUT1 format and CLUT1 = 1. Replaced with α signal when in the αRGB1555 or RGBα5551 format and α = 1.
23 to 16	GR2_G1 [7:0]	0	R/W	CLUT1 G1 Signal Replaced with G signal when in the CLUT1 format and CLUT1 = 1.
15 to 8	GR2_B1 [7:0]	0	R/W	CLUT1 B1 Signal Replaced with B signal when in the CLUT1 format and CLUT1 = 1.
7 to 0	GR2_R1 [7:0]	0	R/W	CLUT1 R1 Signal Replaced with R signal when in the CLUT1 format and CLUT1 = 1.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

38.6.2.20 Background Color Control Register (Graphics 2) (GR2_BASE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR2_BASE_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR2_BASE_B[7:0]								GR2_BASE_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR2_BASE_G [7:0]	0	R/W	Background Color G Signal G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR2_BASE_B [7:0]	0	R/W	Background Color B Signal B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR2_BASE_R [7:0]	0	R/W	Background Color R Signal R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

38.6.2.21 CLUT Table Control Register (Graphics 2) (GR2_CLUT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR2_CLT_SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR2_CLT_SEL	0	R/W	CLUT Table Select Signal 0: Selects CLUT table 0. The format is converted to αRGB8888 based on the CLUT table 0. CLUT table 1 can be read from or written to by the CPU. 1: Selects CLUT table 1. The format is converted to αRGB8888 based on the CLUT table 1. CLUT table 0 can be read from or written to by the CPU.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: This register is updated when GR2_P_VEN in GR2_UPDATE is 1.

NOTE

Before switching CLUT table by GR_CLT_SEL, CPU must wait until the read value from the last written address is same as the last written value.

Refer to Section 38.6.1.15, CLUT Table for detail.

38.6.2.22 Status Monitor Register (Graphics 2) (GR2_MON)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR2_ARC_ST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR2_ARC_ST	0	R	Status Flag for Alpha Blending in Rectangular Area 0: Addition or subtraction has been completed. (α value is 0 or 255) 1: Addition or subtraction is in progress.

38.6.2.23 Graphics 3 Register Update Control Register (GR3_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GR3_UPDATE	—	—	—	GR3_P_VEN	—	—	—	GR3_IBUS_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR3_UPDATE	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR3_P_VEN	0	R/WC1	Graphics Display Register Update 0: Registers are not updated 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR3_IBUS_VEN	0	R/WC1	Frame Buffer Read Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.6.2.24 Frame Buffer Read Control Register (Graphics 3) (GR3_FLM_RD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR3_R_ENB
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR3_R_ENB	0	R/W	Frame Buffer Read Enable 0: Frame buffer reading is disabled. 1: Frame buffer reading is enabled.

Note: This register is updated when GR3_IBUS_VEN in GR3_UPDATE is 1.

38.6.2.25 Frame Buffer Control Register 1 (Graphics 3) (GR3_FLM1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR3_LN_OFF_DIR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR3_FLM_SEL[1:0]		—	—	—	—	—	—	—	Caution *
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

CAUTION

The initial value “0” of bit 0 must be changed to “1”.

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR3_LN_OFF_DIR	0	R/W	Selects the line offset address direction of the frame buffer. 0: Increments the address by the line offset address. 1: Decrements the address by the line offset address.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	GR3_FLM_SEL[1:0]	0	R/W	Selects a frame buffer address setting signal. 0: Selects frame 0. 1: Selects register GR3_FLM_NUM[9:0]. 2: Selects frame 0. 3: Setting prohibited
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	Bit0	0	R/W	**CAUTION** The initial value “0” of this bit must be changed to “1”.

Note: GR3_LN_OFF_DIR and GR3_FLM_SEL[1:0] are updated when GR3_IBUS_VEN in GR3_UPDATE is 1.

38.6.2.26 Frame Buffer Control Register 2 (Graphics 3) (GR3_FLM2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR3_BASE[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_BASE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	GR3_BASE [31:0]	0	R/W	Frame Buffer Base Address (upper) Sets the start address of the frame buffer where frame data is to be stored. The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when GR3_IBUS_VEN and GR3_P_VEN in GR3_UPDATE are 1.

38.6.2.27 Frame Buffer Control Register 3 (Graphics 3) (GR3_FLM3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	GR3_LN_OFF[14:0]														
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	GR3_FLM_NUM[9:0]									
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 16	GR3_LN_OFF[14:0]	0	R/W	Frame Buffer Line Offset Address Sets the line offset address for calculating the start address of each line. Line 0: GR3_BASE[31:0] Line 1: GR3_BASE[31:0] + GR3_LN_OFF[14:0] × 1 : Line n: GR3_BASE[31:0] + GR3_LN_OFF[14:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	GR3_FLM_NUM[9:0]	0	R/W	Frame Number of Frame Buffer Manually set the frame number when GR3_FLM_SEL[1:0] = 1.

Note: This register is updated when GR3_IBUS_VEN in GR3_UPDATE is 1.

38.6.2.28 Frame Buffer Control Register 4 (Graphics 3) (GR3_FLM4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	GR3_FLM_OFF[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_FLM_OFF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	GR3_FLM_OFF[22:0]	0	R/W	Frame Buffer Frame Offset Address (upper) Specifies the frame offset address used for calculating the start address of each frame buffer when more than one buffer is used. Buffer 0: GR3_BASE[31:0] Buffer 1: GR3_BASE[31:0] + GR3_FLM_OFF[22:0] × 1 : Buffer n: GR3_BASE[31:0] + GR3_FLM_OFF[22:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when GR3_IBUS_VEN in GR3_UPDATE is 1.

38.6.2.29 Frame Buffer Control Register 5 (Graphics 3) (GR3_FLM5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR3_FLM_LNUM[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR3_FLM_LOOP[10:0]										
Initial value:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR3_FLM_LNUM[10:0]	0	R/W	Sets number of lines in a frame The number of lines is (GR3_FLM_LNUM[10:0] + 1).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR3_FLM_LOOP[10:0]	1023	R/W	Number of lines when reading the addresses repeatedly by returning to the start address after reaching the end address. The number of lines is (GR3_FLM_LOOP[10:0] + 1).

Note: This register is updated when GR3_IBUS_VEN in GR3_UPDATE is 1.

38.6.2.30 Frame Buffer Control Register 6 (Graphics 3) (GR3_FLM6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR3_FORMAT[3:0]				—	GR3_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	GR3_RDSWA[2:0]			—	—	—	—	GR3_STA_POS[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	GR3_ FORMAT [3:0]	0	R/W	Sets the format of the frame buffer read signal. 0: RGB565 1: RGB888 2: αRGB1555 3: αRGB4444 4: αRGB8888 5: CLUT8 6: CLUT4 7: CLUT1 8: Setting prohibited 9: Setting prohibited 10: RGBα5551 11: RGBα8888 12 to 15: Setting prohibited
27	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
26 to 16	GR3_HW [10:0]	0	R/W	Sets the width of the horizontal valid period. The width is (GR3_HW[10:0] + 1) pixels. Note: Set to 2 or greater.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12 to 10	GR3_ RDSWA [2:0]	0	R/W	Sets 8-, 16-, and 32-bit swap. These three bits specify the method for swapping the bits of frame buffer read data as follows. Bit 0 0: 8 bits are not swapped. 1: 8 bits are swapped. Bit 1 0: 16 bits are not swapped. 1: 16 bits are swapped. Bit 2 0: 32 bits are not swapped. 1: 32 bits are swapped. When eight bits are put together, they are swapped as follows. Each of (1) to (8) indicates eight-bit data. 000: (1) (2) (3) (4) (5) (6) (7) (8) [No swap] 001: (2) (1) (4) (3) (6) (5) (8) (7) [8-bit swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [16-bit swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [16-bit swap + 8-bit swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [32-bit swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [32-bit swap + 8-bit swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [32-bit swap + 16-bit swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [32-bit swap + 16-bit swap + 8-bit swap]
9 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	GR3_STA_ POS[5:0]	0	R/W	Sets the amount of data to be skipped through. Specifically data amount equal to the amount indicated by GR3_STA_POS[5:0] is skipped from the start of the line.

Note: GR3_STA_POS[5:0] is updated when GR3_P_VEN in GR3_UPDATE is 1. GR3_RDSWA[2:0] is updated when GR3_UPDATE in GR3_UPDATE is 1.
GR3_FORMAT[3:0] and GR3_HW[10:0] are updated when GR3_IBUS_VEN and GR3_P_VEN in GR3_UPDATE are 1.

38.6.2.31 Alpha Blending Control Register 1 (Graphics 3) (GR3_AB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_ARC_MUL	GR3_ACALC_MD	—	GR3_ARC_ON	—	—	—	GR3_ARC_DISP_ON	—	—	—	GR3_GRC_DISP_ON	—	—	GR3_DISP_SEL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15	GR3_ARC_MUL	0	R/W	Turns on/off multiplication processing with current alpha at alpha blending in a rectangular area. 0: Off 1: On
14	GR3_ACALC_MD	0	R/W	Turns on/off premultiplication processing at alpha blending in one-pixel units. 0: Off 1: On
13	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	GR3_ARC_ON	0	R/W	Turns on/off alpha blending in a rectangular area. 0: Off 1: On
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR3_ARC_DISP_ON	0	R/W	Turns on/off frame-line display of the image area for alpha blending in a rectangular area. 0: Frame-line display off 1: Frame-line display on
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR3_GRC_DISP_ON	0	R/W	Turns on/off frame-line display of the graphics image area. 0: Frame-line display off 1: Frame-line display on
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	GR3_DISP_SEL[1:0]	0	R/W	Selects the graphics display mode. 0: Background color display 1: Lower-layer graphics display 2: Current graphics display 3: Blended display of lower-layer graphics and current graphics

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

38.6.2.32 Alpha Blending Control Register 2 (Graphics 3) (GR3_AB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR3_GRC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR3_GRC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR3_GRC_VS[10:0]	0	R/W	Sets the vertical start position of the graphics image area. Note: Set to 4 or greater lines and the result of GR3_GRC_VS[10:0] + GR3_GRC_VW[10:0] should be smaller than or equal to 2039 lines.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR3_GRC_VW[10:0]	0	R/W	Sets the vertical width of the graphics image area. **NOTE** GR3_GRC_VS[10:0] + GR3_GRC_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

38.6.2.33 Alpha Blending Control Register 3 (Graphics 3) (GR3_AB3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR3_GRC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR3_GRC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR3_GRC_HS[10:0]	0	R/W	Sets the horizontal start position of the graphics image area. Note: Set to 16 or greater clocks and the result of GR3_GRC_HS[10:0] + GR3_GRC_HW[10:0] should be smaller than or equal to 2015 clocks.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR3_GRC_HW[10:0]	0	R/W	Sets the horizontal width of the graphics image area. **NOTES** 1. For displaying an image with 1- or 2-pixel horizontal width, set GR3_HW[10:0] to 2 and GR3_GRC_HW[10:0] to 1 (1-pixel) or 2 (2-pixel). 2. GR3_GRC_HS[10:0] + GR3_GRC_HW[10:0] should be smaller than or equal to 2015 clocks.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

38.6.2.34 Alpha Blending Control Register 4 (Graphics 3) (GR3_AB4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR3_ARC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR3_ARC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR3_ARC_VS[10:0]	0	R/W	Sets the vertical start position of the valid image area for alpha blending in a rectangular area.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR3_ARC_VW[10:0]	0	R/W	Sets the vertical width of the valid image area for alpha blending in a rectangular area.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

38.6.2.35 Alpha Blending Control Register 5 (Graphics 3) (GR3_AB5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR3_ARC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR3_ARC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR3_ARC_HS[10:0]	0	R/W	Sets the horizontal start position of the valid image area for alpha blending in a rectangular area.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR3_ARC_HW[10:0]	0	R/W	Sets the horizontal width of the valid image area for alpha blending in a rectangular area.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

38.6.2.36 Alpha Blending Control Register 6 (Graphics 3) (GR3_AB6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	GR3_ARC_MODE	GR3_ARC_COEF[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	GR3_ARC_RATE[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	GR3_ARC_MODE	0	R/W	Alpha Blending Mode in Rectangular Area 0: Addition 1: Subtraction
23 to 16	GR3_ARC_COEF[7:0]	0	R/W	Sets the alpha coefficient for alpha blending in a rectangular area. (0 to 255) [7:0]: Variation (absolute value)
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	GR3_ARC_RATE[7:0]	0	R/W	Sets the value obtained by subtracting 1 from the frame rate for alpha blending in a rectangular area.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

38.6.2.37 Alpha Blending Control Register 7 (Graphics 3) (GR3_AB7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR3_ARC_DEF[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR3_CK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR3_ARC_DEF[7:0]	255	R/W	Sets the initial alpha value for alpha blending in a rectangular area.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR3_CK_ON	0	R/W	Turns on/off CLUT-index/RGB-index chroma-key processing. 0: Off 1: On

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

38.6.2.38 Alpha Blending Control Register 8 (Graphics 3) (GR3_AB8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR3_CK_KCLUT[7:0]								GR3_CK_KG[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_CK_KB[7:0]								GR3_CK_KR[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR3_CK_KCLUT[7:0]	0	R/W	CLUT Signal for CLUT-Index Chroma-Key Processing CLUT: Unsigned 8 bits (0 to 255 [LSB])
23 to 16	GR3_CK_KG[7:0]	0	R/W	G Signal for RGB-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR3_CK_KB[7:0]	0	R/W	B Signal for RGB-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR3_CK_KR[7:0]	0	R/W	R Signal for RGB-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

38.6.2.39 Alpha Blending Control Register 9 (Graphics 3) (GR3_AB9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR3_CK_A[7:0]								GR3_CK_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_CK_B[7:0]								GR3_CK_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR3_CK_A [7:0]	0	R/W	Replaced Alpha Signal after RGB/CLUT-Index Chroma-Key Processing α: Unsigned 8 bits (0 to 255 [LSB])
23 to 16	GR3_CK_G [7:0]	0	R/W	Replaced G Signal after RGB/CLUT-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR3_CK_B [7:0]	0	R/W	Replaced B Signal after RGB/CLUT-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR3_CK_R [7:0]	0	R/W	Replaced R Signal after RGB/CLUT-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

38.6.2.40 Alpha Blending Control Register 10 (Graphics 3) (GR3_AB10)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR3_A0[7:0]								GR3_G0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_B0[7:0]								GR3_R0[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR3_A0 [7:0]	0	R/W	CLUT0 α0 Signal Replaced with α signal when in the CLUT0 format and CLUT0 = 0. Replaced with α signal when in the αRGB1555 or RGBα5551 format and α = 0.
23 to 16	GR3_G0 [7:0]	0	R/W	CLUT0 G0 Signal Replaced with G signal when in the CLUT0 format and CLUT0 = 0.
15 to 8	GR3_B0 [7:0]	0	R/W	CLUT0 B0 Signal Replaced with B signal when in the CLUT0 format and CLUT0 = 0.
7 to 0	GR3_R0 [7:0]	0	R/W	CLUT0 R0 Signal Replaced with R signal when in the CLUT0 format and CLUT0 = 0.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

38.6.2.41 Alpha Blending Control Register 11 (Graphics 3) (GR3_AB11)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR3_A1[7:0]								GR3_G1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_B1[7:0]								GR3_R1[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR3_A1 [7:0]	0	R/W	CLUT1 α1 Signal Replaced with α signal when in the CLUT1 format and CLUT1 = 1. Replaced with α signal when in the αRGB1555 or RGBα5551 format and α = 1.
23 to 16	GR3_G1 [7:0]	0	R/W	CLUT1 G1 Signal Replaced with G signal when in the CLUT1 format and CLUT1 = 1.
15 to 8	GR3_B1 [7:0]	0	R/W	CLUT1 B1 Signal Replaced with B signal when in the CLUT1 format and CLUT1 = 1.
7 to 0	GR3_R1 [7:0]	0	R/W	CLUT1 R1 Signal Replaced with R signal when in the CLUT1 format and CLUT1 = 1.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

38.6.2.42 Background Color Control Register (Graphics 3) (GR3_BASE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR3_BASE_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR3_BASE_B[7:0]								GR3_BASE_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR3_BASE_G [7:0]	0	R/W	Background Color G Signal G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR3_BASE_B [7:0]	0	R/W	Background Color B Signal B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR3_BASE_R [7:0]	0	R/W	Background Color R Signal R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

38.6.2.43 CLUT Table and Interrupt Control Register (Graphics 3) (GR3_CLUT_INT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR3_CLT_SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR3_LINE[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	GR3_CLT_SEL	0	R/W	CLUT Table Select Signal 0: Selects CLUT table 0. The format is converted to αRGB8888 based on the CLUT table 0. CLUT table 1 can be read from or written to by the CPU. 1: Selects CLUT table 1. The format is converted to αRGB8888 based on the CLUT table 1. CLUT table 0 can be read from or written to by the CPU.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR3_LINE[10:0]	0	R/W	Line Interrupt Set When number of lines matches the value of the GR3_LINE[10:0] bits, an interrupt signal is output. This function is enabled even when the graphics 3 process is not used.

Note: This register is updated when GR3_P_VEN in GR3_UPDATE is 1.

NOTE

Before switching CLUT table by GR_CLT_SEL, CPU must wait until the read value from the last written address is same as the last written value.

Refer to Section 38.6.1.15, CLUT Table for detail.

38.6.2.44 Status Monitor Register (Graphics 3) (GR3_MON)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR3_LIN_STAT[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR3_ARC_ST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR3_LIN_STAT[10:0]	0	R	Line Position of Image being Currently Read
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR3_ARC_ST	0	R	Status Flag for Alpha Blending in Rectangular Area 0: Addition or subtraction has been completed. (α value is 0 or 255) 1: Addition or subtraction is in progress.

38.6.2.45 VIN Synthesizer Register Update Control Register (GR_VIN_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GR_VIN_UPDATE	—	—	—	GR_VIN_P_VEN	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR_VIN_UPDATE	0	R/WC1	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR_VIN_P_VEN	0	R/WC1	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

38.6.2.46 Alpha Blending Control Register 1 (VIN Synthesizer) (GR_VIN_AB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	GR_VIN_ARC_ON	—	—	—	GR_VIN_ARC_DISP_ON	—	—	—	GR_VIN_GRC_DISP_ON	—	GR_VIN_SCL_UND_SEL	GR_VIN_DISP_SEL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	GR_VIN_ARC_ON	0	R/W	Turns on/off alpha blending in a rectangular area. 0: Off 1: On
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR_VIN_ARC_DISP_ON	0	R/W	Turns on/off frame-line display of the image area for alpha blending in a rectangular area. 0: Frame-line display off 1: Frame-line display on
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR_VIN_GRC_DISP_ON	0	R/W	Turns on/off frame-line display of the graphics image area. 0: Frame-line display off 1: Frame-line display on
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2	GR_VIN_SCL_UND_SEL	0	R/W	Selection of Lower-Layer Plane in Scaler 0: Selects graphics 0 as lower-layer graphics and graphics 1 as current graphics 1: Selects graphics 1 as lower-layer graphics and graphics 0 as current graphics
1, 0	GR_VIN_DISP_SEL[1:0]	0	R/W	Selects the graphics display mode. 0: Background color display 1: Lower-layer graphics display 2: Current graphics display 3: Blended display of lower-layer graphics and current graphics, or setting prohibited*1

Note 1. This setting is prohibited when the graphics block 0 in scaler 0 and graphics block 1 in scaler 1 are cascaded (GR1_AB1.GR1_CUS_CON_ON = 1).

Note: GR_VIN_SCL_UND_SEL is updated when GR_VIN_UPDATE in GR_VIN_UPDATE is 1. The other bits of this register are updated when GR_VIN_P_VEN in GR_VIN_UPDATE is 1.

38.6.2.47 Alpha Blending Control Register 2 (VIN Synthesizer) (GR_VIN_AB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR_VIN_GRC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR_VIN_GRC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR_VIN_GRC_VS[10:0]	0	R/W	Sets the vertical start position of the graphics image area. **NOTE** Set to 4 or greater lines and the result of GR_VIN_GRC_VS[10:0] + GR_VIN_GRC_VW[10:0] should be smaller than or equal to 2039 lines.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR_VIN_GRC_VW[10:0]	0	R/W	Sets the vertical width of the graphics image area. **NOTE** GR_VIN_GRC_VS[10:0] + GR_VIN_GRC_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when GR_VIN_P_VEN in GR_VIN_UPDATE is 1.

38.6.2.48 Alpha Blending Control Register 3 (VIN Synthesizer) (GR_VIN_AB3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR_VIN_GRC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR_VIN_GRC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR_VIN_GRC_HS[10:0]	0	R/W	Sets the horizontal start position of the graphics image area. **NOTE** Set to 16 or greater clocks and the result of GR_VIN_GRC_HS[10:0] + GR_VIN_GRC_HW[10:0] should be smaller than or equal to 2015 clocks.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR_VIN_GRC_HW[10:0]	0	R/W	Sets the horizontal width of the graphics image area. **NOTES** 1. For displaying an image with 1- or 2-pixel horizontal width, set GR_VIN_GRC_HW[10:0] to 1 (1-pixel) or 2 (2-pixel). 2. GR_VIN_GRC_HS[10:0] + GR_VIN_GRC_HW[10:0] should be smaller than or equal to 2015 clocks.

Note: This register is updated when GR_VIN_P_VEN in GR_VIN_UPDATE is 1.

38.6.2.49 Alpha Blending Control Register 4 (VIN Synthesizer) (GR_VIN_AB4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR_VIN_ARC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR_VIN_ARC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR_VIN_ARC_VS[10:0]	0	R/W	Sets the vertical start position of the valid image area for alpha blending in a rectangular area.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR_VIN_ARC_VW[10:0]	0	R/W	Sets the vertical width of the valid image area for alpha blending in a rectangular area.

Note: This register is updated when GR_VIN_P_VEN in GR_VIN_UPDATE is 1.

38.6.2.50 Alpha Blending Control Register 5 (VIN Synthesizer) (GR_VIN_AB5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR_VIN_ARC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR_VIN_ARC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR_VIN_ARC_HS[10:0]	0	R/W	Sets the horizontal start position of the valid image area for alpha blending in a rectangular area.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR_VIN_ARC_HW[10:0]	0	R/W	Sets the horizontal width of the valid image area for alpha blending in a rectangular area.

Note: This register is updated when GR_VIN_P_VEN in GR_VIN_UPDATE is 1.

38.6.2.51 Alpha Blending Control Register 6 (VIN Synthesizer) (GR_VIN_AB6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	GR_VIN_ARC_MODE	GR_VIN_ARC_COEF[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	GR_VIN_ARC_RATE[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	GR_VIN_ARC_MODE	0	R/W	Alpha Blending Mode in Rectangular Area 0: Addition 1: Subtraction
23 to 16	GR_VIN_ARC_COEF[7:0]	0	R/W	Sets the alpha coefficient for alpha blending in a rectangular area. (0 to 255) [7:0]: Variation (absolute value)
15 to 8	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
7 to 0	GR_VIN_ARC_RATE[7:0]	0	R/W	Sets the value obtained by subtracting 1 from the frame rate for alpha blending in a rectangular area.

Note: This register is updated when GR_VIN_P_VEN in GR_VIN_UPDATE is 1.

38.6.2.52 Alpha Blending Control Register 7 (VIN Synthesizer) (GR_VIN_AB7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR_VIN_ARC_DEF[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR_VIN_ ARC_DEF [7:0]	255	R/W	Sets the initial alpha value for alpha blending in a rectangular area.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: This register is updated when GR_VIN_P_VEN in GR_VIN_UPDATE is 1.

38.6.2.53 Background Color Control Register (VIN Synthesizer) (GR_VIN_BASE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR_VIN_BASE_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR_VIN_BASE_B[7:0]								GR_VIN_BASE_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR_VIN_BASE_G[7:0]	0	R/W	Background Color G Signal G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR_VIN_BASE_B[7:0]	0	R/W	Background Color B Signal B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR_VIN_BASE_R[7:0]	0	R/W	Background Color R Signal R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when GR_VIN_P_VEN in GR_VIN_UPDATE is 1.

38.6.2.54 Status Monitor Register (VIN Synthesizer) (GR_VIN_MON)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR_VIN_ARC_ST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR_VIN_ARC_ST	0	R	Status Flag for Alpha Blending in Rectangular Area 0: Addition or subtraction has been completed. (α value is 0 or 255) 1: Addition or subtraction is in progress.

38.6.3 Usage Method

38.6.3.1 Mute Image

The initial values of the GR0_DISP_SEL[1:0], GR1_DISP_SEL[1:0], GR2_DISP_SEL[1:0], GR3_DISP_SEL[1:0], GR_VIN_DISP_SEL[1:0], and GR_OIR_DISP_SEL[1:0] bits are all 0. Accordingly, in the initial setting, a background color is displayed both inside and outside the graphics area for the graphics 0, 1, 2, 3, and OIR processes and the VIN synthesizer. Since the default background color is black, the black mute image is displayed in the initial state.

38.6.3.2 Alpha Blending in Rectangular Area

The alpha coefficient and the frame rate can be changed during fade in and fade out by modifying the GR_ARC_MODE, GR_ARC_COEF[7:0] and GR_ARC_RATE[7:0] bits, respectively.

38.7 Output Image Generator

NOTE

The Output Image Generator is not available for D1L2(H) devices.

38.7.1 Output Image Generation Functions

38.7.1.1 Overview of Functions

The output image generator can read the graphics data and display them. In cooperation with the Video Output Warping Engine (VOWE) (channel 0 only), it can also distort the output image according to a display panel.

When the output image generator is not in use, the signals from the image synthesizer can be output directly to the output control block, bypassing the output image generator.

The output image generator has the same circuit configuration as the scaler. However, note the following differences from the scaler.

(1) The scaling-up/-down function is not available.

(2) The rotation control function is not available.

(3) The pointer buffer function is not available.

(4) The field control function is not available because the inputs are always progressive signals.

(5) The I/O signals are always RGB signals.

The functional block diagram of the output image generator is shown below.

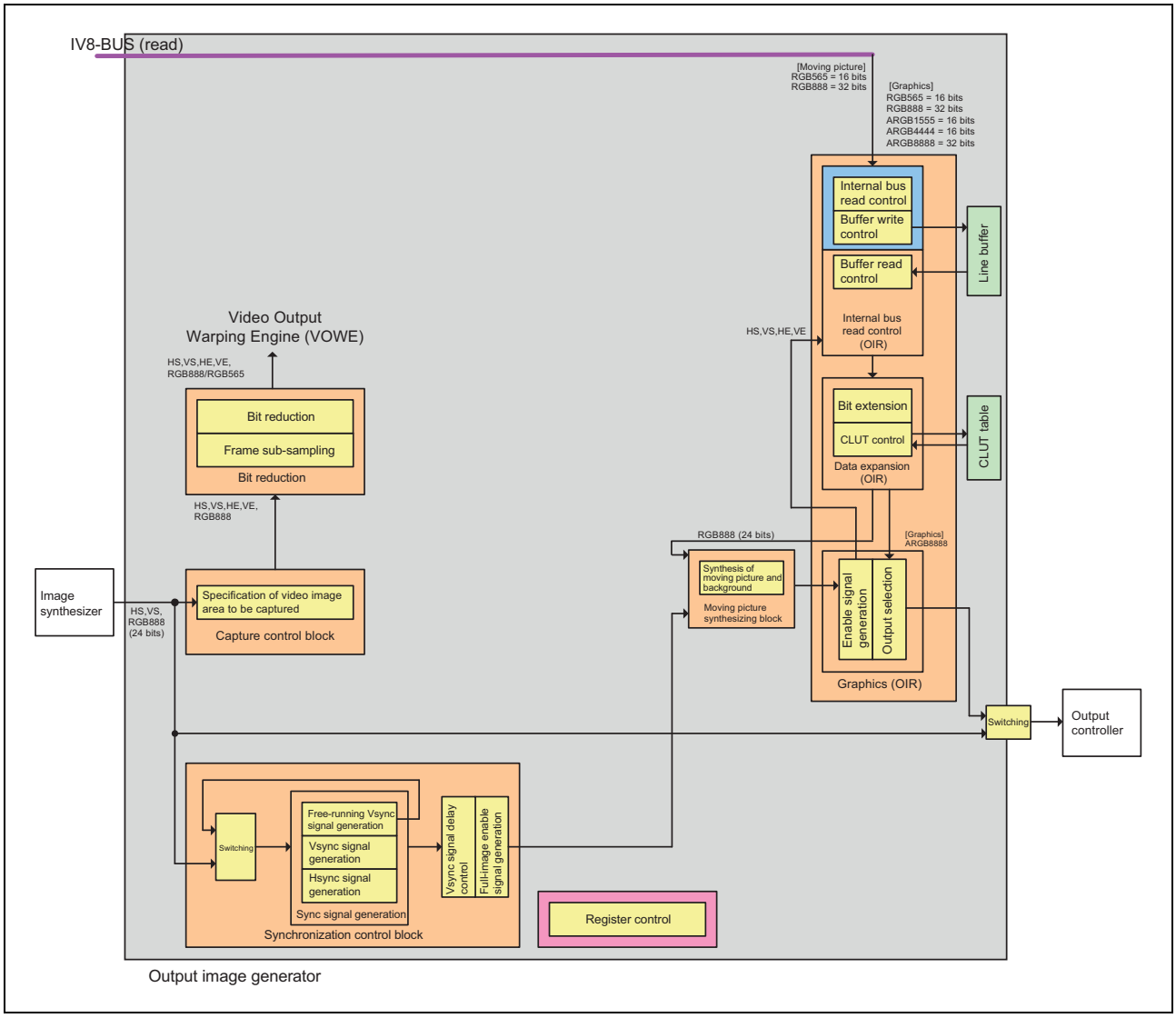


Figure 38.56 Functional Block Diagram of Output Image Generator

38.7.1.2 Register Control

(1) Updating Registers

The Vsync signal is used to control the update timing of all the registers of the output image generator and graphics block (OIR) except some registers of the sync control block.

After 1 is set to the bits in the update control register, the contents of the relevant registers are modified at the rising edge of the Vsync signal. The update control register is automatically cleared to 0 after the modification.

Table 38.129 Register Update Control

Register Name	Bit Name	Initial Value	Description
OIR_SCL0_ UPDATE	OIR_SCL0_ UPDATE	0	SYNC Control Register Update 0: Registers are not updated. 1: Registers are updated.
OIR_SCL0_ UPDATE	OIR_SCL0_ VEN_D	0	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
OIR_SCL0_ UPDATE	OIR_SCL0_ VEN_C	0	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
OIR_SCL0_ UPDATE	OIR_SCL0_ VEN_B	0	Synchronization Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
OIR_SCL0_ UPDATE	OIR_SCL0_ VEN_A	0	Scaling-Down Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
OIR_SCL1_ UPDATE	OIR_SCL1_ UPDATE_A	0	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated.
OIR_SCL1_ UPDATE	OIR_SCL1_ VEN_B	0	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
OIR_SCL1_ UPDATE	OIR_SCL1_ VEN_A	0	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GR_OIR_UPDATE	GR_OIR_P_ VEN	0	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GR_OIR_UPDATE	GR_OIR_IBUS_ VEN	0	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

The registers controlled by OIR_SCL0_VEN_A, OIR_SCL_0_VEN_C, OIR_SCL1_VEN_A, and OIR_SCL1_VEN_B are modified at the rising edge of the input Vsync signal.

The registers controlled by OIR_SCL0_VEN_B, OIR_SCL0_VEN_D, GR_OIR_P_VEN, and GR_OIR_IBUS_VEN are modified at the rising edge of the output Vsync signal.

38.7.1.3 Enabling or Disabling Output Image Generator

The output image generator can be enabled or disabled. If it is disabled, the input data from the image synthesizer is directly output, bypassing the output image generator.

Table 38.130 Output Image Generator Enable Control

Register Name	Bit Name	Initial Value	Description
OIR_SCL0_FRC3	OIR_RES_EN	0	For VDCE0 only:*1*2 Enabling or Disabling the Output Image Generator 0: The output image generator is disabled (the input data from the image synthesizer is directly output). 1: The output image generator is enabled.

Note 1. For VDCE1 the output image generator must be disabled, thus OIR_RES_EN = 0.

Note 2. For D1L2(H) devices OIR_RES_EN must remain 0.

38.7.1.4 Synchronization Control

(1) Selecting Vsync Signal

The Vsync signal to be output from the output image generator can be selected.

Table 38.131 Vsync Signal Selection Control

Register Name	Bit Name	Initial Value	Description
OIR_SCL0_FRC3	OIR_RES_VS_SEL	1	Vsync Signal Output Select 0: Vsync signal input from the image synthesizer 1: Internally generated free-running Vsync signal

CAUTION

Using external VSYNC does not mean the whole timing is taken from externally. Only the VSYNC is synchronised.

This means if the same timing as for SC0 should be used the same timing for OIR layer as SC0 has to be selected and external VSYNC input has to be selected.

(2) Masking Repeated Vsync Signals

Take measures against the repeated Vsync signals by using the Vsync control block in the scaler. Accordingly, repeated Vsync signal masking control must be disabled in the output image generator when it is in use.

Table 38.132 Repeated Vsync Signal Mask Control

Register Name	Bit Name	Initial Value	Description
OIR_SCL0_FRC1	OIR_RES_VMASK_ON	1	Repeated Vsync Signal Masking Control This bit should always be set to 0 when the output image generator is in use. 0: Repeated Vsync signal masking control is disabled. 1: Repeated Vsync signal masking control is enabled.

(3) Compensating for Missing Vsync Signals

Take measures against the missed Vsync signals by using the Vsync control block in the scaler. Accordingly, compensation of missing Vsync signals must be disabled in the output image generator when it is in use.

Table 38.133 Missing Vsync Compensation Control

Register Name	Bit Name	Initial Value	Description
OIR_SCL0_FRC2	OIR_RES_VLACK_ON	1	Missing Vsync Signal Compensation This bit should always be set to 0 when the output image generator is in use. 0: Compensation of missing Vsync signals is disabled. 1: Compensation of missing Vsync signals is enabled.

(4) Free-Running Period

Free-running Vsync and Hsync periods can be set.

Hsync period [usec] = (RES_FH + 1) ÷ pixel clock frequency [MHz]

Vsync period [usec] = horizontal period [usec] × (RES_FV + 1)

Table 38.134 Free-Running Period Control

Register Name	Bit Name	Initial Value	Description
OIR_SCL0_FRC4	OIR_RES_FV [10:0]	524	Free-Running Vsync Period Setting Free-running Vsync period = (OIR_RES_FV[10:0] + 1) × horizontal period [usec]
OIR_SCL0_FRC4	OIR_RES_FH [10:0]	799	Hsync Period Setting Hsync period [usec] = (OIR_RES_FH[10:0] +1) ÷ pixel clock frequency [MHz]

When selecting a Vsync signal input from the image synthesizer, set the OIR_RES_VS_SEL bit to 0. At this time, the internally generated free-running Vsync signal is not output.

In the meantime, the Hsync signal is always generated according to the free-running signal setting and output from the output image generator.

(5) Vsync Signal Delay Control

Delay of Vsync signal output from the output image generator can be controlled.

The delay is used to adjust the frame buffer read timing.

Table 38.135 Vsync Output Delay Control

Register Name	Bit Name	Initial Value	Description
OIR_SCL0_ FRC5	OIR_RES_VSDLY [7:0]	1	Vsync Signal Delay Control Adjusts the Vsync signal delay in the output Hsync period units. Vsync signal delay [usec]: OIR_RES_VSDLY[7:0] × output Hsync period [usec]

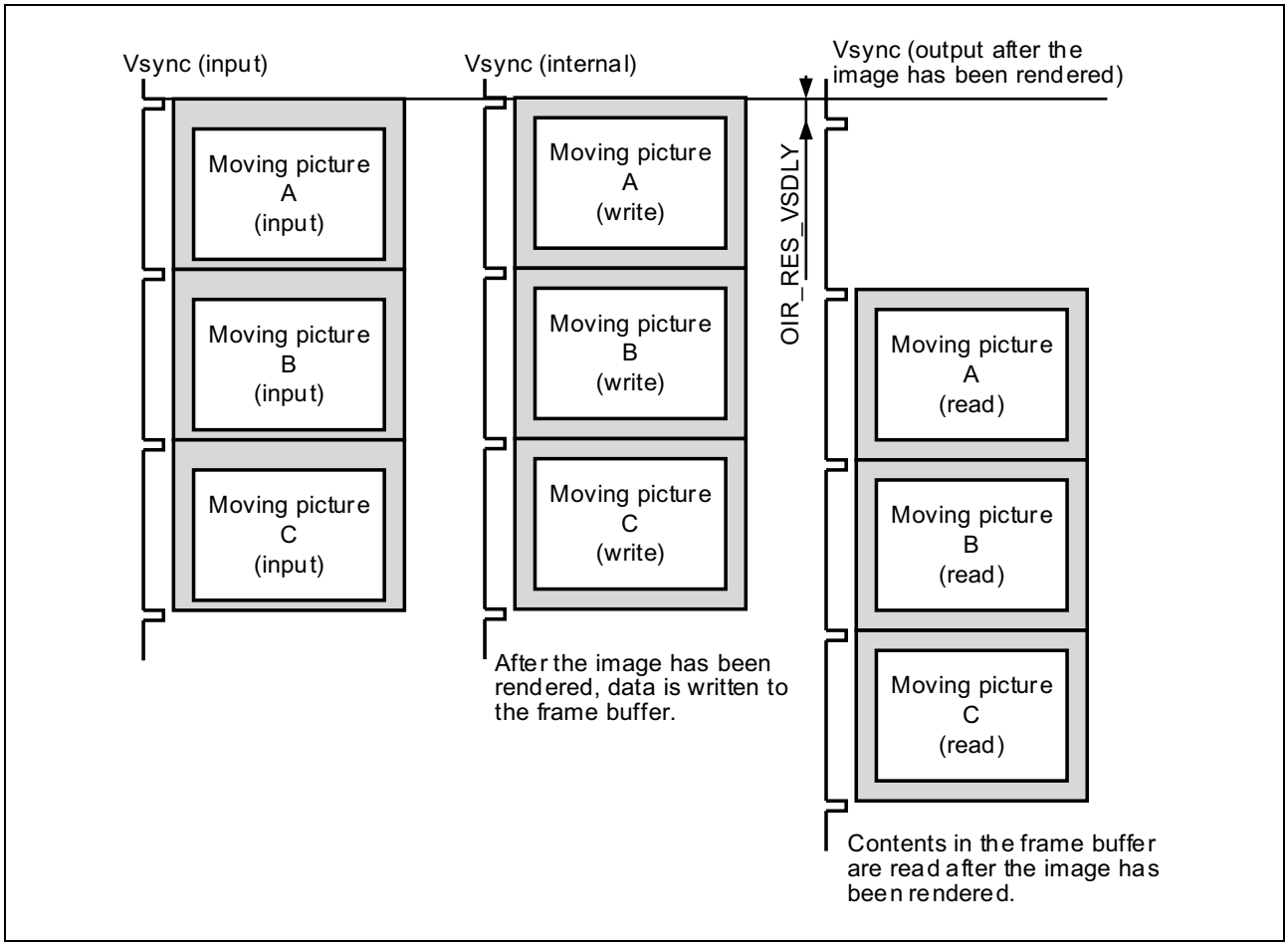


Figure 38.57 Vsync Signal Phases (Two Frame-Buffer Planes Used)

38.7.1.5 Setting Angle of View

(1) Setting Image Area to be Captured

The area to be captured for the input image can be set.

The area is defined by specifying its start position and width based on the input Hsync and Vsync signals.

Table 38.136 Control of Image Area to be Captured

Register Name	Bit Name	Initial Value	Description
OIR_SCL0_DS2	OIR_RES_VS [10:0]	18	Vertical Position Setting for Video Signal Capturing (VSYNC + (V backporch - 1) lines) Note: The set value should be four or more (lines). OIR_RES_VS[10:0] + OIR_RES_VW[10:0] should be equal to or less than 2039 (lines).
OIR_SCL0_DS2	OIR_RES_VW [10:0]	240	Vertical Width of Video Signal to be Captured (lines) Note: OIR_RES_VS[10:0] + OIR_RES_VW[10:0] should be equal to or less than 2039 (lines).
OIR_SCL0_DS3	OIR_RES_HS [10:0]	244	Horizontal Position Setting for Video Signal Capturing (HSYNC + H backporch video-image clock cycles) Note: The set value should be 16 or more (clock cycles). OIR_RES_HS[10:0] + OIR_RES_HW[10:0] should be equal to or less than 2015 (clock cycles).
OIR_SCL0_DS3	OIR_RES_HW [10:0]	1440	Horizontal Width of Video Signal to be Captured (video-image clock cycles) Note: OIR_RES_HS[10:0] + OIR_RES_HW[10:0] should be equal to or less than 2015 (clock cycles).

(2) Generating a Full-Screen Enable Signal

The valid period of the full screen to be output from the output image generator can be set.

The valid period is defined by specifying its start position and width based on the Hsync and Vsync signals output from the output image generator.

The vertical front porch should be set to four or more lines, and the horizontal front porch should be 16 or more clock cycles.

Table 38.137 Full-Screen Enable Control

Register Name	Bit Name	Initial Value	Description
OIR_SCL0_FRC6	OIR_RES_F_VS [10:0]	35	Vertical Enable Signal Start Position for Full Screen. (VSYNC + V backporch lines) Note: The set value should be four or more (lines). OIR_RES_F_VS[10:0] + OIR_RES_F_VW[10:0] should be equal to or less than 2039 (lines).
OIR_SCL0_FRC6	OIR_RES_F_VW [10:0]	480	Vertical Enable Signal Width for Full Screen (lines) Note: OIR_RES_F_VS[10:0] + OIR_RES_F_VW[10:0] should be equal to or less than 2039 (lines).
OIR_SCL0_FRC7	OIR_RES_F_HS [10:0]	144	Horizontal Enable Signal Start Position for Full Screen. (HSYNC + H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). OIR_RES_F_HS[10:0] + OIR_RES_F_HW[10:0] should be equal to or less than 2015 (clock cycles).
OIR_SCL0_FRC7	OIR_RES_F_HW [10:0]	640	Horizontal Enable Signal Width for Full Screen (pixel-clock cycles) Note: OIR_RES_F_HS[10:0] + OIR_RES_F_HW[10:0] should be equal to or less than 2015 (clock cycles).

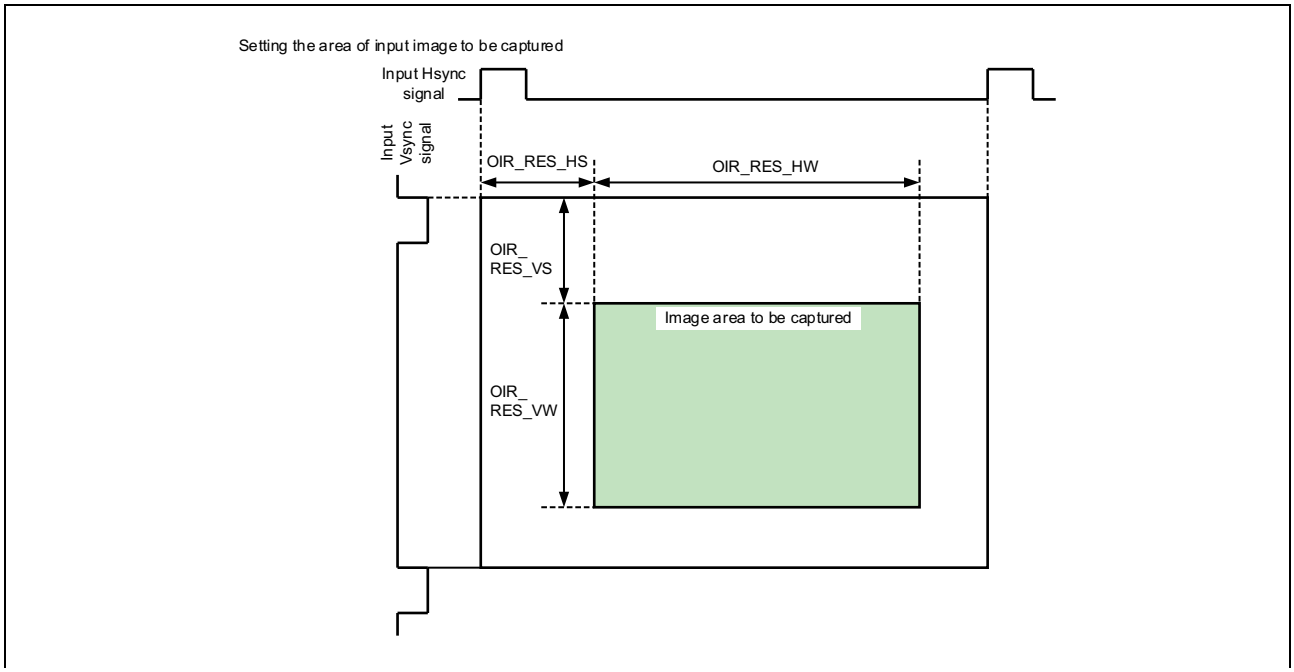


Figure 38.58 Enable Settings

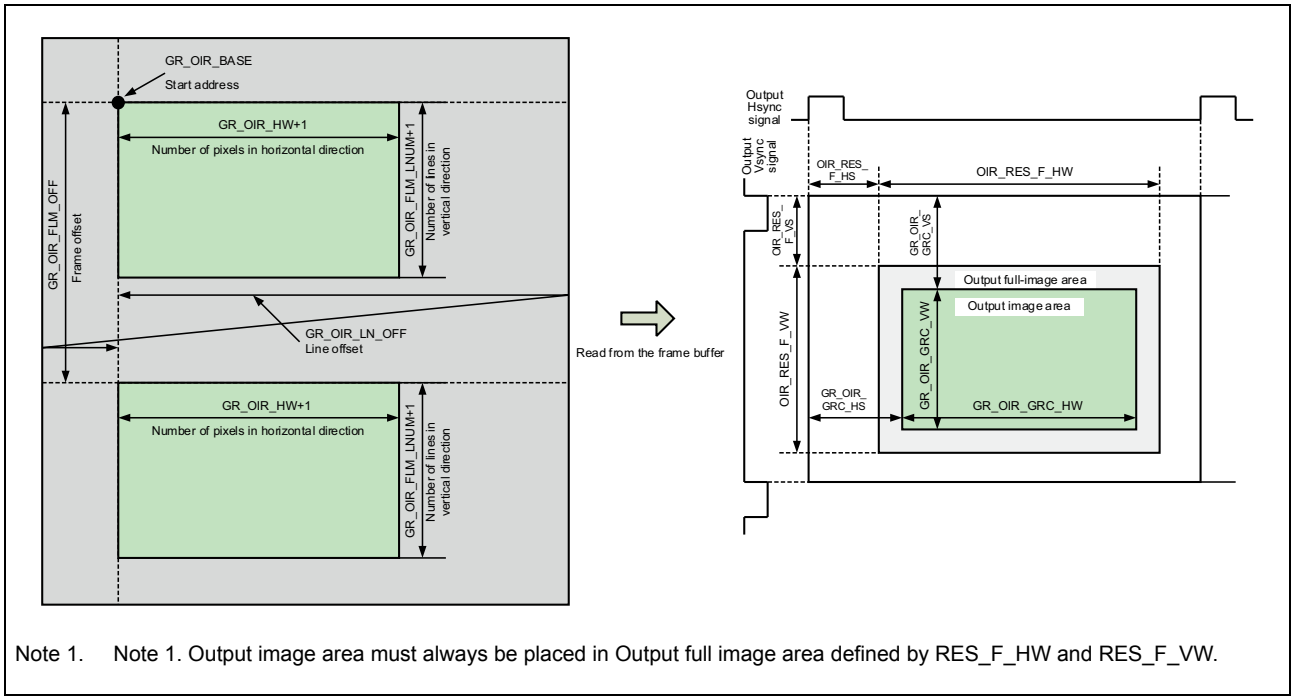
38.7.1.6 Graphics (OIR) Processing

Scale-up processing is not possible in the generator.

Table 38.138 Graphics (OIR) Processing Block

Type of Output Image Generation Display	Sync Signals for Frame Buffer Read	Frame Buffer Read Size Setting Bits	Display Enabling Bits
Graphics display	Output from graphics (OIR) processing block	GR_OIR_FLM_LNUM[10:0]*1 GR_OIR_HW[10:0]*1	GR_OIR_GRC_VS[10:0] GR_OIR_GRC_VW[10:0] GR_OIR_GRC_HS[10:0] GR_OIR_GRC_HW[10:0]

Note 1. The value set to the register + 1 is the actual read size.



Note 1. Note 1. Output image area must always be placed in Output full image area defined by RES_F_HW and RES_F_VW.

Figure 38.59 Area Setting for Graphics Display

For details on the graphics processing, refer to the Section 38.6, Image Synthesizer.

38.7.2 Register Descriptions

Table 38.139 shows the register configuration.

- Symbols used in Register Description:

  Initial value: Register value after a reset

  —: Undefined value

  R/W: Readable/writable. The written value can be read.

  R/WC0: Readable/writable. Writing 0 initializes the bit. Writing 1 is ignored.

  R/WC1: Readable/writable. Writing 1 initializes the bit. Writing 0 is ignored.

  R: Read-only. The write value should always be 0.

  —/W: Write-only. The read value is undefined.

Table 38.139 Register Configuration of the Output Image Generator (Ch0) (1/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	SCL0 register update control register (OIR)	OIR_SCL0_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1E80	32
VDCE0	Mask control register (OIR)	OIR_SCL0_FRC1	R/W	H'0AF0 0001	<VDCE0_base> + 1E84	32
VDCE0	Missing Vsync compensation control register (OIR)	OIR_SCL0_FRC2	R/W	H'0E10 0001	<VDCE0_base> + 1E88	32
VDCE0	Output sync select register (OIR)	OIR_SCL0_FRC3	R/W	H'0000 0001	<VDCE0_base> + 1E8C	32
VDCE0	Free-running period control register (OIR)	OIR_SCL0_FRC4	R/W	H'020C 031F	<VDCE0_base> + 1E90	32
VDCE0	Output delay control register (OIR)	OIR_SCL0_FRC5	R/W	H'0000 0101	<VDCE0_base> + 1E94	32
VDCE0	Full-screen vertical size register (OIR)	OIR_SCL0_FRC6	R/W	H'0023 01E0	<VDCE0_base> + 1E98	32
VDCE0	Full-screen horizontal size register (OIR)	OIR_SCL0_FRC7	R/W	H'0090 0280	<VDCE0_base> + 1E9C	32
VDCE0	Scaling-down control register (OIR)	OIR_SCL0_DS1	R/W	H'0000 0011	<VDCE0_base> + 1EAC	32
VDCE0	Vertical capture size register (OIR)	OIR_SCL0_DS2	R/W	H'0012 00F0	<VDCE0_base> + 1EB0	32
VDCE0	Horizontal capture size register (OIR)	OIR_SCL0_DS3	R/W	H'00F4 05A0	<VDCE0_base> + 1EB4	32
VDCE0	Capture control block output size register (OIR)	OIR_SCL0_DS7	R/W	H'00F0 0280	<VDCE0_base> + 1EC4	32
VDCE0	Scaling-up control register (OIR)	OIR_SCL0_US1	R/W	H'0000 0011	<VDCE0_base> + 1EC8	32
VDCE0	Output image vertical size register (OIR)	OIR_SCL0_US2	R/W	H'0023 01E0	<VDCE0_base> + 1ECC	32
VDCE0	Output image horizontal size register (OIR)	OIR_SCL0_US3	R/W	H'0090 0280	<VDCE0_base> + 1ED0	32
VDCE0	Frame buffer read select register (OIR)	OIR_SCL0_US8	R/W	H'0000 0000	<VDCE0_base> + 1EE4	32
VDCE0	SCL1 register update control register (OIR)	OIR_SCL1_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1F00	32
VDCE0	Writing mode register (OIR)	OIR_SCL1_WR1	R/W	H'0000 0000	<VDCE0_base> + 1F08	32

Table 38.139 Register Configuration of the Output Image Generator (Ch0) (2/2)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Frame sub-sampling register (OIR)	OIR_SCL1_WR5	R/W	H'0000 1000	<VDCE0_base> + 1F1C	32
VDCE0	Graphics (OIR) register update control register	GR_OIR_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1F80	32
VDCE0	Frame buffer read control register (Graphics (OIR))	GR_OIR_FLM_RD	R/W	H'0000 0000	<VDCE0_base> + 1F84	32
VDCE0	Frame buffer control register 1 (Graphics (OIR))	GR_OIR_FLM1	R/W	H'0000 0000	<VDCE0_base> + 1F88	32
VDCE0	Frame buffer control register 2 (Graphics (OIR))	GR_OIR_FLM2	R/W	H'0000 0000	<VDCE0_base> + 1F8C	32
VDCE0	Frame buffer control register 3 (Graphics (OIR))	GR_OIR_FLM3	R/W	H'0800 0001	<VDCE0_base> + 1F90	32
VDCE0	Frame buffer control register 4 (Graphics (OIR))	GR_OIR_FLM4	R/W	H'0008 0000	<VDCE0_base> + 1F94	32
VDCE0	Frame buffer control register 5 (Graphics (OIR))	GR_OIR_FLM5	R/W	H'0000 03FF	<VDCE0_base> + 1F98	32
VDCE0	Frame buffer control register 6 (Graphics (OIR))	GR_OIR_FLM6	R/W	H'8000 0000	<VDCE0_base> + 1F9C	32
VDCE0	Alpha blending control register 1 (Graphics (OIR))	GR_OIR_AB1	R/W	H'0000 0000	<VDCE0_base> + 1FA0	32
VDCE0	Alpha blending control register 2 (Graphics (OIR))	GR_OIR_AB2	R/W	H'0000 0000	<VDCE0_base> + 1FA4	32
VDCE0	Alpha blending control register 3 (Graphics (OIR))	GR_OIR_AB3	R/W	H'0000 0000	<VDCE0_base> + 1FA8	32
VDCE0	Alpha blending control register 7 (Graphics (OIR))	GR_OIR_AB7	R/W	H'00FF 0000	<VDCE0_base> + 1FB8	32
VDCE0	Alpha blending control register 8 (Graphics (OIR))	GR_OIR_AB8	R/W	H'0000 0000	<VDCE0_base> + 1FBC	32
VDCE0	Alpha blending control register 9 (Graphics (OIR))	GR_OIR_AB9	R/W	H'0000 0000	<VDCE0_base> + 1FC0	32
VDCE0	Background color control register (Graphics (OIR))	GR_OIR_BASE	R/W	H'0000 8080	<VDCE0_base> + 1FCC	32
VDCE0	CLUT table control register (Graphics (OIR))	GR_OIR_CLUT	R/W	H'0000 0000	<VDCE0_base> + 1FD0	32
VDCE0	Status monitor register (Graphics (OIR))	GR_OIR_MON	R	H'0000 0000	<VDCE0_base> + 1FD4	32

NOTE

Register access sizes other than defined in the table above are not supported.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

38.7.2.1 SCL0 Register Update Control Register (OIR_SCL0_UPDATE)

NOTE

This register is not available in D1L2 devices.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	OIR_SCL0_VEN_D	OIR_SCL0_VEN_C	—	—	—	OIR_SCL0_UPDATE	—	—	—	OIR_SCL0_VEN_B	—	—	—	OIR_SCL0_VEN_A
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/WC1	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13	OIR_SCL0_VEN_D	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
12	OIR_SCL0_VEN_C	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	OIR_SCL0_UPDATE	0	R/WC1	SYNC Control Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	OIR_SCL0_VEN_B	0	R/WC1	Synchronization Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	OIR_SCL0_VEN_A	0	R/WC1	Capture Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.7.2.2 Mask Control Register (OIR_SCL0_FRC1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OIR_RES_VMASK[15:0]															
Initial value:	0	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OIR_RES_VMASK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	OIR_RES_VMASK[15:0]	2800	R/W	Repeated Vsync Signal Masking Period This setting is ignored.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	OIR_RES_VMASK_ON	1	R/W	Repeated Vsync Signal Masking Control This bit should always be set to 0 when the output image generator is in use. 0: Repeated Vsync signal masking control is disabled. 1: Repeated Vsync signal masking control is enabled.

Note: This register is updated when the OIR_SCL0_UPDATE bit in the OIR_SCL0 register update control register (OIR_SCL0_UPDATE) is 1.

38.7.2.3 Missing Vsync Compensation Control Register (OIR_SCL0_FRC2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OIR_RES_VLACK[15:0]															
Initial value:	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OIR_RES_VLACK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	OIR_RES_VLACK[15:0]	3600	R/W	Missing-Sync Compensating Pulse Output Wait Time This setting is ignored.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	OIR_RES_VLACK_ON	1	R/W	Missing Vsync Signal Compensation This bit should always be set to 0 when the output image generator is in use. 0: Compensation of missing Vsync signals is disabled. 1: Compensation of missing Vsync signals is enabled.

Note: This register is updated when the OIR_SCL0_UPDATE bit in the OIR_SCL0 register update control register (OIR_SCL0_UPDATE) is 1.

38.7.2.4 Output Sync Select Register (OIR_SCL0_FRC3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OIR_RES_EN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OIR_RES_VS_SEL
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	OIR_RES_EN	0	R/W	Enabling or Disabling the Output Image Generator 0: The input signal from the image synthesizer is directly output. 1: Signals from the graphics block (OIR) are output.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	OIR_RES_VS_SEL	1	R/W	Vsync Signal Output Select 0: Vsync signal input from the image synthesizer 1: Internally generated free-running Vsync signal

Note: This register is updated when the OIR_SCL0_UPDATE bit in the OIR_SCL0 register update control register (OIR_SCL0_UPDATE) is 1.

38.7.2.5 Free-Running Period Control Register (OIR_SCL0_FRC4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	OIR_RES_FV[10:0]										
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	OIR_RES_FH[10:0]										
Initial value:	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	OIR_RES_FV[10:0]	524	R/W	Free-Running Vsync Period Setting Free-running Vsync period = (OIR_RES_FV[10:0] + 1) × horizontal period [usec]
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	OIR_RES_FH[10:0]	799	R/W	Hsync Period Setting Hsync period [usec] = (OIR_RES_FH[10:0] +1) ÷ pixel clock frequency [MHz]

Note: This register is updated when the OIR_SCL0_UPDATE bit in the OIR_SCL0 register update control register (OIR_SCL0_UPDATE) is 1.

38.7.2.6 Output Delay Control Register (OIR_SCL0_FRC5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OIR_RES_VSDLY[7:0]							
Initial value:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	—	1	R	Reserved This bit is always read as 1. The write value should always be 1.
7 to 0	OIR_RES_VSDLY[7:0]	1	R/W	Vsync Signal Delay Control Adjusts the Vsync signal delay in the output Hsync period units. Vsync signal delay [usec]: OIR_RES_VSDLY[7:0] × output Hsync period [usec]

Note: This register is updated when the OIR_SCL0_VEN_B bit in the OIR_SCL0 register update control register (OIR_SCL0_UPDATE) is 1.

38.7.2.7 Full-Screen Vertical Size Register (OIR_SCL0_FRC6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	OIR_RES_F_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	OIR_RES_F_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	OIR_RES_F_VS[10:0]	35	R/W	Vertical Enable Signal Start Position for Full Screen. (VSYNC + V backporch lines) Note: The set value should be four or more (lines). OIR_RES_F_VS[10:0] + OIR_RES_F_VW[10:0] should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	OIR_RES_F_VW[10:0]	480	R/W	Vertical Enable Signal Width for Full Screen (lines) Note: OIR_RES_F_VS[10:0] + OIR_RES_F_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when the OIR_SCL0_VEN_B bit in the OIR_SCL0 register update control register (OIR_SCL0_UPDATE) is 1.

38.7.2.8 Full-Screen Horizontal Size Register (OIR_SCL0_FRC7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	OIR_RES_F_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	OIR_RES_F_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	OIR_RES_F_HS[10:0]	144	R/W	Horizontal Enable Signal Start Position for Full Screen. (HSYNC+H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). OIR_RES_F_HS[10:0] + OIR_RES_F_HW[10:0] should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	OIR_RES_F_HW[10:0]	640	R/W	Horizontal Enable Signal Width for Full Screen (pixel-clock cycles) **Note:** 1. OIR_RES_F_HS[10:0] + OIR_RES_F_HW[10:0] should be equal to or less than 2015 (clock cycles). 2. In the case of video output format is Serial RGB(OUT_SET.OUT_FORMAT[[1:0] = 3), it is nessesary to add 2 to value in OIR_RES_F_HW.

Note: This register is updated when the OIR_SCL0_VEN_B bit in the OIR_SCL0 register update control register (OIR_SCL0_UPDATE) is 1.

38.7.2.9 Scaling-Down Control Register (OIR_SCL0_DS1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	Caution *	—	—	—	Caution *
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

CAUTION

The initial value “1” of bits 4 and 0 must be changed to “0”.

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	Bit4	1	R/W	**CAUTION** The initial value “1” of this bit must be changed to “0”.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	Bit0	1	R/W	**CAUTION** The initial value “1” of this bit must be changed to “0”.

38.7.2.10 Vertical Capture Size Register (OIR_SCL0_DS2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	OIR_RES_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	OIR_RES_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	OIR_RES_VS [10:0]	18	R/W	Vertical Position Setting for Video Signal Capturing (VSYNC + (V backporch - 1) lines) Note: The set value should be four or more (lines). OIR_RES_VS[10:0] + OIR_RES_VW[10:0] should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	OIR_RES_VW [10:0]	240	R/W	Vertical Width of Video Signal to be Captured (Lines) Note: OIR_RES_VS[10:0] + OIR_RES_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when the OIR_SCL0_VEN_A bit in the OIR_SCL0 register update control register (OIR_SCL0_UPDATE) is 1.

38.7.2.11 Horizontal Capture Size Register (OIR_SCL0_DS3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	OIR_RES_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	OIR_RES_HW[10:0]										
Initial value:	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	OIR_RES_HS [10:0]	244	R/W	Horizontal Position Setting for Video Signal Capturing (HSYNC + H backporch video-image clock cycles) Note: The set value should be 16 or more (clock cycles). OIR_RES_HS[10:0] + OIR_RES_HW[10:0] should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	OIR_RES_HW [10:0]	1440	R/W	Horizontal Width of Video Signal to be Captured (Video-image clock cycles) Note: OIR_RES_HS[10:0] + OIR_RES_HW[10:0] should be equal to or less than 2015 (clock cycles).

Note: This register is updated when the OIR_SCL0_VEN_A bit in the OIR_SCL0 register update control register (OIR_SCL0_UPDATE) is 1.

38.7.2.12 Scaling-Down Control Block Output Size Register (OIR_SCL0_DS7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	OIR_RES_OUT_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	OIR_RES_OUT_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	OIR_RES_OUT_VW[10:0]	240	R/W	Number of Valid Lines in Vertical Direction Output by Scaling-down Control Block (lines) This bit setting is used for the number of lines to be written to VOWE.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	OIR_RES_OUT_HW[10:0]	640	R/W	Number of Valid Horizontal Pixels Output by Scaling-Down Control Block (video-image clock cycles) Note: The OIR_RES_OUT_HW[10:0] value should be aligned in 4-pixel units and equal to or smaller than the OIR_RES_HW[10:0] value.

Note: This register is updated when the OIR_SCL0_VEN_A or OIR_SCL0_VEN_C bit in the OIR_SCL0 register update control register (OIR_SCL0_UPDATE) is 1.

38.7.2.13 Scaling-Up Control Register (OIR_SCL0_US1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	Caution *	—	—	—	Caution *
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R/W

CAUTION

The initial value “1” of bits 4 and 0 must be changed to “0”.

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	Bit4	1	R/W	**CAUTION** The initial value “1” of this bit must be changed to “0”.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	Bit0	1	R/W	**CAUTION** The initial value “1” of this bit must be changed to “0”.

38.7.2.14 Output Image Vertical Size Register (OIR_SCL0_US2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	OIR_RES_P_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	OIR_RES_P_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	OIR_RES_P_VS[10:0]	35	R/W	Vertical Enable Signal Start Position for Output Image (VSYNC + V backporch lines) Note: The set value should be four or more (lines). OIR_RES_P_VS[10:0] + OIR_RES_P_VW[10:0] should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	OIR_RES_P_VW[10:0]	480	R/W	Vertical Enable Signal Width for Output Image (lines) Note: OIR_RES_P_VS[10:0] + OIR_RES_P_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when the OIR_SCL0_VEN_B bit in the OIR_SCL0 register update control register (OIR_SCL0_UPDATE) is 1.

38.7.2.15 Output Image Horizontal Size Register (OIR_SCL0_US3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	OIR_RES_P_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	OIR_RES_P_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	OIR_RES_P_HS[10:0]	144	R/W	Horizontal Enable Signal Start Position for Output Image (HSYNC+H backporch pixel-clock cycles) Note: The set value should be 16 or more (clock cycles). OIR_RES_P_HS[10:0] + OIR_RES_P_HW[10:0] should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	OIR_RES_P_HW[10:0]	640	R/W	Horizontal Enable Signal Width for Output Image (pixel-clock cycles) Note: OIR_RES_P_HS[10:0] + OIR_RES_P_HW[10:0] should be equal to or less than 2015 (clock cycles).

Note: This register is updated when the OIR_SCL0_VEN_B bit in the OIR_SCL0 register update control register (OIR_SCL0_UPDATE) is 1.

38.7.2.16 Frame Buffer Read Select Register (OIR_SCL0_US8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	Caution *	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R

CAUTION

The initial value “0” of bit 4 must be changed to “1”.

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	Bit4	0	R/W	**CAUTION** The initial value “0” of this bit must be changed to “1”.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

38.7.2.17 SCL1 Register Update Control Register (OIR_SCL1_UPDATE)

NOTE

This register is not available in D1L2 devices.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OIR_SCL1_UPDATE_A
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	OIR_SCL1_VEN_B	—	—	—	OIR_SCL1_VEN_A
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	OIR_SCL1_UPDATE_A	0	R/W	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated.
15 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	OIR_SCL1_VEN_B	0	R/WC1	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	OIR_SCL1_VEN_A	0	R/WC1	Frame Buffer Write Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.7.2.18 Writing Mode Register (OIR_SCL1_WR1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	Caution*		—	Caution*
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W

CAUTIONS

1. **The initial value "00_B" of bits [3:2] must be changed to "10_B".**
2. **The initial value "0" of bit 0 must be changed to "1".**

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 4	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
3, 2	Bit[3:2]	All 0	R/W	**CAUTION** The initial value "00_B" of these bits must be changed to "10_B".
1	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
0	Bit0	0	R/W	**CAUTION** The initial value "0" of this bit must be changed to "1".

38.7.2.19 Frame Sub-Sampling Register (OIR_SCL1_WR5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	Caution *	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R/W	R/W	R	R	R	R	R	R	R	R/W

CAUTION

The initial value “1” of bit 12 must be changed to “0”.

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	Bit12	1	R/W	**CAUTION** The initial value “1” of this bit must be changed to “0”.
11 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

38.7.2.20 Graphics (OIR) Register Update Control Register (GR_OIR_UPDATE)

NOTE

This register is not available in D1L2 devices.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	GR_OIR_UPDATE	—	—	—	GR_OIR_P_VEN	—	—	—	GR_OIR_IBUS_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/WC1	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR_OIR_UPDATE	0	R/W	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR_OIR_P_VEN	0	R/WC1	Graphics Display Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR_OIR_IBUS_VEN	0	R/WC1	Frame Buffer Read Control Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.7.2.21 Frame Buffer Read Control Register (Graphics (OIR)) (GR_OIR_FLM_RD)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR_OIR_R_ENB
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR_OIR_R_ENB	0	R/W	Frame Buffer Read Enable 0: Frame buffer reading is disabled. 1: Frame buffer reading is enabled.

Note: This register is updated when the GR_OIR_IBUS_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.22 Frame Buffer Control Register 1 (Graphics (OIR)) (GR_OIR_FLM1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	Caution*		—	—	—	—	—	—	—	Caution*
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W

CAUTIONS

1. **The initial value "00_B" of bits [9:8] must be changed to "10_B".**
2. **The initial value "0" of bit 0 must be changed to "1".**

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	Bit[9:8]	0	R/W	**CAUTION** The initial value "00_B" of these bits must be changed to "10_B".
7 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	Bit0	0	R/W	**CAUTION** The initial value "0" of this bit must be changed to "1".

NOTES

1. Bit[9:8] is updated when the GR_OIR_IBUS_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.
   Bit0 is updated when the GR_OIR_IBUS_VEN and GR_OIR_P_VEN bits in the graphics (OIR) register update control register (GR_OIR_UPDATE) are 1.
2. Check restriction frame buffer addressing in Section 38.6.1.2, Graphics Data Read Control.

38.7.2.23 Frame Buffer Control Register 2 (Graphics (OIR)) (GR_OIR_FLM2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR_OIR_BASE[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR_OIR_BASE[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 0	GR_OIR_BASE [31:0]	0	R/W	Frame Buffer Base Address Sets the start address of the frame buffer where frame data is to be stored. The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when the GR_OIR_IBUS_VEN or GR_OIR_P_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.24 Frame Buffer Control Register 3 (Graphics (OIR)) (GR_OIR_FLM3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	GR_OIR_LN_OFF[14:0]														
Initial value:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
30 to 16	GR_OIR_LN_OFF [14:0]	2048	R/W	Frame Buffer Line Offset Address Sets the line offset address for calculating the start address of each line. Line 0: GR_OIR_BASE[31:0] Line 1: GR_OIR_BASE[31:0] + GR_OIR_LN_OFF[14:0] × 1 : Line n: GR_OIR_BASE[31:0] + GR_OIR_LN_OFF[14:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note: This register is updated when the GR_OIR_IBUS_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.25 Frame Buffer Control Register 4 (Graphics (OIR)) (GR_OIR_FLM4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	GR_OIR_FLM_OFF[22:16]						
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR_OIR_FLM_OFF[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 23	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
22 to 0	GR_OIR_FLM_OFF[22:0]	524288	R/W	Frame Buffer Frame Offset Address Specifies the frame offset address used for calculating the start address of each frame buffer when more than one buffer is used. Buffer 0: GR_OIR_BASE[31:0] Buffer 1: GR_OIR_BASE[31:0] + GR_OIR_FLM_OFF[22:0] × 1 : Buffer n: GR_OIR_BASE[31:0] + GR_OIR_FLM_OFF[22:0] × n The address must be 128 byte aligned, so the lower 7 bits must be set to 0.

Note: This register is updated when the GR_OIR_IBUS_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.26 Frame Buffer Control Register 5 (Graphics (OIR)) (GR_OIR_FLM5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR_OIR_FLM_LNUM[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR_OIR_FLM_LOOP[10:0]										
Initial value:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR_OIR_FLM_LNUM[10:0]	0	R/W	Sets number of lines in a frame Number of lines is (GR_OIR_FLM_LNUM[10:0] + 1).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR_OIR_FLM_LOOP[10:0]	1023	R/W	Number of lines when reading the addresses repeatedly by returning to the start address after reaching the end address. (GR_OIR_FLM_LOOP[10:0] + 1) lines are read.

Note: This register is updated when the GR_OIR_IBUS_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.27 Frame Buffer Control Register 6 (Graphics (OIR)) (GR_OIR_FLM6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR_OIR_FORMAT[3:0]				—	GR_OIR_HW[10:0]										
Initial value:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	GR_OIR_RDSWA[2:0]			—	—	—	—	GR_OIR_STA_POS[5:0]					
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 28	GR_OIR_FORMAT [3:0]	8	R/W	Sets the format of the frame buffer read signal. 0: RGB565 1: RGB888 2: αRGB1555 3: αRGB4444 4: αRGB8888 5: Setting prohibited 6: Setting prohibited 7: Setting prohibited 8: Setting prohibited 9: Setting prohibited 10: RGBα5551 11: RGBα8888 12 to 15: Setting prohibited
27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR_OIR_HW[10:0]	0	R/W	Sets the width of the horizontal valid period. The width is (GR_OIR_HW[10:0] + 1) pixels. Note: The set value should be equal to or more than two.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
12 to 10	GR_OIR_RDSWA [2:0]	0	R/W	8-Bit, 16-Bit, or 32-Bit Swap Setting These bits control swapping in frame buffer writing as follows. Bit 0 0: Swapped in 8-bit units. 1: Not swapped in 8-bit units. Bit 1 0: Swapped in 16-bit units. 1: Not swapped in 16-bit units. Bit 2 0: Swapped in 32-bit units. 1: Not swapped in 32-bit units. According to the setting of these bits, data is swapped as follows. Each number in parentheses ((1) to (8)) indicates 8-bit data. 000: (1) (2) (3) (4) (5) (6) (7) (8) [Not swapped] 001: (2) (1) (4) (3) (6) (5) (8) (7) [Swapped in 8-bit units] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Swapped in 16-bit units] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Swapped in 16-bit units + 8-bit units] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Swapped in 32-bit units] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Swapped in 32-bit units + 8-bit units] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Swapped in 32-bit units + 16-bit units] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Swapped in 32-bit units + 16-bit units + 8-bit units]
9 to 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5 to 0	GR_OIR_STA_POS [5:0]	0	R/W	Sets the amount of data to be skipped through. Specifically data amount equal to the amount indicated by GR_OIR_STA_POS[5:0] is skipped from the start of the line.

Note: GR_OIR_STA_POS[5:0] is updated when GR_OIR_P_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.
GR_OIR_RDSWA[2:0] is updated when GR_OIR_UPDATE bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.
GR_OIR_FORMAT[3:0] and GR_OIR_HW[10:0] are updated when GR_OIR_IBUS_VEN or GR_OIR_P_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.28 Alpha Blending Control Register 1 (Graphics (OIR)) (GR_OIR_AB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	GR_OIR_GRC_DISP_ON	—	—	GR_OIR_DISP_SEL[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR_OIR_GRC_DISP_ON	0	R/W	Turns on/off frame-line display of the graphics image area. 0: Frame-line display off 1: Frame-line display on **NOTE** GR_OIR_GRC_DISP_ON is not available in D1L2 devices.
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	GR_OIR_DISP_SEL[1:0]	0	R/W	Selects the graphics display mode. 0: Background color display (GR_OIR_BASE[31:0]) 1: Setting prohibited 2: Current graphics display When displaying graphics, select this setting. 3: Blended display of lower-layer graphics and current graphics* Note: * Select this setting whenever chroma-key processing is to proceed. Since only current graphics are to be displayed by chroma-key processing, set the α values for both pixels to be subject to chroma-keying and pixels not to be subject to chroma-keying to 255.

Note: This register is updated when GR_OIR_P_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.29 Alpha Blending Control Register 2 (Graphics (OIR)) (GR_OIR_AB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR_OIR_GRC_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR_OIR_GRC_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR_OIR_GRC_VS[10:0]	0	R/W	Vertical Start Position of Graphics Image Area. **NOTE** The set value should be four or more (lines). GR_OIR_GRC_VS[10:0] + GR_OIR_GRC_VW[10:0] should be equal to or less than 2039 (lines).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR_OIR_GRC_VW[10:0]	0	R/W	Vertical Width of Graphics Image Area. **NOTE** GR_OIR_GRC_VS[10:0] + GR_OIR_GRC_VW[10:0] should be equal to or less than 2039 (lines).

Note: This register is updated when the GR_OIR_P_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.30 Alpha Blending Control Register 3 (Graphics (OIR)) (GR_OIR_AB3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GR_OIR_GRC_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR_OIR_GRC_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	GR_OIR_GRC_HS[10:0]	0	R/W	Horizontal Start Position of Graphics Image Area. **NOTE** The set value should be 16 or more (clock cycles). GR_OIR_GRC_HS[10:0] + GR_OIR_GRC_HW[10:0] should be equal to or less than 2015 (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR_OIR_GRC_HW[10:0]	0	R/W	Horizontal Width of Graphics Image Area. **NOTES** 1. For displaying an image with 1- or 2-pixel horizontal width, set GR_OIR_HW[10:0] to 2 and GR_OIR_GRC_HW[10:0] to 1 (1 pixel) or 2 (2 pixels). 2. GR_OIR_GRC_HS[10:0] + GR_OIR_GRC_HW[10:0] should be equal to or less than 2015 (clock cycles).

Note: This register is updated when the GR_OIR_P_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.31 Alpha Blending Control Register 7 (Graphics (OIR)) (GR_OIR_AB7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GR_OIR_CK_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR_OIR_CK_ON	0	R/W	RGB-Index Chroma-Key Processing On/Off 0: Off 1: On

Note: This register is updated when the GR_OIR_P_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.32 Alpha Blending Control Register 8 (Graphics (OIR)) (GR_OIR_AB8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR_OIR_CK_KG[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR_OIR_CK_KB[7:0]								GR_OIR_CK_KR[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR_OIR_CK_KG[7:0]	0	R/W	G Signal for RGB-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR_OIR_CK_KB[7:0]	0	R/W	B Signal for RGB-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR_OIR_CK_KR[7:0]	0	R/W	R Signal for RGB-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when the GR_OIR_P_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.33 Alpha Blending Control Register 9 (Graphics (OIR)) (GR_OIR_AB9)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GR_OIR_CK_A[7:0]								GR_OIR_CK_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR_OIR_CK_B[7:0]								GR_OIR_CK_R[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GR_OIR_ CK_A[7:0]	0	R/W	Replaced Alpha Signal after RGB-Index Chroma-Key Processing α: Unsigned 8 bits (0 to 255 [LSB]) Note: These bits should always be set to 255 to display the current graphics only.
23 to 16	GR_OIR_ CK_G[7:0]	0	R/W	Replaced G Signal after RGB-Index Chroma-Key Processing G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR_OIR_ CK_B[7:0]	0	R/W	Replaced B Signal after RGB-Index Chroma-Key Processing B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR_OIR_ CK_R[7:0]	0	R/W	Replaced R Signal after RGB-Index Chroma-Key Processing R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when the GR_OIR_P_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.34 Background Color Control Register (Graphics (OIR)) (GR_OIR_BASE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GR_OIR_BASE_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GR_OIR_BASE_B[7:0]								GR_OIR_BASE_R[7:0]							
Initial value:	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GR_OIR_BASE_G[7:0]	0	R/W	Background Color G Signal G: Unsigned 8 bits (0 to 255 [LSB])
15 to 8	GR_OIR_BASE_B[7:0]	128	R/W	Background Color B Signal B: Unsigned 8 bits (0 to 255 [LSB])
7 to 0	GR_OIR_BASE_R[7:0]	128	R/W	Background Color R Signal R: Unsigned 8 bits (0 to 255 [LSB])

Note: This register is updated when the GR_OIR_P_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.35 CLUT Table Control Register (Graphics (OIR)) (GR_OIR_CLUT)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GR_OIR_LINE[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	GR_OIR_LINE[10:0]	0	R/W	Line interrupt Setting Outputs an interrupt signal when the number of lines matches with the value of GR_OIR_LINE[10:0]. This function is enabled only when the output image generator is enabled (OIR_SCL0_FRC3.OIR_RES_EN = 1).

Note: This register is updated when the GR_OIR_P_VEN bit in the graphics (OIR) register update control register (GR_OIR_UPDATE) is 1.

38.7.2.36 Status Monitor Register (GR_OIR_MON)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	OIR_LIN_STAT[10:0]										
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial Value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	OIR_LIN_STAT[10:0]*1	All 0	R	Line Position of Image Currently Being Read
15 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Note 1. OIR_LIN_STAT[10:0] changes immediately, when the video output clock is enabled.

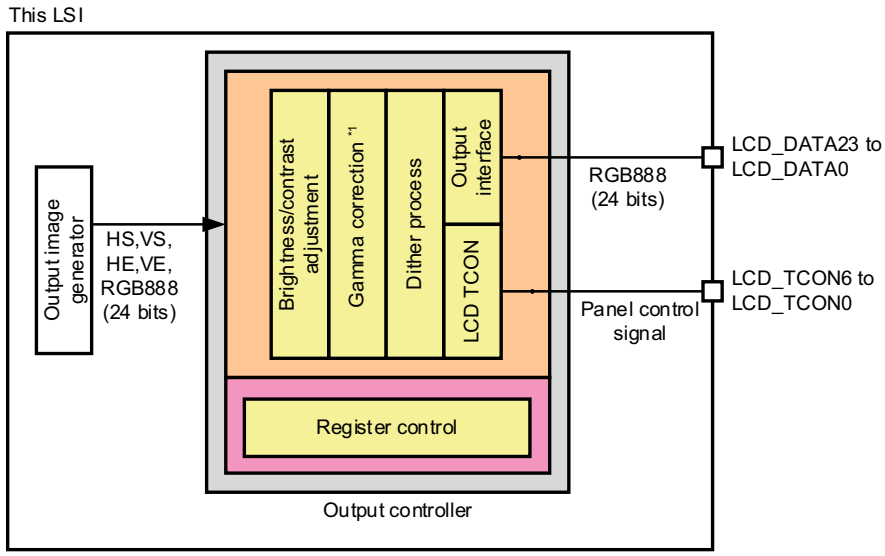
38.8 Output Controller

38.8.1 Output Controller Functions

38.8.1.1 Overview of Functions

The output controller subjects RGB signals output from the output image generator to brightness adjustment, contrast adjustment, gamma correction (not for D1L2(H) devices) of individual RGB, dither process, and output format conversion. The output controller also generates various timing signals for LCD panel drive.

Figure 38.60 shows the function block diagram of the output controller.



Note 1. Gamma correction is not available for D1L2(H) devices.

Figure 38.60 Functional Block Diagram of Output Controller

38.8.1.2 Register Update Control

The Vsync signal is used to control the update timing of all the registers of the output controller.

After 1 is set to the bits in the update control register, the contents of the relevant registers are actually modified at the rising edge of the Vsync signal, when the update control register is automatically cleared to 0.

Table 38.140 Register Update Control (1/2)

Register Name	Bit Name	Initial Value	Description
OUT_UPDATE	OUTCNT_VEN	0	Brightness/Contrast Control, Dither Process, Output Interface Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GAM_G_UPDATE	GAM_G_VEN	0	Gamma Correction (G) Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

Table 38.140 Register Update Control (2/2)

Register Name	Bit Name	Initial Value	Description
GAM_B_UPDATE	GAM_B_VEN	0	Gamma Correction (B) Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
GAM_R_UPDATE	GAM_R_VEN	0	Gamma Correction (R) Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.
TCON_UPDATE	TCON_VEN	0	LCD TCON Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.8.1.3 Route Selection

The processing sequence of the brightness/contrast control and gamma correction control can be swapped according to the settings of the register.

Table 38.141 Route Selection

Register Name	Bit Name	Initial Value	Description
OUT_CLK_PHASE	OUTCNT_ FRONT_GAM	0	Correction Circuit Sequence Control 0: Brightness → contrast → gamma correction 1: Gamma correction → contrast → brightness

38.8.1.4 Panel Brightness Adjustment

Brightness (DC) adjustment is individually performed for RGB signals from output image generator.

(BRT_R/G/BOUT after brightness adjustment has many bits to prevent overflow or underflow. The overflow or underflow process is performed at contrast calculation.)

(1) Calculation formulas for brightness (DC) adjustment

$$BRT\_GOUT = GIN + PBRT\_G - 512$$

$$BRT\_BOUT = BIN + PBRT\_B - 512$$

$$BRT\_ROUT = RIN + PBRT\_R - 512$$

Table 38.142 Brightness (DC) Adjustment

Register Name	Bit Name	Initial Value	Description
OUT_BRIGHT1	PBRT_G[9:0]	512	Brightness (DC) Adjustment of G Signal Unsigned (0 (-512) to 512 (0) to 1023 (+511) [LSB], 512 [LSB] with offset)
OUT_BRIGHT2	PBRT_B[9:0]	512	Brightness (DC) Adjustment of B Signal Unsigned (0 (-512) to 512 (0) to 1023 (+511) [LSB], 512 [LSB] with offset)
OUT_BRIGHT2	PBRT_R[9:0]	512	Brightness (DC) Adjustment of R Signal Unsigned (0 (-512) to 512 (0) to 1023 (+511) [LSB], 512 [LSB] with offset)

38.8.1.5 Contrast Adjustment

Contrast is calculated for RGB signals obtained after brightness calculation.

(If an overflow or underflow occurs, contrast is clipped to the maximum or minimum value.)

(1) Calculation formulas for contrast (gain) adjustment

GOUT = BRT_GOUT × CONT_G/128

BOUT = BRT_BOUT × CONT_B/128

ROUT = BRT_ROUT × CONT_R/128

Table 38.143 Contrast (Gain) Adjustment

Register Name	Bit Name	Initial Value	Description
OUT_CONTRAST	CONT_G [7:0]	128	Contrast (Gain) Adjustment of G Signal 0/128 to 255/128 (approx.2 times)
OUT_CONTRAST	CONT_B [7:0]	128	Contrast (Gain) Adjustment of B Signal 0/128 to 255/128 (approx.2 times)
OUT_CONTRAST	CONT_R [7:0]	128	Contrast (Gain) Adjustment of R Signal 0/128 to 255/128 (approx.2 times)

38.8.1.6 Gamma Correction

Gamma correction is carried out by dividing an input signal having 256 gradation levels into 32 and controlling the gain of each area. Gain coefficient of each area can be set as 0 to approx. 2.0 [times]

(1) Gamma correction formula for each area

$DOUT = ((DIN - TH_{(n)}) \times GAIN_{(n)} + OFFSET_{(n)})/256$

DIN: Input signal (8-bit)

DOUT: Output signal (10-bit)

TH(n): Threshold (8-bit)

OFFSET(n): Offset value (19-bit)

GAIN(n): Gain coefficient (11-bit)

(2) Offset calculation formulas for each area

$OFFSET_{(n)} = OFFSET_{(n-1)} + DEF\_O_{(n)}$ (When n = 0, $OFFSET_{(0)} = 0$.)

$DEF\_O_{(n)} = (TH_{(n)} - TH_{(n-1)}) \times GAIN_{(n-1)}$ (When n = 0, $OFFSET_{(0)} = 0$.)

OFFSET(n): Offset value of current area (19-bit)

OFFSET(n-1): Offset value of previous area (19-bit)

DEF_O(n): Difference in offset value of Current and previous area (19-bit)

TH(n): Threshold of current area (8-bit)

TH(n-1): Threshold of previous area (8-bit)

GAIN(n-1): Gain coefficient of previous area (11-bit)

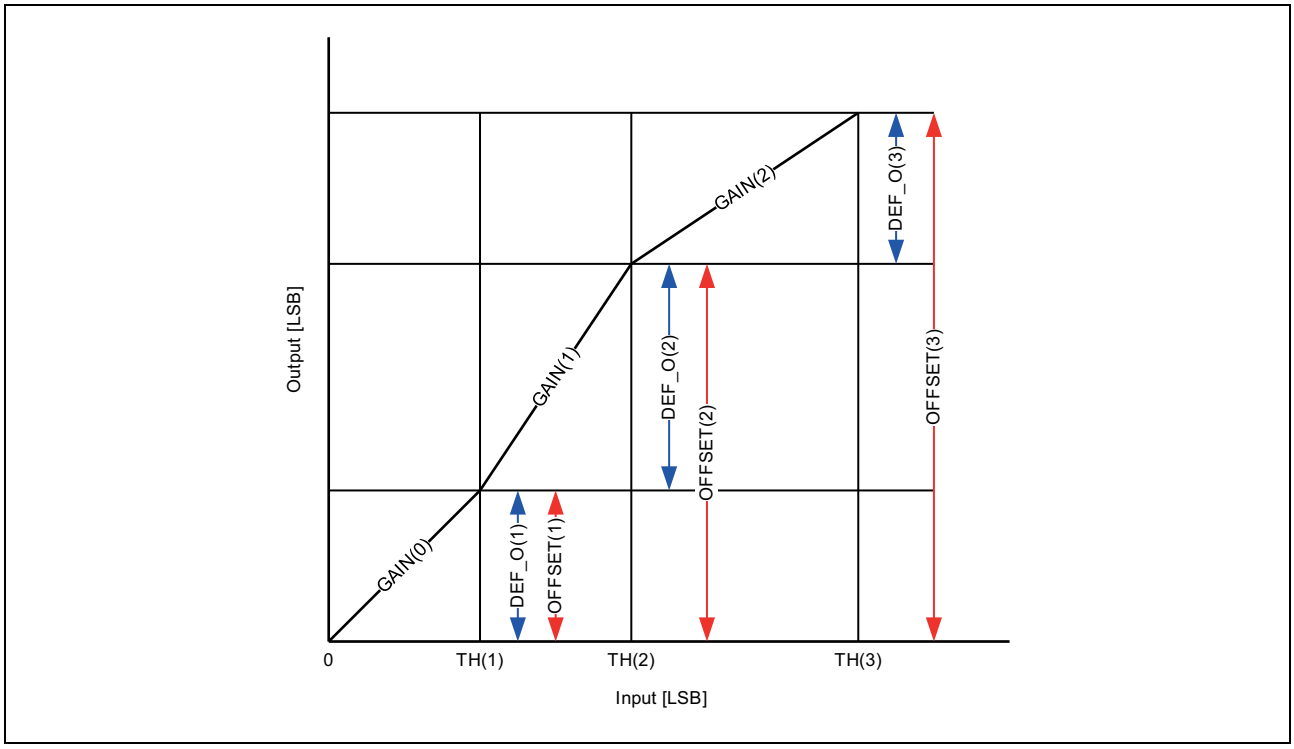


Figure 38.61 Corresponding Chart of Offset Calculation Formulas

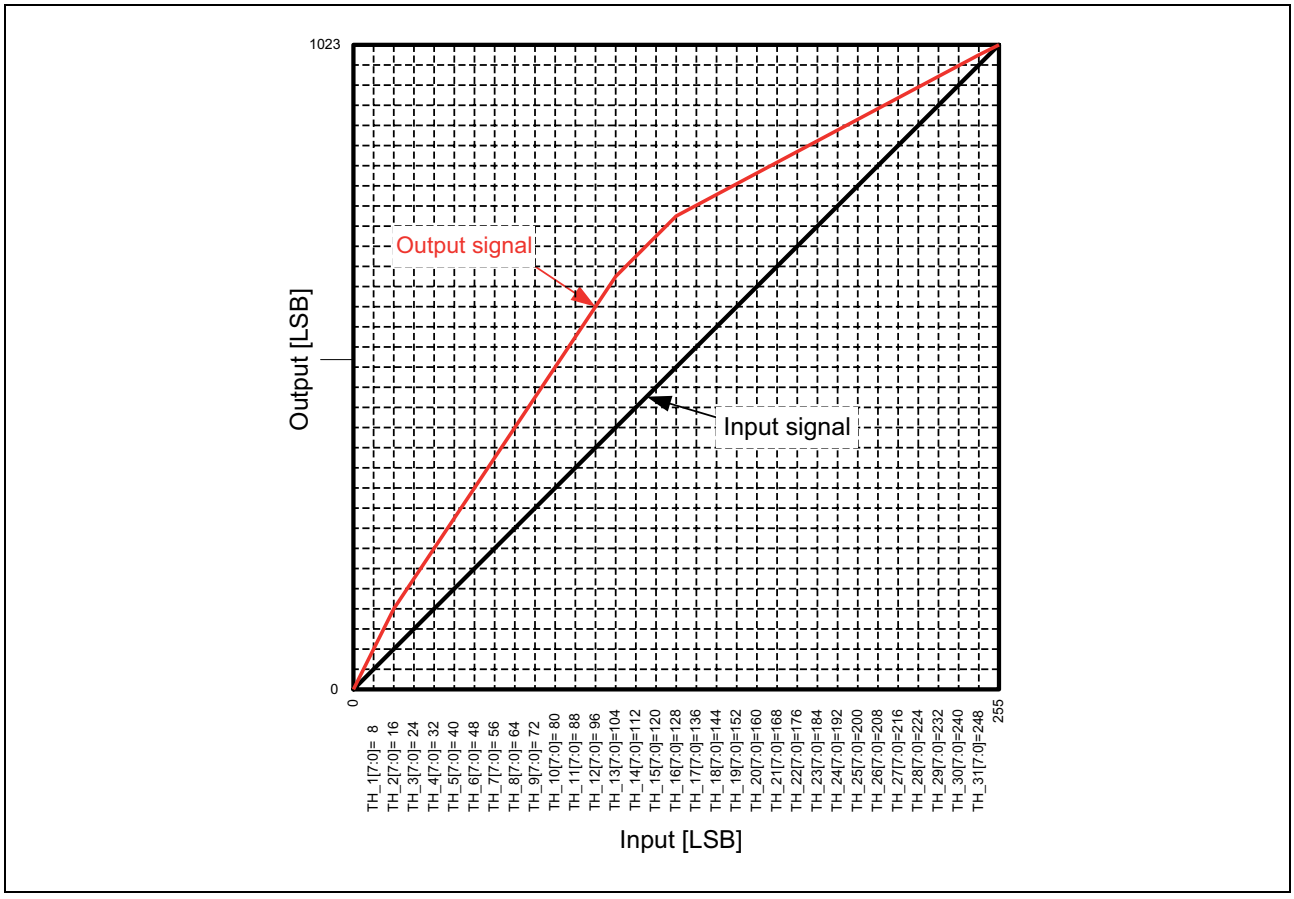


Figure 38.62 Example of Input-Output Characteristics of Gamma Correction

Table 38.144 Gamma Correction (1/2)

Register Name	Bit Name	Initial Value	Description
GAM_SW	GAM_ON	0	Gamma Correction On/Off Control 0: Off 1: On
GAM_G_AREA1 to GAM_G_AREA8	GAM_G_TH_01 to GAM_G_TH_31 [7:0]	*	Start Threshold of Area 1 to 31 of G Signal Unsigned (0 to 255 [LSB]) TH_01: 0 < Threshold of current area < Threshold of next area TH_02-30: Threshold of previous area < Threshold of current area < Threshold of next area TH_31: Threshold of previous area < Threshold of current area ≤ 255 *Initial Value GAM_G_TH_01:8, GAM_G_TH_02:16, GAM_G_TH_03:24, GAM_G_TH_04:32, GAM_G_TH_05:40, GAM_G_TH_06:48, GAM_G_TH_07:56, GAM_G_TH_08:64, GAM_G_TH_09:72, GAM_G_TH_10:80 GAM_G_TH_11:88, GAM_G_TH_12:96, GAM_G_TH_13:104, GAM_G_TH_14:112, GAM_G_TH_15:120, GAM_G_TH_16:128, GAM_G_TH_17:136, GAM_G_TH_18:144, GAM_G_TH_19:152, GAM_G_TH_20:160, GAM_G_TH_21:168, GAM_G_TH_22:176, GAM_G_TH_23:184, GAM_G_TH_24:192, GAM_G_TH_25:200, GAM_G_TH_26:208, GAM_G_TH_27:216, GAM_G_TH_28:224, GAM_G_TH_29:232, GAM_G_TH_30:240, GAM_G_TH_31:248
GAM_G_LUT1 to GAM_G_LUT16	GAM_G_GAIN_00 to GAM_G_GAIN_31 [10:0]	1024	Gain Adjustment of Area 0 to 31 of G Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])
GAM_B_AREA1 to GAM_B_AREA8	GAM_B_TH_01 to GAM_B_TH_31 [7:0]	*	Start Threshold of Area 1 to 31 of B Signal Unsigned (0 to 255 [LSB]) TH_01: 0 < Threshold of current area < Threshold of next area TH_02-30: Threshold of previous area < Threshold of current area < Threshold of next area TH_31: Threshold of previous area < Threshold of current area ≤ 255 *Initial Value GAM_B_TH_01:8, GAM_B_TH_02:16, GAM_B_TH_03:24, GAM_B_TH_04:32, GAM_B_TH_05:40, GAM_B_TH_06:48, GAM_B_TH_07:56, GAM_B_TH_08:64, GAM_B_TH_09:72, GAM_B_TH_10:80 GAM_B_TH_11:88, GAM_B_TH_12:96, GAM_B_TH_13:104, GAM_B_TH_14:112, GAM_B_TH_15:120, GAM_B_TH_16:128, GAM_B_TH_17:136, GAM_B_TH_18:144, GAM_B_TH_19:152, GAM_B_TH_20:160, GAM_B_TH_21:168, GAM_B_TH_22:176, GAM_B_TH_23:184, GAM_B_TH_24:192, GAM_B_TH_25:200, GAM_B_TH_26:208, GAM_B_TH_27:216, GAM_B_TH_28:224, GAM_B_TH_29:232, GAM_B_TH_30:240, GAM_B_TH_31:248

Table 38.144 Gamma Correction (2/2)

Register Name	Bit Name	Initial Value	Description
GAM_B_LUT1 to GAM_B_LUT16	GAM_B_GAIN_00 to GAM_B_GAIN_31 [10:0]	1024	Gain Adjustment of Area 0 to 31 of B Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])
GAM_R_AREA1 to GAM_R_AREA8	GAM_R_TH_01 to GAM_R_TH_31 [7:0]	*	Start Threshold of Area 1 to 31 of R Signal Unsigned (0 to 255 [LSB]) TH_01: 0 < Threshold of current area < Threshold of next area TH_02-30: Threshold of previous area < Threshold of current area < Threshold of next area TH_31: Threshold of previous area < Threshold of current area ≤ 255 *Initial Value GAM_R_TH_01:8, GAM_R_TH_02:16, GAM_R_TH_03:24, GAM_R_TH_04:32, GAM_R_TH_05:40, GAM_R_TH_06:48, GAM_R_TH_07:56, GAM_R_TH_08:64, GAM_R_TH_09:72, GAM_R_TH_10:80 GAM_R_TH_11:88, GAM_R_TH_12:96, GAM_R_TH_13:104, GAM_R_TH_14:112, GAM_R_TH_15:120, GAM_R_TH_16:128, GAM_R_TH_17:136, GAM_R_TH_18:144, GAM_R_TH_19:152, GAM_R_TH_20:160, GAM_R_TH_21:168, GAM_R_TH_22:176, GAM_R_TH_23:184, GAM_R_TH_24:192, GAM_R_TH_25:200, GAM_R_TH_26:208, GAM_R_TH_27:216, GAM_R_TH_28:224, GAM_R_TH_29:232, GAM_R_TH_30:240, GAM_R_TH_31:248
GAM_R_LUT1 to GAM_R_LUT16	GAM_R_GAIN_00 to GAM_R_GAIN_31[10:0]	1024	Gain Adjustment of Area 0 to 31 of R Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])

38.8.1.7 Dither Process

Dither process is carried out by adjusting brightness/contrast or reducing 10-bit RGB signals output from the gamma correction block to 8-bit, 6-bit, or 5-bit RGB signals. The operation mode of dither process can be selected from truncate mode, round-off mode, 2 × 2 pattern dither mode and random pattern dither mode.

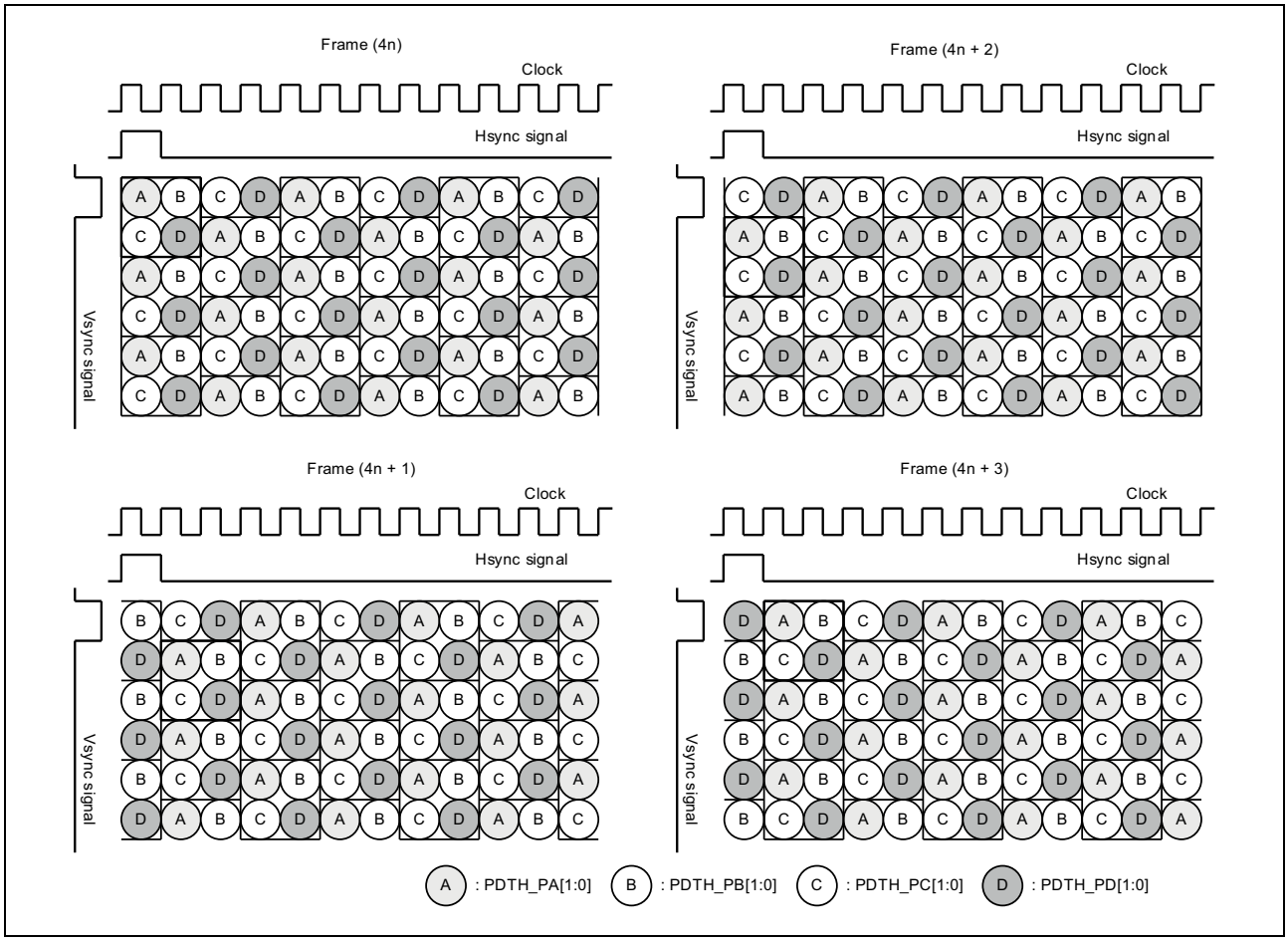


Figure 38.63 Operation Specification of 2 × 2 Pattern Dither

The conversion equations are as follows.

[Truncate mode]

(a) 10 bits to 8 bits

Output RGB data[7:0] = Input RGB data[9:0] ÷ 4 (truncate the number below the decimal point)

(b) 10 bits to 6 bits

Output RGB data[7:2] = Input RGB data[9:0] ÷ 16 (truncate the number below the decimal point)

(c) 10 bits to 5 bits

Output RGB data[7:3] = Input RGB data[9:0] ÷ 32 (truncate the number below the decimal point)

[Round-off mode]

(a) 10 bits to 8 bits

Output RGB data[7:0] = Input RGB data[9:0] ÷ 4 (round off to an integer)

(b) 10 bits to 6 bits

Output RGB data[7:2] = Input RGB data[9:0] ÷ 16 (round off to an integer)

(c) 10 bits to 5 bits

Output RGB data[7:3] = Input RGB data[9:0] ÷ 32 (round off to an integer)

[2 × 2 pattern dither mode, random pattern dither mode]

(a) 10 bits to 8 bits

Output RGB data[7:0] = Input RGB data[9:0] ÷ 4 + pattern value at the first decimal place (truncate the number below the decimal point after addition)

(b) 10 bits to 6 bits

Output RGB data[7:2] = Input RGB data[9:0] ÷ 16 + pattern value at the first decimal place (truncate the number below the decimal point after addition)

(c) 10 bits to 5 bits

Output RGB data[7:3] = Input RGB data[9:0] ÷ 32 + pattern value at the first decimal place (truncate the number below the decimal point after addition)

Table 38.145 Panel Dither Correction

Register Name	Bit Name	Initial Value	Description
OUT_PDTHA	PDTH_SEL[1:0]	0	Panel Dither Operation Mode 0: Truncate 1: Round-off 2: 2 × 2 pattern dither 3: Random pattern dither
OUT_PDTHA	PDTH_FORMAT[1:0]	0	Panel Dither Output Format Select 0: RGB888 1: RGB666 2: RGB565 3: Setting prohibited
OUT_PDTHA	PDTH_PA[1:0]	3	Pattern Value (A) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])
OUT_PDTHA	PDTH_PB[1:0]	0	Pattern Value (B) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])
OUT_PDTHA	PDTH_PC[1:0]	2	Pattern Value (C) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])
OUT_PDTHA	PDTH_PD[1:0]	1	Pattern Value (D) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])

38.8.1.8 Output Format Conversion

In output format conversion, the RGB signal after dither process is converted to LCD output signal having any of the following formats, namely, parallel RGB888, parallel RGB666, parallel RGB565, and serial RGB.

Further, converted data can be allocated to LCD output pins as selected.

(1) Bit Allocation of LCD Signals for RGB888 Output

Table 38.146 shows the RGB signal input allocated to the LCD signal output for RGB888 output.

R/G/BIN[7:0] are the RGB internal signals after dither process.

Table 38.146 Bit Allocation of RGB Signal Input for RGB888 Output

OUT_FORMAT[1:0]	0	0	0	0
OUT_ENDIAN_ON	0	0	1	1
OUT_SWAP_ON	0	1	0	1
LCD_DATA23	RIN[7]	BIN[7]	RIN[0]	BIN[0]
LCD_DATA22	RIN[6]	BIN[6]	RIN[1]	BIN[1]
LCD_DATA21	RIN[5]	BIN[5]	RIN[2]	BIN[2]
LCD_DATA20	RIN[4]	BIN[4]	RIN[3]	BIN[3]
LCD_DATA19	RIN[3]	BIN[3]	RIN[4]	BIN[4]
LCD_DATA18	RIN[2]	BIN[2]	RIN[5]	BIN[5]
LCD_DATA17	RIN[1]	BIN[1]	RIN[6]	BIN[6]
LCD_DATA16	RIN[0]	BIN[0]	RIN[7]	BIN[7]
LCD_DATA15	GIN[7]	GIN[7]	GIN[0]	GIN[0]
LCD_DATA14	GIN[6]	GIN[6]	GIN[1]	GIN[1]
LCD_DATA13	GIN[5]	GIN[5]	GIN[2]	GIN[2]
LCD_DATA12	GIN[4]	GIN[4]	GIN[3]	GIN[3]
LCD_DATA11	GIN[3]	GIN[3]	GIN[4]	GIN[4]
LCD_DATA10	GIN[2]	GIN[2]	GIN[5]	GIN[5]
LCD_DATA9	GIN[1]	GIN[1]	GIN[6]	GIN[6]
LCD_DATA8	GIN[0]	GIN[0]	GIN[7]	GIN[7]
LCD_DATA7	BIN[7]	RIN[7]	BIN[0]	RIN[0]
LCD_DATA6	BIN[6]	RIN[6]	BIN[1]	RIN[1]
LCD_DATA5	BIN[5]	RIN[5]	BIN[2]	RIN[2]
LCD_DATA4	BIN[4]	RIN[4]	BIN[3]	RIN[3]
LCD_DATA3	BIN[3]	RIN[3]	BIN[4]	RIN[4]
LCD_DATA2	BIN[2]	RIN[2]	BIN[5]	RIN[5]
LCD_DATA1	BIN[1]	RIN[1]	BIN[6]	RIN[6]
LCD_DATA0	BIN[0]	RIN[0]	BIN[7]	RIN[7]

(2) Bit Allocation of LCD Signal for RGB666 Output

Table 38.147 shows the RGB signal input allocated to the LCD signal output for RGB666 output.

R/G/BIN[7:0] are the RGB internal signals after dither process.

Table 38.147 Bit Allocation of RGB Signal Input for RGB666 Output (1/2)

OUT_FORMAT[1:0]	1	1	1	1
OUT_ENDIAN_ON	0	0	1	1
OUT_SWAP_ON	0	1	0	1
LCD_DATA23	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA22	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA21	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA20	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA19	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0

Table 38.147 Bit Allocation of RGB Signal Input for RGB666 Output (2/2)

OUT_FORMAT[1:0]	1	1	1	1
OUT_ENDIAN_ON	0	0	1	1
OUT_SWAP_ON	0	1	0	1
LCD_DATA18	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA17	RIN[7]	BIN[7]	RIN[2]	BIN[2]
LCD_DATA16	RIN[6]	BIN[6]	RIN[3]	BIN[3]
LCD_DATA15	RIN[5]	BIN[5]	RIN[4]	BIN[4]
LCD_DATA14	RIN[4]	BIN[4]	RIN[5]	BIN[5]
LCD_DATA13	RIN[3]	BIN[3]	RIN[6]	BIN[6]
LCD_DATA12	RIN[2]	BIN[2]	RIN[7]	BIN[7]
LCD_DATA11	GIN[7]	GIN[7]	GIN[2]	GIN[2]
LCD_DATA10	GIN[6]	GIN[6]	GIN[3]	GIN[3]
LCD_DATA9	GIN[5]	GIN[5]	GIN[4]	GIN[4]
LCD_DATA8	GIN[4]	GIN[4]	GIN[5]	GIN[5]
LCD_DATA7	GIN[3]	GIN[3]	GIN[6]	GIN[6]
LCD_DATA6	GIN[2]	GIN[2]	GIN[7]	GIN[7]
LCD_DATA5	BIN[7]	RIN[7]	BIN[2]	RIN[2]
LCD_DATA4	BIN[6]	RIN[6]	BIN[3]	RIN[3]
LCD_DATA3	BIN[5]	RIN[5]	BIN[4]	RIN[4]
LCD_DATA2	BIN[4]	RIN[4]	BIN[5]	RIN[5]
LCD_DATA1	BIN[3]	RIN[3]	BIN[6]	RIN[6]
LCD_DATA0	BIN[2]	RIN[2]	BIN[7]	RIN[7]

(3) Bit Allocation of LCD Signal for RGB565 Output

Table 38.148 shows the RGB signal input allocated to the LCD signal output for RGB565 output.

R/G/BIN[7:0] are the RGB internal signals after dither process.

Table 38.148 Bit Allocation of RGB Signal Input for RGB565 Output

OUT_FORMAT[1:0]	2	2	2	2
OUT_ENDIAN_ON	0	0	1	1
OUT_SWAP_ON	0	1	0	1
LCD_DATA23	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA22	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA21	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA20	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA19	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA18	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA17	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA16	Fixed to 0	Fixed to 0	Fixed to 0	Fixed to 0
LCD_DATA15	RIN[7]	BIN[7]	RIN[3]	BIN[3]
LCD_DATA14	RIN[6]	BIN[6]	RIN[4]	BIN[4]
LCD_DATA13	RIN[5]	BIN[5]	RIN[5]	BIN[5]
LCD_DATA12	RIN[4]	BIN[4]	RIN[6]	BIN[6]
LCD_DATA11	RIN[3]	BIN[3]	RIN[7]	BIN[7]
LCD_DATA10	GIN[7]	GIN[7]	GIN[2]	GIN[2]

Table 38.148 Bit Allocation of RGB Signal Input for RGB565 Output

OUT_FORMAT[1:0]	**2**	**2**	**2**	**2**
OUT_ENDIAN_ON	**0**	**0**	**1**	**1**
OUT_SWAP_ON	**0**	**1**	**0**	**1**
LCD_DATA9	GIN[6]	GIN[6]	GIN[3]	GIN[3]
LCD_DATA8	GIN[5]	GIN[5]	GIN[4]	GIN[4]
LCD_DATA7	GIN[4]	GIN[4]	GIN[5]	GIN[5]
LCD_DATA6	GIN[3]	GIN[3]	GIN[6]	GIN[6]
LCD_DATA5	GIN[2]	GIN[2]	GIN[7]	GIN[7]
LCD_DATA4	BIN[7]	RIN[7]	BIN[3]	RIN[3]
LCD_DATA3	BIN[6]	RIN[6]	BIN[4]	RIN[4]
LCD_DATA2	BIN[5]	RIN[5]	BIN[5]	RIN[5]
LCD_DATA1	BIN[4]	RIN[4]	BIN[6]	RIN[6]
LCD_DATA0	BIN[3]	RIN[3]	BIN[7]	RIN[7]

(4) Bit Allocation of LCD Serial for Serial RGB Output

For serial RGB output, RGB signal input shown **Table 38.149** is allocated to rgb internal signals and the signals are converted from parallel to serial format and output as LCD signals. R/G/BIN[7:0] are the RGB internal signals after dither process.

The internal signals r[7:0], g[7:0], and b[7:0] are serially output to LCD_DATA7 to LCD_DATA0.

Table 38.149 Bit Allocation of RGB Signal Input for RGB565 Output

OUT_FORMAT[1:0]	3	3	3	3
OUT_ENDIAN_ON	0	0	1	1
OUT_SWAP_ON	0	1	0	1
r[7]	RIN[7]	BIN[7]	RIN[0]	BIN[0]
r[6]	RIN[6]	BIN[6]	RIN[1]	BIN[1]
r[5]	RIN[5]	BIN[5]	RIN[2]	BIN[2]
r[4]	RIN[4]	BIN[4]	RIN[3]	BIN[3]
r[3]	RIN[3]	BIN[3]	RIN[4]	BIN[4]
r[2]	RIN[2]	BIN[2]	RIN[5]	BIN[5]
r[1]	RIN[1]	BIN[1]	RIN[6]	BIN[6]
r[0]	RIN[0]	BIN[0]	RIN[7]	BIN[7]
g[7]	GIN[7]	GIN[7]	GIN[0]	GIN[0]
g[6]	GIN[6]	GIN[6]	GIN[1]	GIN[1]
g[5]	GIN[5]	GIN[5]	GIN[2]	GIN[2]
g[4]	GIN[4]	GIN[4]	GIN[3]	GIN[3]
g[3]	GIN[3]	GIN[3]	GIN[4]	GIN[4]
g[2]	GIN[2]	GIN[2]	GIN[5]	GIN[5]
g[1]	GIN[1]	GIN[1]	GIN[6]	GIN[6]
g[0]	GIN[0]	GIN[0]	GIN[7]	GIN[7]
b[7]	BIN[7]	RIN[7]	BIN[0]	RIN[0]
b[6]	BIN[6]	RIN[6]	BIN[1]	RIN[1]
b[5]	BIN[5]	RIN[5]	BIN[2]	RIN[2]
b[4]	BIN[4]	RIN[4]	BIN[3]	RIN[3]
b[3]	BIN[3]	RIN[3]	BIN[4]	RIN[4]
b[2]	BIN[2]	RIN[2]	BIN[5]	RIN[5]
b[1]	BIN[1]	RIN[1]	BIN[6]	RIN[6]
b[0]	BIN[0]	RIN[0]	BIN[7]	RIN[7]

(5) Parallel to Serial Conversion

As shown in the table below, four types of parallel to serial conversions are possible by controlling clock speed mode and selecting the scan direction ('n' in the table are natural numbers).

Table 38.150 Specifications of Serial RGB Output

OUT_FRQ_SEL[1:0]	1	1	1	1	2	2
OUT_DIR_SEL	**0**	**1**	**0**	**1**	**0**	**1**
OUT_SEQ_SEL	**0**	**0**	**1**	**1**	**–**	**–**
Line (2n-1)	Repeated (r → g → b)	Repeated (b → g → r)	Repeated (r → g → b)	Repeated (b → g → r)	Repeated (r → g → b → X)	Repeated (X → b → g → r)
Line 2n	Repeated (g → b → r)	Repeated (r → b → g)	Repeated (r → g → b)	Repeated (b → g → r)	Repeated (r → g → b → X)	Repeated (X → b → g → r)

The following figures show the timing of parallel to serial conversion in triple speed and quadruple speed modes, respectively.

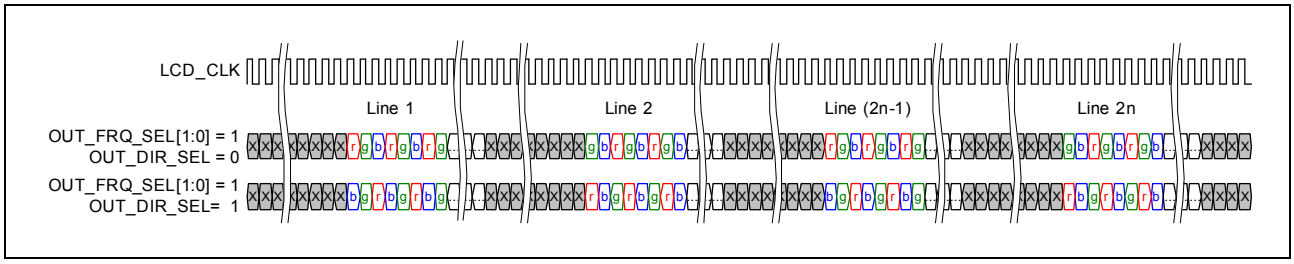


Figure 38.64 Timing of Parallel to Serial Conversion in Triple Speed Mode (OUT_SEQ_SEL = 0)

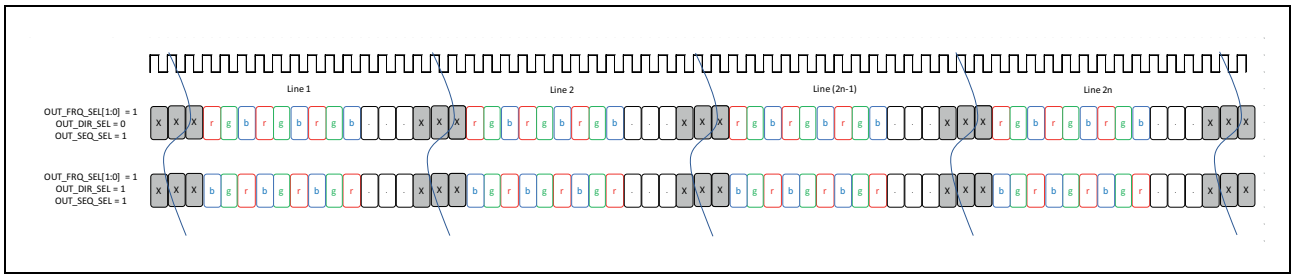


Figure 38.65 Timing of Parallel to Serial Conversion in Triple Speed Mode Mode (OUT_SEQ_SEL = 1)

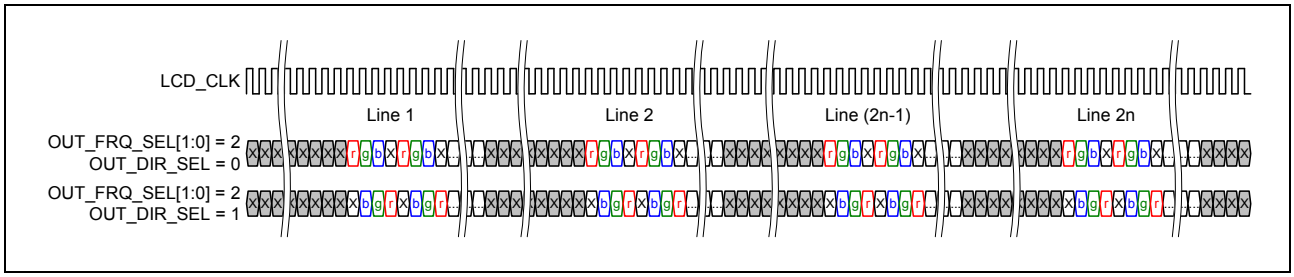


Figure 38.66 Timing of Parallel to Serial Conversion in Quadruple Speed Mode

During serial output, the phase timing with the HE signal can be adjusted by OUT_PHASE[0:1].

The following figure shows the timing of the clock phases of the serial RGB output (triple speed mode).

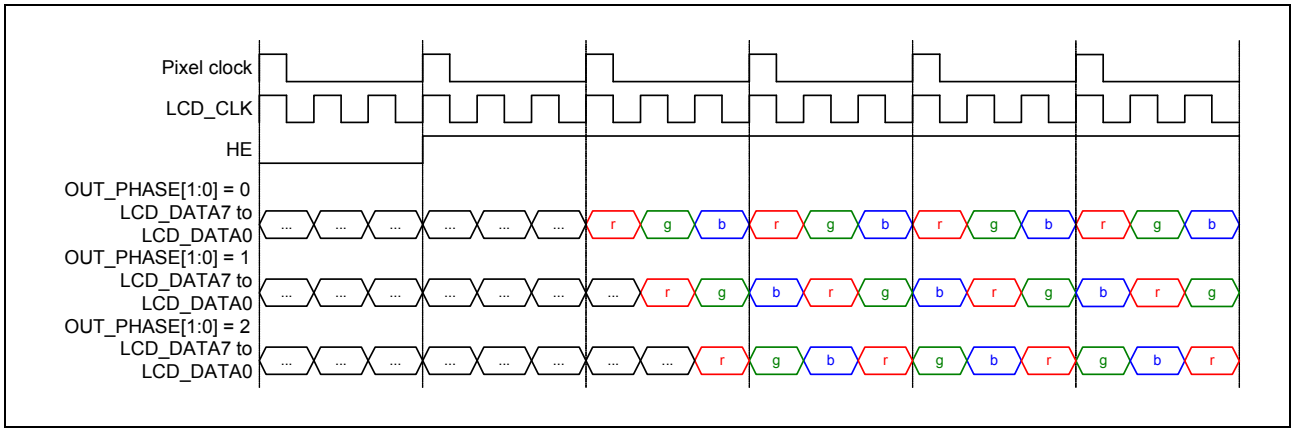


Figure 38.67 Timing of Clock Phases of Serial RGB Output (Triple Speed Mode)

The following figure shows the timing of the clock phases of the serial RGB output (quadruple speed mode).

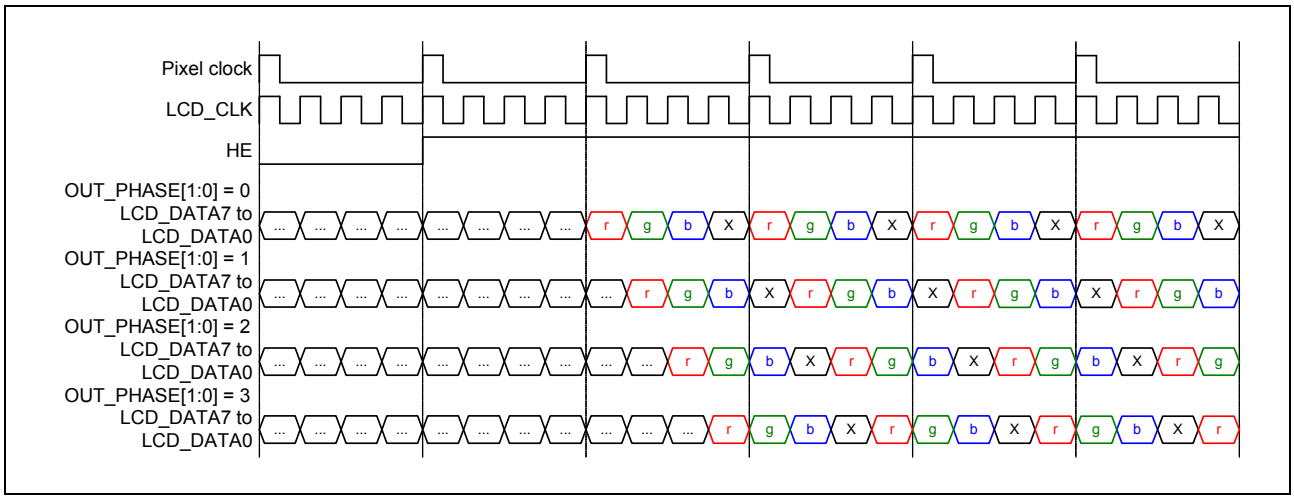


Figure 38.68 Timing of Clock Phases of Serial RGB Output (Quadruple Speed Mode)

Table 38.151 Output Format Conversion (1/2)

Register Name	Bit Name	Initial Value	Description
OUT_SET	OUT_FORMAT[1:0]	0	Output Format Select 0: RBG888 1: RGB666 2: RGB565 3: Serial RGB
OUT_SET	OUT_ENDIAN_ON	0	Bit Endian Change On/Off Control 0: Off 1: On
OUT_SET	OUT_SWAP_ON	0	B/R Signal Swap On/Off Control 0: Off 1: On

Table 38.151 Output Format Conversion (2/2)

Register Name	Bit Name	Initial Value	Description
OUT_SET	OUT_FRQ_SEL[1:0]	0	Clock Frequency Control 0: 100% speed — (parallel RGB) 1: Triple speed — (serial RGB) 2: Quadruple speed — (serial RGB) 3: Setting prohibited
OUT_SET	OUT_DIR_SEL	0	Scan Direction Select 0: Forward scan 1: Reverse scan
OUT_SET	OUT_PHASE[1:0]	0	Clock Phase Adjustment for Serial RGB Output • Triple speed mode 0: 0 (clk) 1: 1 (clk) 2: 2 (clk) 3: Setting prohibited • Quadruple speed mode 0: 0 (clk) 1: 1 (clk) 2: 2 (clk) 3: 3 (clk)

NOTE

In the case of Serial RGB, horizontal enable signal width for full screen (SC0_SCL0_FRC7.SC0_RES_F_HW[10:0]) must be set the original (parallel RGB) width plus 2.

38.8.1.9 LCD TCON

The LCD TCON generates various timing signals for driving the LCD panel.

Specifically, the timing include two vertical panel driver signals, five horizontal panel driver signals, and one composite signal of the vertical and horizontal panel driver signals. **Table 38.152** lists the timing signals that are generated by LCD TCON

Table 38.152 Signals Generated by LCD TCON

Signal Name	Type	Description
STVA/VS	Vertical	• Gate start signal The pulse width, pulse position, and pulse polarity of the signal can be controlled. • Vsync signal The width, position, and polarity of the sync signal can be controlled.
STVB/VE	Vertical	• Gate start signal The pulse width, pulse position, and pulse polarity of the signal can be controlled. • Vertical enable signal The width, position, and polarity of the sync signal can be controlled.
STVA/HS	Horizontal	• Source start signal The pulse width, pulse position, and pulse polarity of the signal can be controlled. • Hsync signal The width, position, and polarity of the sync signal can be controlled.
STB/LP/HE	Horizontal	• Source strobe signal The pulse width, pulse position, and pulse polarity of the signal can be controlled. • Horizontal enable signal The width, position, and polarity of the enable signal can be controlled.
CPV/GCK	Horizontal	• Gate clock signal The pulse width, pulse position, and pulse polarity of the signal can be controlled.
POLA	Horizontal	• VCOM voltage polarity control signal The polarity inversion position, and polarity inversion operation (1 × 1, 1 × 2, 2 × 2) can be controlled.
POLB	Horizontal	• VCOM voltage polarity control signal The polarity inversion position, and polarity inversion operation (1 × 1, 1 × 2, 2 × 2) can be controlled.
DE	Horizontal/Vertical	• Data enable signal The width, position, and polarity of the enable signal can be controlled.

(1) Horizontal Reference Offset Control

The horizontal reference offset control enables generation of a reference signal with a clock delay equivalent to the value of TCON_OFFSET[10:0] from the rising edge of the Hsync signal. If a signal that spans across the Hsync signal needs to be generated, such a signal is generated with reference to the offset reference signal.

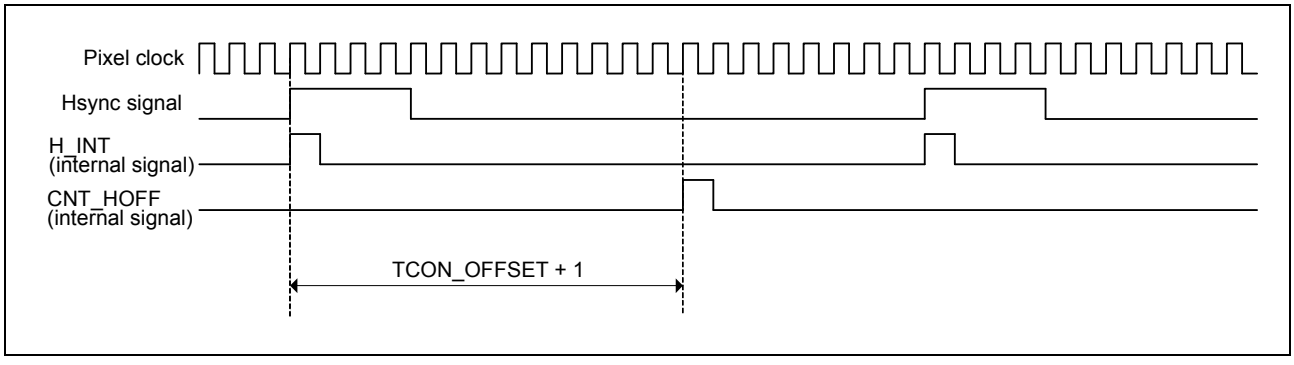


Figure 38.69 Generation of Offset Horizontal Reference (H_OFF) Signal

Table 38.153 Horizontal Reference Signal Selection

Register Name	Bit Name	Initial Value	Description
TCON_TIM	TCON_OFFSET[10:0]	0	Offset Hsync Signal Timing Sets the clock cycle count from the rising edge of the Hsync signal.
TCON_TIM_STH2	TCON_STH_HS_SEL	0	STH Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
TCON_TIM_STB2	TCON_STB_HS_SEL	0	STB Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
TCON_TIM_CPV2	TCON_CPV_HS_SEL	0	CPV Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
TCON_TIM_POLA2	TCON_POLA_HS_SEL	0	POLA Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
TCON_TIM_POLB2	TCON_POLB_HS_SEL	0	POLB Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference

Note: When generating the POLA and POLB signals in reverse mode, the bits TCON_POLA_HS_SEL and TCON_POLB_HS_SEL should be set to 0.

(2) Horizontal Panel Driver Signal Generation (A)

Horizontal synchronous panel driver signal generation (A) involves generation of a timing signal that changes twice in a horizontal period according to the values of TCON_xxxx_HS[10:0] and TCON_xxxx_HW[10:0] bits, which set the first changing timing and the second changing timing, respectively.

The internal counter performs the following operations.

1. Resets the counter value at the rising edge of the Hsync signal as the reference.
2. Increments the counter value at the rising edge of the panel clock.

A fixed output value of 0 can be obtained by setting 0 in TCON_xxxx_HW[10:0], which set the second changing timing.

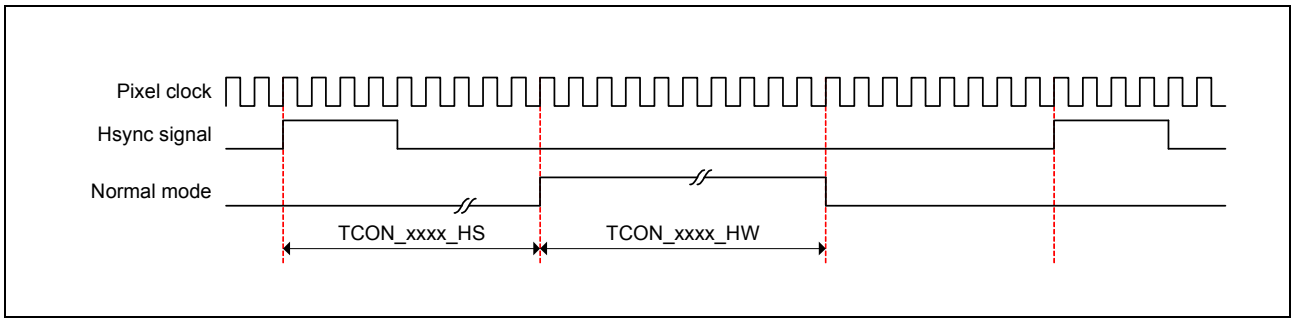


Figure 38.70 Horizontal Panel Driver Signal (in Normal Mode)

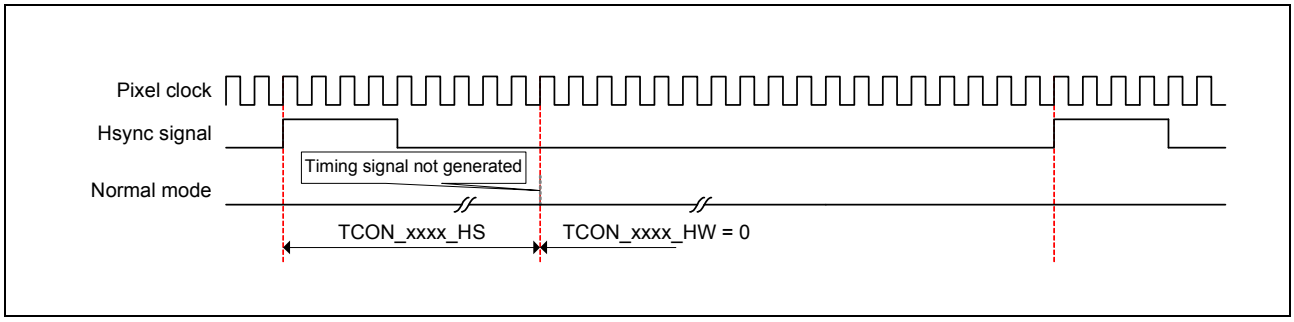


Figure 38.71 Horizontal Panel Driver Signal (in Normal Mode and When TCON_xxxx_HW_ = 0)

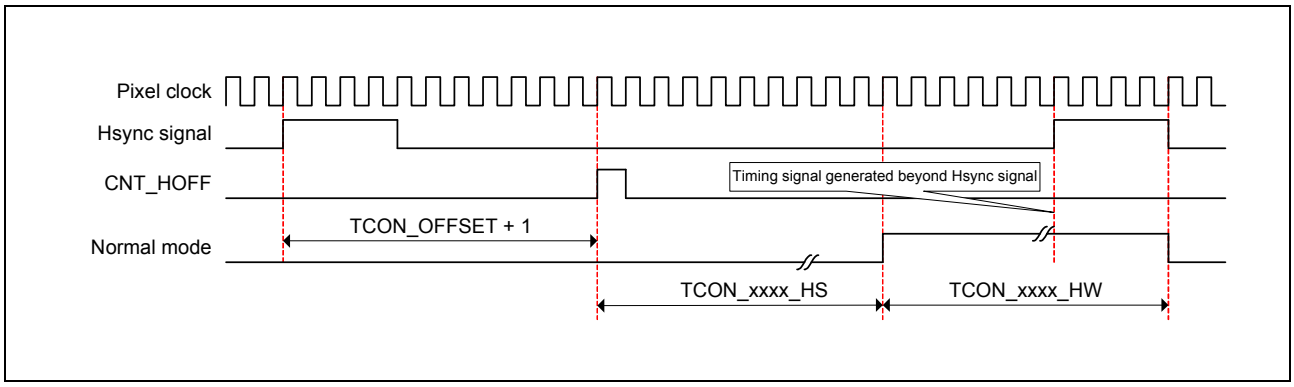


Figure 38.72 Horizontal Panel Driver Signal (in Normal Mode and When Offset Horizontal Reference is Used)

Table 38.154 Settings for Horizontal Panel Driver Signal Generation (A)

Register Name	Bit Name	Initial Value	Description
TCON_TIM_STH1	TCON_STH_HS[10:0]	0	STH Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STH_HS[10:0] from the rising edge of the Hsync signal (clock cycles)
TCON_TIM_STH1	TCON_STH_HW[10:0]	96	STH Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STH_HW[10:0] (clock cycles)
TCON_TIM_STB1	TCON_STB_HS[10:0]	144	STB Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STB_HS[10:0] from the rising edge of the Hsync signal (clock cycles)
TCON_TIM_STB1	TCON_STB_HW[10:0]	640	STB Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STB_HW[10:0] (clock cycles)
TCON_TIM_CPV1	TCON_CPV_HS[10:0]	0	CPV Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_CPV_HS[10:0] from the rising edge of the Hsync signal (clock cycles)
TCON_TIM_CPV1	TCON_CPV_HW[10:0]	0	CPV Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_CPV_HW[10:0] (clock cycles)

(3) Horizontal Panel Driver Signal Generation (B)

In addition to the normal mode operation described in (2), reverse mode operation, that is, horizontal panel driver signal generation (B) is provided. In reverse mode, operation starts at the rising edge of the

Vsync signal as the reference and a signal is generated such that its polarity is inverted every horizontal period in the timing set by the TCON_xxxx_HS[10:0] bits, which set the first changing timing.

In reverse mode, regardless of whether the number of lines in the vertical direction is odd or even, the polarity of the signals generated is inverted every vertical period. The following three reverse modes are selectable for polarity inversion operation.

Table 38.155 Horizontal Panel Driver Signal Generation Modes

Register Name	Bit Name	Initial Value	Description
TCON_TIM_POLA2	TCON_POLA_MD [1:0]	1	POLA Signal Generation Mode Select 0: Normal mode Generates the signal that changes twice a horizontal period. 1: 1 × 1 reverse mode Generates the signal whose polarity is inverted every horizontal period. 2: 1 × 2 reverse mode Generates the signal whose polarity is inverted in the first horizontal period and is subsequently inverted every two horizontal periods. 3: 2 × 2 reverse mode Generates the signal whose polarity is inverted every two horizontal periods.
TCON_TIM_POLB2	TCON_POLB_MD [1:0]	1	POLB Signal Generation Mode Select 0: Normal mode Generates the signal that changes twice a horizontal period. 1: 1 × 1 reverse mode Generates the signal whose polarity is inverted every horizontal period. 2: 1 × 2 reverse mode Generates the signal whose polarity is inverted in the first horizontal period and is subsequently inverted every two horizontal periods. 3: 2 × 2 reverse mode Generates the signal whose polarity is inverted every two horizontal periods.

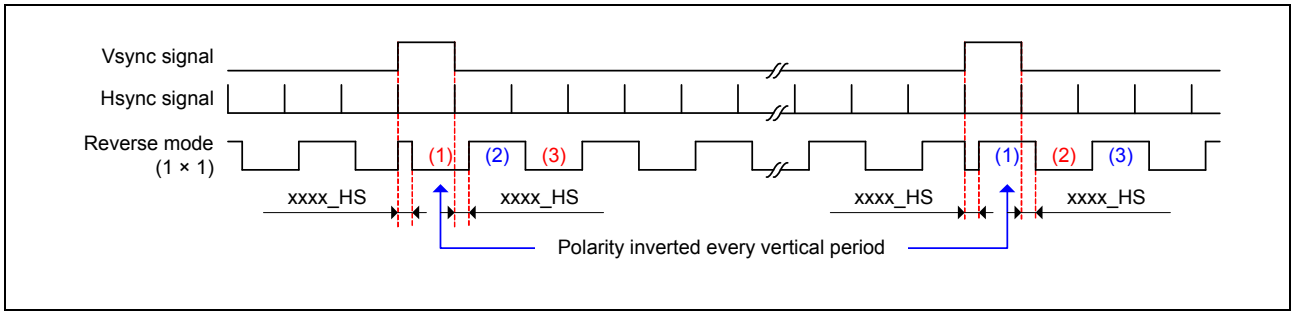


Figure 38.73 Horizontal Panel Driver Signal (in 1 × 1 Reverse Mode)

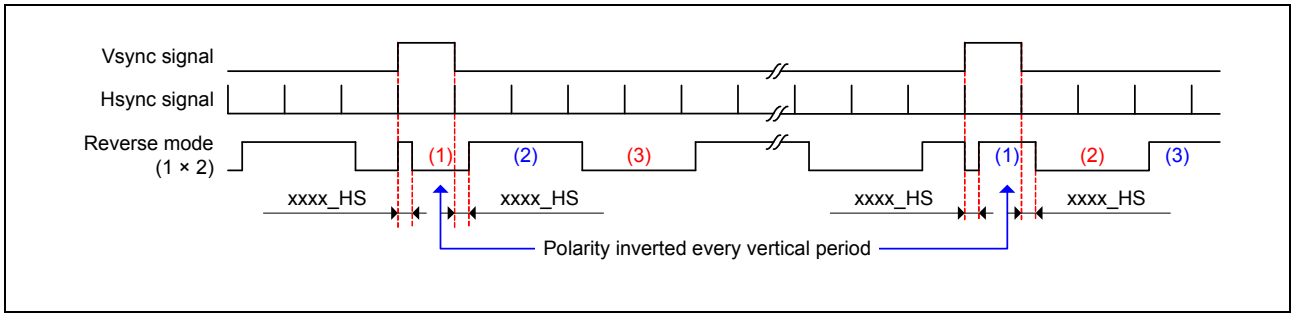


Figure 38.74 Horizontal Panel Driver Signal (in 1 × 2 Reverse Mode)

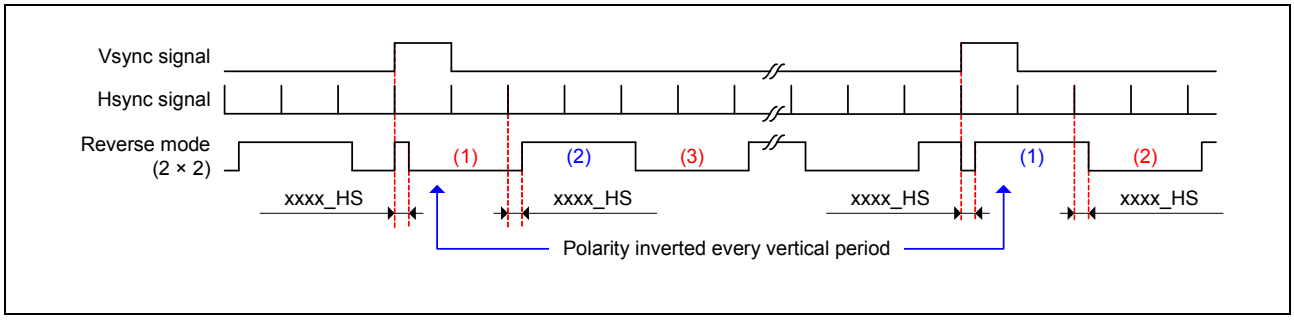


Figure 38.75 Horizontal Panel Driver Signal (in 2 × 2 Reverse Mode)

Table 38.156 Settings of Horizontal Panel Driver Signal Generation (B)

Register Name	Bit Name	Initial Value	Description
TCON_TIM_POLA1	TCON_POLA_HS [10:0]	0	POLA Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_POLA_HS[10:0] from the rising edge of the Hsync signal (clock cycles) Note: When 1 × 1, 1 × 2, or 2 × 2 reverse mode is selected, these bits should be set to 1 or greater.
TCON_TIM_POLA1	TCON_POLA_HW [10:0]	0	POLA Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_POLA_HW[10:0] (clock cycles)
TCON_TIM_POLB1	TCON_POLB_HS [10:0]	0	POLBA Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_POLB_HS[10:0] from the rising edge of the Hsync signal (clock cycles) Note: When 1 × 1, 1 × 2, or 2 × 2 reverse mode is selected, these bits should be set to 1 or greater.
TCON_TIM_POLB1	TCON_POLB_HW [10:0]	0	POLB Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_POLB_HW[10:0] (clock cycles)

(4) Vertical Panel Driver Signal Generation

The vertical synchronous panel driver signal generation involves the following operations.

1. Initialization at the rising edge of the Vsync signal
2. Generation of a timing signal that changes twice in a vertical period according to the values of the internal counter, and TCON_xxxx_VS[10:0] and TCON_xxxx_VW[10:0] bits, which set the first changing timing and the second changing timing, respectively.

The internal counter increments the counter value in the following two cases.

1. At the rising edge of the Hsync signal
2. At the point reached after a clock delay specified by the value of TCON_HALF[10:0] from the rising edge of the Hsync signal (normally, 1/2fH is set).

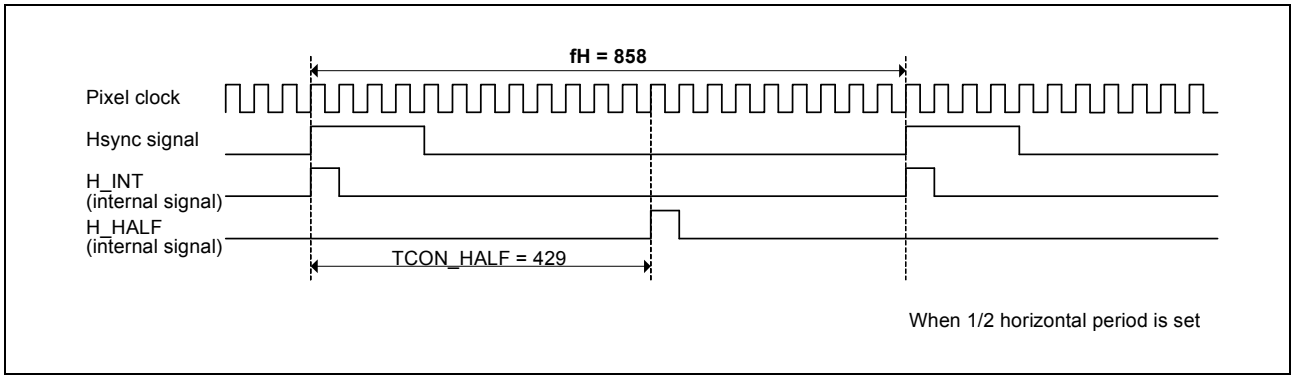


Figure 38.76 1/2 Pulse (H_HALF) Signal Generation

Table 38.157 Settings of 1/2 Pulse (H_HALF) Signal Generation

Register Name	Bit Name	Initial Value	Description
TCON_TIM	TCON_HALF[10:0]	400	1/2fH Timing Specifies the clock count from the rising edge of the Hsync signal as the counting timing of horizontal counter

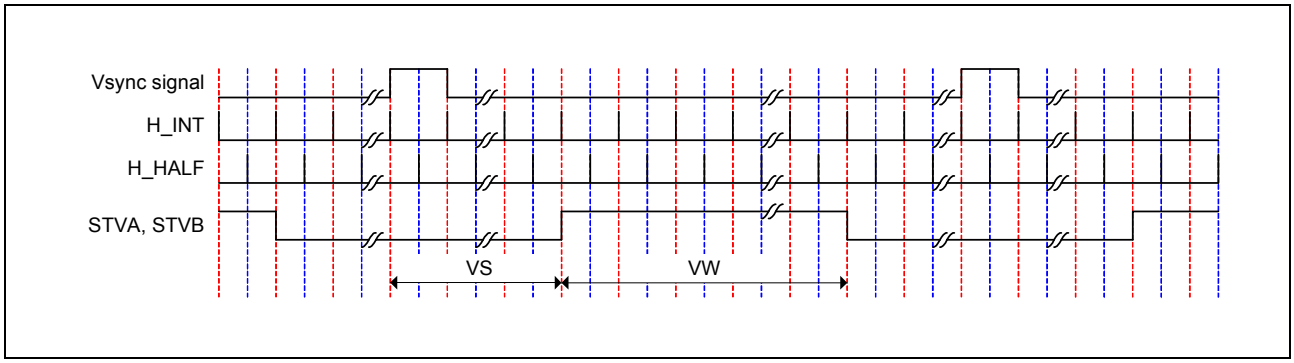


Figure 38.77 Vertical Panel Driver Signal (H_INT Reference Operation)

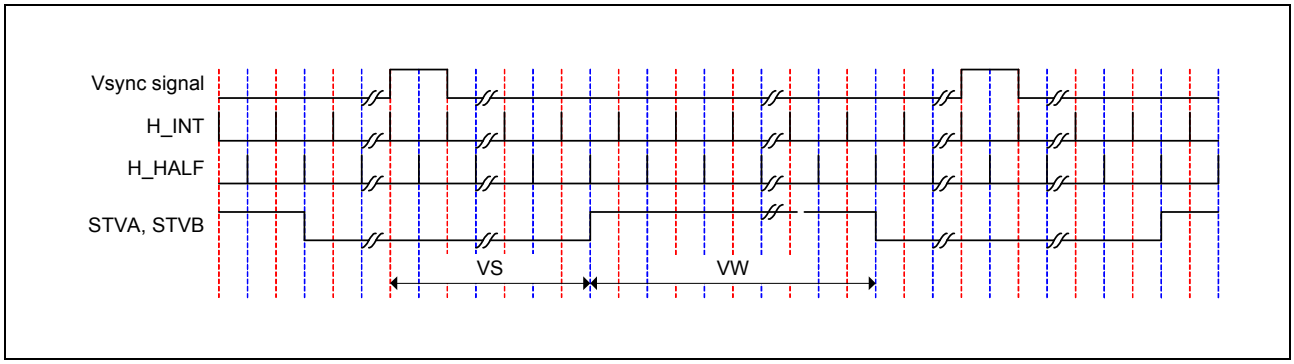


Figure 38.78 Vertical Panel Driver Signal (H_HALF Reference Operation)

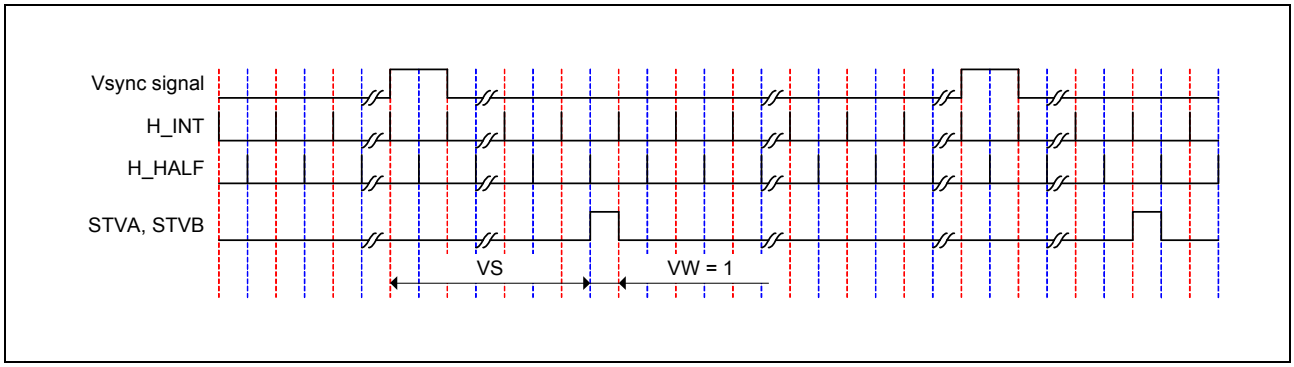


Figure 38.79 Vertical Panel Driver Signal (H_INT and H_HALF Reference Operation)

Table 38.158 Vertical Panel Driver Signal Generation

Register Name	Bit Name	Initial Value	Description
TCON_TIM_STVA1	TCON_STVA_VS[10:0]	0	STVA Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STVA_VS[10:0] from the rising edge of the Vsync signal (1/2fH cycles)
TCON_TIM_STVA1	TCON_STVA_VW[10:0]	4	STVA Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STVA_VW[10:0] (1/2fH cycles)
TCON_TIM_STVB1	TCON_STVB_VS[10:0]	70	STVB Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STVB_VS[10:0] from the rising edge of the Vsync signal (1/2fH cycles)
TCON_TIM_STVB1	TCON_STVB_VW[10:0]	960	STVB Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STVB_VW[10:0] (1/2fH cycles)

(5) DE Timing Signal Generation

DE timing signal generation involves generation of data enable signal (DE) that indicates the valid period of the video signal by synthesizing the horizontal panel driver (HE) signal and the vertical panel driver (VE) signal (AND).

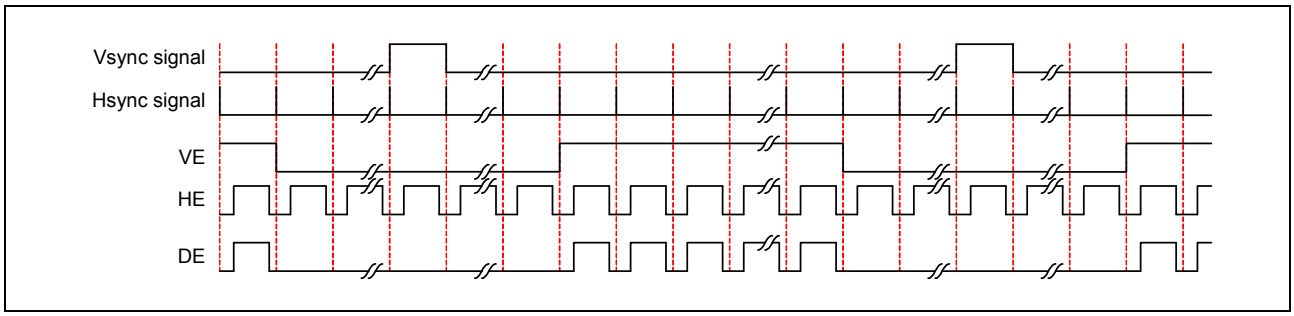


Figure 38.80 Data Enable Signal Generation

(6) Polarity Inversion

Polarity inversion enables inversion of polarity of each signal generated by the signal generating circuit.

Table 38.159 Panel Driver Signal Polarity Inversion Control

Register Name	Bit Name	Initial Value	Description
TCON_TIM_STVA2	TCON_STVA_INV	1	Polarity Inversion Control of STVA Signal 0: Not inverted 1: Inverted
TCON_TIM_STVB2	TCON_STVB_INV	0	Polarity Inversion Control of STVB Signal 0: Not inverted 1: Inverted
TCON_TIM_STH2	TCON_STH_INV	1	Polarity Inversion Control of STH Signal 0: Not inverted 1: Inverted
TCON_TIM_STB2	TCON_STB_INV	0	Polarity Inversion Control of STB Signal 0: Not inverted 1: Inverted
TCON_TIM_CPV2	TCON_CPV_INV	0	Polarity Inversion Control of CPV Signal 0: Not inverted 1: Inverted
TCON_TIM_POLA2	TCON_POLA_INV	0	Polarity Inversion Control of POLA Signal 0: Not inverted 1: Inverted
TCON_TIM_POLB2	TCON_POLB_INV	0	Polarity Inversion Control of POLB Signal 0: Not inverted 1: Inverted
TCON_TIM_DE	TCON_DE_INV	0	Polarity Inversion Control of DE Signal 0: Not inverted 1: Inverted

(7) Output Selection

An output pin is selected for every signal subjected to polarity inversion control.

Table 38.160 Panel Driver Signal Output Selection

Register Name	Bit Name	Initial Value	Description
TCON_TIM_STVA2	TCON_STVA_SEL [2:0]	0	Output Signal Select for LCD_TCON0 Pin 0: STVA/VS Others: prohibited
TCON_TIM_STVB2	TCON_STVB_SEL [2:0]	1	Output Signal Select for LCD_TCON1 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE
TCON_TIM_STH2	TCON_STH_SEL [2:0]	2	Output Signal Select for LCD_TCON2 Pin 2: STH/SP/HS Others: prohibited
TCON_TIM_STB2	TCON_STB_SEL [2:0]	7	Output Signal Select for LCD_TCON3 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE*1
TCON_TIM_CPV2	TCON_CPV_SEL [2:0]	4	Output Signal Select for LCD_TCON4 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE
TCON_TIM_POLA2	TCON_POLA_SEL [2:0]	5	Output Signal Select for LCD_TCON5 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE
TCON_TIM_POLB2	TCON_POLB_SEL [2:0]	6	Output Signal Select for LCD_TCON6 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

Note 1. Though the signal of the LCD_TCON3 pin can be freely selected from the list, it is recommended to select the DE signal (TCON_STB_SEL[2:0] = 7). This way the DE signal is located in the neighborhood of the other TCON signals at the device package, which might ease the PCB layout.

(8) Output Phase Selection

The output phase can be individually selected for the video output signal and the various timing output signals based on the LCD_CLK (panel clock).

Table 38.161 Panel Output Signal Phase Selection

Register Name	Bit Name	Initial Value	Description
OUT_CLK_PHASE	OUTCNT_LCD_EDGE	0	Output Phase Control of LCD_DATA23 to LCD_DATA0 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_STVA_EDGE	0	Output Phase Control of LCD_TCON0 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_STVB_EDGE	0	Output Phase Control of LCD_TCON1 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_STH_EDGE	0	Output Phase Control of LCD_TCON2 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_STB_EDGE	0	Output Phase Control of LCD_TCON3 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_CPV_EDGE	0	Output Phase Control of LCD_TCON4 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_POLA_EDGE	0	Output Phase Control of LCD_TCON5 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
OUT_CLK_PHASE	OUTCNT_POLB_EDGE	0	Output Phase Control of LCD_TCON6 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin

38.8.2 Register Descriptions

Table 38.162 to **Table 38.167** shows the register configuration.

- Symbols used in Register Description:

  Initial value: Register value after a reset

  —: Undefined value

  R/W: Readable/writable. The written value can be read.

  R/WC0: Readable/writable. Writing 0 initializes the bit. Writing 1 is ignored.

  R/WC1: Readable/writable. Writing 1 initializes the bit. Writing 0 is ignored.

  R: Read-only. The write value should always be 0.

  —/W: Write-only. The read value is undefined.

Table 38.162 Gamma Correction Block Register Configuration (Channel 0) (1/4)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Register update control register G in gamma correction block	GAM_G_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1800	32
VDCE0	Function switch register in gamma correction block	GAM_SW	R/W	H'0000 0000	<VDCE0_base> + 1804	32
VDCE0	Table setting register G1 in gamma correction block	GAM_G_LUT1	R/W	H'0400 0400	<VDCE0_base> + 1808	32
VDCE0	Table setting register G2 in gamma correction block	GAM_G_LUT2	R/W	H'0400 0400	<VDCE0_base> + 180C	32
VDCE0	Table setting register G3 in gamma correction block	GAM_G_LUT3	R/W	H'0400 0400	<VDCE0_base> + 1810	32
VDCE0	Table setting register G4 in gamma correction block	GAM_G_LUT4	R/W	H'0400 0400	<VDCE0_base> + 1814	32
VDCE0	Table setting register G5 in gamma correction block	GAM_G_LUT5	R/W	H'0400 0400	<VDCE0_base> + 1818	32
VDCE0	Table setting register G6 in gamma correction block	GAM_G_LUT6	R/W	H'0400 0400	<VDCE0_base> + 181C	32
VDCE0	Table setting register G7 in gamma correction block	GAM_G_LUT7	R/W	H'0400 0400	<VDCE0_base> + 1820	32
VDCE0	Table setting register G8 in gamma correction block	GAM_G_LUT8	R/W	H'0400 0400	<VDCE0_base> + 1824	32
VDCE0	Table setting register G9 in gamma correction block	GAM_G_LUT9	R/W	H'0400 0400	<VDCE0_base> + 1828	32
VDCE0	Table setting register G10 in gamma correction block	GAM_G_LUT10	R/W	H'0400 0400	<VDCE0_base> + 182C	32
VDCE0	Table setting register G11 in gamma correction block	GAM_G_LUT11	R/W	H'0400 0400	<VDCE0_base> + 1830	32
VDCE0	Table setting register G12 in gamma correction block	GAM_G_LUT12	R/W	H'0400 0400	<VDCE0_base> + 1834	32
VDCE0	Table setting register G13 in gamma correction block	GAM_G_LUT13	R/W	H'0400 0400	<VDCE0_base> + 1838	32
VDCE0	Table setting register G14 in gamma correction block	GAM_G_LUT14	R/W	H'0400 0400	<VDCE0_base> + 183C	32
VDCE0	Table setting register G15 in gamma correction block	GAM_G_LUT15	R/W	H'0400 0400	<VDCE0_base> + 1840	32
VDCE0	Table setting register G16 in gamma correction block	GAM_G_LUT16	R/W	H'0400 0400	<VDCE0_base> + 1844	32

Table 38.162 Gamma Correction Block Register Configuration (Channel 0) (2/4)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Area setting register G1 in gamma correction block	GAM_G_AREA1	R/W	H'0008 1018	<VDCE0_base> + 1848	32
VDCE0	Area setting register G2 in gamma correction block	GAM_G_AREA2	R/W	H'2028 3038	<VDCE0_base> + 184C	32
VDCE0	Area setting register G3 in gamma correction block	GAM_G_AREA3	R/W	H'4048 5058	<VDCE0_base> + 1850	32
VDCE0	Area setting register G4 in gamma correction block	GAM_G_AREA4	R/W	H'6068 7078	<VDCE0_base> + 1854	32
VDCE0	Area setting register G5 in gamma correction block	GAM_G_AREA5	R/W	H'8088 9098	<VDCE0_base> + 1858	32
VDCE0	Area setting register G6 in gamma correction block	GAM_G_AREA6	R/W	H'A0A8 B0B8	<VDCE0_base> + 185C	32
VDCE0	Area setting register G7 in gamma correction block	GAM_G_AREA7	R/W	H'C0C8 D0D8	<VDCE0_base> + 1860	32
VDCE0	Area setting register G8 in gamma correction block	GAM_G_AREA8	R/W	H'E0E8 F0F8	<VDCE0_base> + 1864	32
VDCE0	Register update control register B in gamma correction block	GAM_B_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1880	32
VDCE0	Table setting register B1 in gamma correction block	GAM_B_LUT1	R/W	H'0400 0400	<VDCE0_base> + 1888	32
VDCE0	Table setting register B2 in gamma correction block	GAM_B_LUT2	R/W	H'0400 0400	<VDCE0_base> + 188C	32
VDCE0	Table setting register B3 in gamma correction block	GAM_B_LUT3	R/W	H'0400 0400	<VDCE0_base> + 1890	32
VDCE0	Table setting register B4 in gamma correction block	GAM_B_LUT4	R/W	H'0400 0400	<VDCE0_base> + 1894	32
VDCE0	Table setting register B5 in gamma correction block	GAM_B_LUT5	R/W	H'0400 0400	<VDCE0_base> + 1898	32
VDCE0	Table setting register B6 in gamma correction block	GAM_B_LUT6	R/W	H'0400 0400	<VDCE0_base> + 189C	32
VDCE0	Table setting register B7 in gamma correction block	GAM_B_LUT7	R/W	H'0400 0400	<VDCE0_base> + 18A0	32
VDCE0	Table setting register B8 in gamma correction block	GAM_B_LUT8	R/W	H'0400 0400	<VDCE0_base> + 18A4	32
VDCE0	Table setting register B9 in gamma correction block	GAM_B_LUT9	R/W	H'0400 0400	<VDCE0_base> + 18A8	32
VDCE0	Table setting register B10 in gamma correction block	GAM_B_LUT10	R/W	H'0400 0400	<VDCE0_base> + 18AC	32
VDCE0	Table setting register B11 in gamma correction block	GAM_B_LUT11	R/W	H'0400 0400	<VDCE0_base> + 18B0	32
VDCE0	Table setting register B12 in gamma correction block	GAM_B_LUT12	R/W	H'0400 0400	<VDCE0_base> + 18B4	32
VDCE0	Table setting register B13 in gamma correction block	GAM_B_LUT13	R/W	H'0400 0400	<VDCE0_base> + 18B8	32
VDCE0	Table setting register B14 in gamma correction block	GAM_B_LUT14	R/W	H'0400 0400	<VDCE0_base> + 18BC	32
VDCE0	Table setting register B15 in gamma correction block	GAM_B_LUT15	R/W	H'0400 0400	<VDCE0_base> + 18C0	32
VDCE0	Table setting register B16 in gamma correction block	GAM_B_LUT16	R/W	H'0400 0400	<VDCE0_base> + 18C4	32
VDCE0	Area setting register B1 in gamma correction block	GAM_B_AREA1	R/W	H'0008 1018	<VDCE0_base> + 18C8	32

Table 38.162 Gamma Correction Block Register Configuration (Channel 0) (3/4)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Area setting register B2 in gamma correction block	GAM_B_AREA2	R/W	H'2028 3038	<VDCE0_base> + 18CC	32
VDCE0	Area setting register B3 in gamma correction block	GAM_B_AREA3	R/W	H'4048 5058	<VDCE0_base> + 18D0	32
VDCE0	Area setting register B4 in gamma correction block	GAM_B_AREA4	R/W	H'6068 7078	<VDCE0_base> + 18D4	32
VDCE0	Area setting register B5 in gamma correction block	GAM_B_AREA5	R/W	H'8088 9098	<VDCE0_base> + 18D8	32
VDCE0	Area setting register B6 in gamma correction block	GAM_B_AREA6	R/W	H'A0A8 B0B8	<VDCE0_base> + 18DC	32
VDCE0	Area setting register B7 in gamma correction block	GAM_B_AREA7	R/W	H'C0C8 D0D8	<VDCE0_base> + 18E0	32
VDCE0	Area setting register B8 in gamma correction block	GAM_B_AREA8	R/W	H'E0E8 F0F8	<VDCE0_base> + 18E4	32
VDCE0	Register update control register R in gamma correction block	GAM_R_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1900	32
VDCE0	Table setting register R1 in gamma correction block	GAM_R_LUT1	R/W	H'0400 0400	<VDCE0_base> + 1908	32
VDCE0	Table setting register R2 in gamma correction block	GAM_R_LUT2	R/W	H'0400 0400	<VDCE0_base> + 190C	32
VDCE0	Table setting register R3 in gamma correction block	GAM_R_LUT3	R/W	H'0400 0400	<VDCE0_base> + 1910	32
VDCE0	Table setting register R4 in gamma correction block	GAM_R_LUT4	R/W	H'0400 0400	<VDCE0_base> + 1914	32
VDCE0	Table setting register R5 in gamma correction block	GAM_R_LUT5	R/W	H'0400 0400	<VDCE0_base> + 1918	32
VDCE0	Table setting register R6 in gamma correction block	GAM_R_LUT6	R/W	H'0400 0400	<VDCE0_base> + 191C	32
VDCE0	Table setting register R7 in gamma correction block	GAM_R_LUT7	R/W	H'0400 0400	<VDCE0_base> + 1920	32
VDCE0	Table setting register R8 in gamma correction block	GAM_R_LUT8	R/W	H'0400 0400	<VDCE0_base> + 1924	32
VDCE0	Table setting register R9 in gamma correction block	GAM_R_LUT9	R/W	H'0400 0400	<VDCE0_base> + 1928	32
VDCE0	Table setting register R10 in gamma correction block	GAM_R_LUT10	R/W	H'0400 0400	<VDCE0_base> + 192C	32
VDCE0	Table setting register R11 in gamma correction block	GAM_R_LUT11	R/W	H'0400 0400	<VDCE0_base> + 1930	32
VDCE0	Table setting register R12 in gamma correction block	GAM_R_LUT12	R/W	H'0400 0400	<VDCE0_base> + 1934	32
VDCE0	Table setting register R13 in gamma correction block	GAM_R_LUT13	R/W	H'0400 0400	<VDCE0_base> + 1938	32
VDCE0	Table setting register R14 in gamma correction block	GAM_R_LUT14	R/W	H'0400 0400	<VDCE0_base> + 193C	32
VDCE0	Table setting register R15 in gamma correction block	GAM_R_LUT15	R/W	H'0400 0400	<VDCE0_base> + 1940	32
VDCE0	Table setting register R16 in gamma correction block	GAM_R_LUT16	R/W	H'0400 0400	<VDCE0_base> + 1944	32
VDCE0	Area setting register R1 in gamma correction block	GAM_R_AREA1	R/W	H'0008 1018	<VDCE0_base> + 1948	32
VDCE0	Area setting register R2 in gamma correction block	GAM_R_AREA2	R/W	H'2028 3038	<VDCE0_base> + 194C	32

Table 38.162 Gamma Correction Block Register Configuration (Channel 0) (4/4)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Area setting register R3 in gamma correction block	GAM_R_AREA3	R/W	H'4048 5058	<VDCE0_base> + 1950	32
VDCE0	Area setting register R4 in gamma correction block	GAM_R_AREA4	R/W	H'6068 7078	<VDCE0_base> + 1954	32
VDCE0	Area setting register R5 in gamma correction block	GAM_R_AREA5	R/W	H'8088 9098	<VDCE0_base> + 1958	32
VDCE0	Area setting register R6 in gamma correction block	GAM_R_AREA6	R/W	H'A0A8 B0B8	<VDCE0_base> + 195C	32
VDCE0	Area setting register R7 in gamma correction block	GAM_R_AREA7	R/W	H'C0C8 D0D8	<VDCE0_base> + 1960	32
VDCE0	Area setting register R8 in gamma correction block	GAM_R_AREA8	R/W	H'E0E8 F0F8	<VDCE0_base> + 1964	32

Table 38.163 TCON Block Register Configuration (Channel 0)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	TCON register update control register	TCON_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1980	32
VDCE0	TCON reference timing setting register	TCON_TIM	R/W	H'0190 0000	<VDCE0_base> + 1984	32
VDCE0	TCON vertical timing setting register A1	TCON_TIM_STVA1	R/W	H'0000 0004	<VDCE0_base> + 1988	32
VDCE0	TCON vertical timing setting register A2	TCON_TIM_STVA2	R/W	H'0000 0010	<VDCE0_base> + 198C	32
VDCE0	TCON vertical timing setting register B1	TCON_TIM_STVB1	R/W	H'0046 03C0	<VDCE0_base> + 1990	32
VDCE0	TCON vertical timing setting register B2	TCON_TIM_STVB2	R/W	H'0000 0001	<VDCE0_base> + 1994	32
VDCE0	TCON horizontal timing setting register STH1	TCON_TIM_STH1	R/W	H'0000 0060	<VDCE0_base> + 1998	32
VDCE0	TCON horizontal timing setting register STH2	TCON_TIM_STH2	R/W	H'0000 0012	<VDCE0_base> + 199C	32
VDCE0	TCON horizontal timing setting register STB1	TCON_TIM_STB1	R/W	H'0090 0280	<VDCE0_base> + 19A0	32
VDCE0	TCON horizontal timing setting register STB2	TCON_TIM_STB2	R/W	H'0000 0007	<VDCE0_base> + 19A4	32
VDCE0	TCON horizontal timing setting register CPV1	TCON_TIM_CPV1	R/W	H'0000 0000	<VDCE0_base> + 19A8	32
VDCE0	TCON horizontal timing setting register CPV2	TCON_TIM_CPV2	R/W	H'0000 0004	<VDCE0_base> + 19AC	32
VDCE0	TCON horizontal timing setting register POLA1	TCON_TIM_POLA1	R/W	H'0000 0000	<VDCE0_base> + 19B0	32
VDCE0	TCON horizontal timing setting register POLA2	TCON_TIM_POLA2	R/W	H'0000 1005	<VDCE0_base> + 19B4	32
VDCE0	TCON horizontal timing setting register POLB1	TCON_TIM_POLB1	R/W	H'0000 0000	<VDCE0_base> + 19B8	32
VDCE0	TCON horizontal timing setting register POLB2	TCON_TIM_POLB2	R/W	H'0000 1006	<VDCE0_base> + 19BC	32
VDCE0	TCON data enable polarity setting register	TCON_TIM_DE	R/W	H'0000 0000	<VDCE0_base> + 19C0	32

Table 38.164 Output Controller Register Configuration (Channel 0)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Register update control register in output controller	OUT_UPDATE	R/WC1	H'0000 0000	<VDCE0_base> + 1A00	32
VDCE0	Output interface register	OUT_SET	R/W	H'001F 0000	<VDCE0_base> + 1A04	32
VDCE0	Brightness (DC) correction register 1	OUT_BRIGHT1	R/W	H'0000 0200	<VDCE0_base> + 1A08	32
VDCE0	Brightness (DC) correction register 2	OUT_BRIGHT2	R/W	H'0200 0200	<VDCE0_base> + 1A0C	32
VDCE0	Contrast (gain) correction register	OUT_CONTRAST	R/W	H'0080 8080	<VDCE0_base> + 1A10	32
VDCE0	Panel dither register	OUT_PDTHA	R/W	H'0000 3021	<VDCE0_base> + 1A14	32
VDCE0	Output phase control register	OUT_CLK_PHASE	R/W	H'0000 0000	<VDCE0_base> + 1A24	32

Table 38.165 Gamma Correction Block Register Configuration (Channel 1) (1/4)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Register update control register G in gamma correction block	GAM_G_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1800	32
VDCE1	Function switch register in gamma correction block	GAM_SW	R/W	H'0000 0000	<VDCE1_base> + 1804	32
VDCE1	Table setting register G1 in gamma correction block	GAM_G_LUT1	R/W	H'0400 0400	<VDCE1_base> + 1808	32
VDCE1	Table setting register G2 in gamma correction block	GAM_G_LUT2	R/W	H'0400 0400	<VDCE1_base> + 180C	32
VDCE1	Table setting register G3 in gamma correction block	GAM_G_LUT3	R/W	H'0400 0400	<VDCE1_base> + 1810	32
VDCE1	Table setting register G4 in gamma correction block	GAM_G_LUT4	R/W	H'0400 0400	<VDCE1_base> + 1814	32
VDCE1	Table setting register G5 in gamma correction block	GAM_G_LUT5	R/W	H'0400 0400	<VDCE1_base> + 1818	32
VDCE1	Table setting register G6 in gamma correction block	GAM_G_LUT6	R/W	H'0400 0400	<VDCE1_base> + 181C	32
VDCE1	Table setting register G7 in gamma correction block	GAM_G_LUT7	R/W	H'0400 0400	<VDCE1_base> + 1820	32
VDCE1	Table setting register G8 in gamma correction block	GAM_G_LUT8	R/W	H'0400 0400	<VDCE1_base> + 1824	32
VDCE1	Table setting register G9 in gamma correction block	GAM_G_LUT9	R/W	H'0400 0400	<VDCE1_base> + 1828	32
VDCE1	Table setting register G10 in gamma correction block	GAM_G_LUT10	R/W	H'0400 0400	<VDCE1_base> + 182C	32
VDCE1	Table setting register G11 in gamma correction block	GAM_G_LUT11	R/W	H'0400 0400	<VDCE1_base> + 1830	32
VDCE1	Table setting register G12 in gamma correction block	GAM_G_LUT12	R/W	H'0400 0400	<VDCE1_base> + 1834	32
VDCE1	Table setting register G13 in gamma correction block	GAM_G_LUT13	R/W	H'0400 0400	<VDCE1_base> + 1838	32
VDCE1	Table setting register G14 in gamma correction block	GAM_G_LUT14	R/W	H'0400 0400	<VDCE1_base> + 183C	32
VDCE1	Table setting register G15 in gamma correction block	GAM_G_LUT15	R/W	H'0400 0400	<VDCE1_base> + 1840	32

Table 38.165 Gamma Correction Block Register Configuration (Channel 1) (2/4)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Table setting register G16 in gamma correction block	GAM_G_LUT16	R/W	H'0400 0400	<VDCE1_base> + 1844	32
VDCE1	Area setting register G1 in gamma correction block	GAM_G_AREA1	R/W	H'0008 1018	<VDCE1_base> + 1848	32
VDCE1	Area setting register G2 in gamma correction block	GAM_G_AREA2	R/W	H'2028 3038	<VDCE1_base> + 184C	32
VDCE1	Area setting register G3 in gamma correction block	GAM_G_AREA3	R/W	H'4048 5058	<VDCE1_base> + 1850	32
VDCE1	Area setting register G4 in gamma correction block	GAM_G_AREA4	R/W	H'6068 7078	<VDCE1_base> + 1854	32
VDCE1	Area setting register G5 in gamma correction block	GAM_G_AREA5	R/W	H'8088 9098	<VDCE1_base> + 1858	32
VDCE1	Area setting register G6 in gamma correction block	GAM_G_AREA6	R/W	H'A0A8 B0B8	<VDCE1_base> + 185C	32
VDCE1	Area setting register G7 in gamma correction block	GAM_G_AREA7	R/W	H'C0C8 D0D8	<VDCE1_base> + 1860	32
VDCE1	Area setting register G8 in gamma correction block	GAM_G_AREA8	R/W	H'E0E8 F0F8	<VDCE1_base> + 1864	32
VDCE1	Register update control register B in gamma correction block	GAM_B_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1880	32
VDCE1	Table setting register B1 in gamma correction block	GAM_B_LUT1	R/W	H'0400 0400	<VDCE1_base> + 1888	32
VDCE1	Table setting register B2 in gamma correction block	GAM_B_LUT2	R/W	H'0400 0400	<VDCE1_base> + 188C	32
VDCE1	Table setting register B3 in gamma correction block	GAM_B_LUT3	R/W	H'0400 0400	<VDCE1_base> + 1890	32
VDCE1	Table setting register B4 in gamma correction block	GAM_B_LUT4	R/W	H'0400 0400	<VDCE1_base> + 1894	32
VDCE1	Table setting register B5 in gamma correction block	GAM_B_LUT5	R/W	H'0400 0400	<VDCE1_base> + 1898	32
VDCE1	Table setting register B6 in gamma correction block	GAM_B_LUT6	R/W	H'0400 0400	<VDCE1_base> + 189C	32
VDCE1	Table setting register B7 in gamma correction block	GAM_B_LUT7	R/W	H'0400 0400	<VDCE1_base> + 18A0	32
VDCE1	Table setting register B8 in gamma correction block	GAM_B_LUT8	R/W	H'0400 0400	<VDCE1_base> + 18A4	32
VDCE1	Table setting register B9 in gamma correction block	GAM_B_LUT9	R/W	H'0400 0400	<VDCE1_base> + 18A8	32
VDCE1	Table setting register B10 in gamma correction block	GAM_B_LUT10	R/W	H'0400 0400	<VDCE1_base> + 18AC	32
VDCE1	Table setting register B11 in gamma correction block	GAM_B_LUT11	R/W	H'0400 0400	<VDCE1_base> + 18B0	32
VDCE1	Table setting register B12 in gamma correction block	GAM_B_LUT12	R/W	H'0400 0400	<VDCE1_base> + 18B4	32
VDCE1	Table setting register B13 in gamma correction block	GAM_B_LUT13	R/W	H'0400 0400	<VDCE1_base> + 18B8	32
VDCE1	Table setting register B14 in gamma correction block	GAM_B_LUT14	R/W	H'0400 0400	<VDCE1_base> + 18BC	32
VDCE1	Table setting register B15 in gamma correction block	GAM_B_LUT15	R/W	H'0400 0400	<VDCE1_base> + 18C0	32
VDCE1	Table setting register B16 in gamma correction block	GAM_B_LUT16	R/W	H'0400 0400	<VDCE1_base> + 18C4	32

Table 38.165 Gamma Correction Block Register Configuration (Channel 1) (3/4)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Area setting register B1 in gamma correction block	GAM_B_AREA1	R/W	H'0008 1018	<VDCE1_base> + 18C8	32
VDCE1	Area setting register B2 in gamma correction block	GAM_B_AREA2	R/W	H'2028 3038	<VDCE1_base> + 18CC	32
VDCE1	Area setting register B3 in gamma correction block	GAM_B_AREA3	R/W	H'4048 5058	<VDCE1_base> + 18D0	32
VDCE1	Area setting register B4 in gamma correction block	GAM_B_AREA4	R/W	H'6068 7078	<VDCE1_base> + 18D4	32
VDCE1	Area setting register B5 in gamma correction block	GAM_B_AREA5	R/W	H'8088 9098	<VDCE1_base> + 18D8	32
VDCE1	Area setting register B6 in gamma correction block	GAM_B_AREA6	R/W	H'A0A8 B0B8	<VDCE1_base> + 18DC	32
VDCE1	Area setting register B7 in gamma correction block	GAM_B_AREA7	R/W	H'C0C8 D0D8	<VDCE1_base> + 18E0	32
VDCE1	Area setting register B8 in gamma correction block	GAM_B_AREA8	R/W	H'E0E8 F0F8	<VDCE1_base> + 18E4	32
VDCE1	Register update control register R in gamma correction block	GAM_R_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1900	32
VDCE1	Table setting register R1 in gamma correction block	GAM_R_LUT1	R/W	H'0400 0400	<VDCE1_base> + 1908	32
VDCE1	Table setting register R2 in gamma correction block	GAM_R_LUT2	R/W	H'0400 0400	<VDCE1_base> + 190C	32
VDCE1	Table setting register R3 in gamma correction block	GAM_R_LUT3	R/W	H'0400 0400	<VDCE1_base> + 1910	32
VDCE1	Table setting register R4 in gamma correction block	GAM_R_LUT4	R/W	H'0400 0400	<VDCE1_base> + 1914	32
VDCE1	Table setting register R5 in gamma correction block	GAM_R_LUT5	R/W	H'0400 0400	<VDCE1_base> + 1918	32
VDCE1	Table setting register R6 in gamma correction block	GAM_R_LUT6	R/W	H'0400 0400	<VDCE1_base> + 191C	32
VDCE1	Table setting register R7 in gamma correction block	GAM_R_LUT7	R/W	H'0400 0400	<VDCE1_base> + 1920	32
VDCE1	Table setting register R8 in gamma correction block	GAM_R_LUT8	R/W	H'0400 0400	<VDCE1_base> + 1924	32
VDCE1	Table setting register R9 in gamma correction block	GAM_R_LUT9	R/W	H'0400 0400	<VDCE1_base> + 1928	32
VDCE1	Table setting register R10 in gamma correction block	GAM_R_LUT10	R/W	H'0400 0400	<VDCE1_base> + 192C	32
VDCE1	Table setting register R11 in gamma correction block	GAM_R_LUT11	R/W	H'0400 0400	<VDCE1_base> + 1930	32
VDCE1	Table setting register R12 in gamma correction block	GAM_R_LUT12	R/W	H'0400 0400	<VDCE1_base> + 1934	32
VDCE1	Table setting register R13 in gamma correction block	GAM_R_LUT13	R/W	H'0400 0400	<VDCE1_base> + 1938	32
VDCE1	Table setting register R14 in gamma correction block	GAM_R_LUT14	R/W	H'0400 0400	<VDCE1_base> + 193C	32
VDCE1	Table setting register R15 in gamma correction block	GAM_R_LUT15	R/W	H'0400 0400	<VDCE1_base> + 1940	32
VDCE1	Table setting register R16 in gamma correction block	GAM_R_LUT16	R/W	H'0400 0400	<VDCE1_base> + 1944	32
VDCE1	Area setting register R1 in gamma correction block	GAM_R_AREA1	R/W	H'0008 1018	<VDCE1_base> + 1948	32

Table 38.165 Gamma Correction Block Register Configuration (Channel 1) (4/4)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Area setting register R2 in gamma correction block	GAM_R_AREA2	R/W	H'2028 3038	<VDCE1_base> + 194C	32
VDCE1	Area setting register R3 in gamma correction block	GAM_R_AREA3	R/W	H'4048 5058	<VDCE1_base> + 1950	32
VDCE1	Area setting register R4 in gamma correction block	GAM_R_AREA4	R/W	H'6068 7078	<VDCE1_base> + 1954	32
VDCE1	Area setting register R5 in gamma correction block	GAM_R_AREA5	R/W	H'8088 9098	<VDCE1_base> + 1958	32
VDCE1	Area setting register R6 in gamma correction block	GAM_R_AREA6	R/W	H'A0A8 B0B8	<VDCE1_base> + 195C	32
VDCE1	Area setting register R7 in gamma correction block	GAM_R_AREA7	R/W	H'C0C8 D0D8	<VDCE1_base> + 1960	32
VDCE1	Area setting register R8 in gamma correction block	GAM_R_AREA8	R/W	H'E0E8 F0F8	<VDCE1_base> + 1964	32

Table 38.166 TCON Block Register Configuration (Channel 1)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	TCON register update control register	TCON_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1980	32
VDCE1	TCON reference timing setting register	TCON_TIM	R/W	H'0190 0000	<VDCE1_base> + 1984	32
VDCE1	TCON vertical timing setting register A1	TCON_TIM_STVA1	R/W	H'0000 0004	<VDCE1_base> + 1988	32
VDCE1	TCON vertical timing setting register A2	TCON_TIM_STVA2	R/W	H'0000 0010	<VDCE1_base> + 198C	32
VDCE1	TCON vertical timing setting register B1	TCON_TIM_STVB1	R/W	H'0046 03C0	<VDCE1_base> + 1990	32
VDCE1	TCON vertical timing setting register B2	TCON_TIM_STVB2	R/W	H'0000 0001	<VDCE1_base> + 1994	32
VDCE1	TCON horizontal timing setting register STH1	TCON_TIM_STH1	R/W	H'0000 0060	<VDCE1_base> + 1998	32
VDCE1	TCON horizontal timing setting register STH2	TCON_TIM_STH2	R/W	H'0000 0012	<VDCE1_base> + 199C	32
VDCE1	TCON horizontal timing setting register STB1	TCON_TIM_STB1	R/W	H'0090 0280	<VDCE1_base> + 19A0	32
VDCE1	TCON horizontal timing setting register STB2	TCON_TIM_STB2	R/W	H'0000 0007	<VDCE1_base> + 19A4	32
VDCE1	TCON horizontal timing setting register CPV1	TCON_TIM_CPV1	R/W	H'0000 0000	<VDCE1_base> + 19A8	32
VDCE1	TCON horizontal timing setting register CPV2	TCON_TIM_CPV2	R/W	H'0000 0004	<VDCE1_base> + 19AC	32
VDCE1	TCON horizontal timing setting register POLA1	TCON_TIM_POLA1	R/W	H'0000 0000	<VDCE1_base> + 19B0	32
VDCE1	TCON horizontal timing setting register POLA2	TCON_TIM_POLA2	R/W	H'0000 1005	<VDCE1_base> + 19B4	32
VDCE1	TCON horizontal timing setting register POLB1	TCON_TIM_POLB1	R/W	H'0000 0000	<VDCE1_base> + 19B8	32
VDCE1	TCON horizontal timing setting register POLB2	TCON_TIM_POLB2	R/W	H'0000 1006	<VDCE1_base> + 19BC	32
VDCE1	TCON data enable polarity setting register	TCON_TIM_DE	R/W	H'0000 0000	<VDCE1_base> + 19C0	32

Table 38.167 Output Controller Register Configuration (Channel 1)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Register update control register in output controller	OUT_UPDATE	R/WC1	H'0000 0000	<VDCE1_base> + 1A00	32
VDCE1	Output interface register	OUT_SET	R/W	H'001F 0000	<VDCE1_base> + 1A04	32
VDCE1	Brightness (DC) correction register 1	OUT_BRIGHT1	R/W	H'0000 0200	<VDCE1_base> + 1A08	32
VDCE1	Brightness (DC) correction register 2	OUT_BRIGHT2	R/W	H'0200 0200	<VDCE1_base> + 1A0C	32
VDCE1	Contrast (gain) correction register	OUT_CONTRAST	R/W	H'0080 8080	<VDCE1_base> + 1A10	32
VDCE1	Panel dither register	OUT_PDTHA	R/W	H'0000 3021	<VDCE1_base> + 1A14	32
VDCE1	Output phase control register	OUT_CLK_PHASE	R/W	H'0000 0000	<VDCE1_base> + 1A24	32

NOTE

Register access sizes other than defined in the table above are not supported.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

38.8.2.1 Register Update Control Register G in Gamma Correction Block (GAM_G_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GAM_G_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GAM_G_VEN	0	R/WC1	Gamma Correction (G) Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.8.2.2 Function Switch Register in Gamma Correction Block (GAM_SW)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GAM_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GAM_ON	0	R/W	Gamma Correction On/Off Control 0: Off 1: On **Note:** For D1L2(H) devices this bit must be set to 0.

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

38.8.2.3 Table Setting Register G1 to G16 in Gamma Correction Block (GAM_G_LUT1 to GAM_G_LUT16)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GAM_G_GAIN_xx[10:0]										
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GAM_G_GAIN_yy[10:0]										
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	*	1024	R/W	GAM_G_LUT1: Gain Adjustment of Area 0 of G Signal GAM_G_LUT2: Gain Adjustment of Area 2 of G Signal GAM_G_LUT3: Gain Adjustment of Area 4 of G Signal GAM_G_LUT4: Gain Adjustment of Area 6 of G Signal GAM_G_LUT5: Gain Adjustment of Area 8 of G Signal GAM_G_LUT6: Gain Adjustment of Area 10 of G Signal GAM_G_LUT7: Gain Adjustment of Area 12 of G Signal GAM_G_LUT8: Gain Adjustment of Area 14 of G Signal GAM_G_LUT9: Gain Adjustment of Area 16 of G Signal GAM_G_LUT10: Gain Adjustment of Area 18 of G Signal GAM_G_LUT11: Gain Adjustment of Area 20 of G Signal GAM_G_LUT12: Gain Adjustment of Area 22 of G Signal GAM_G_LUT13: Gain Adjustment of Area 24 of G Signal GAM_G_LUT14: Gain Adjustment of Area 26 of G Signal GAM_G_LUT15: Gain Adjustment of Area 28 of G Signal GAM_G_LUT16: Gain Adjustment of Area 30 of G Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])
26 to 16	*	1024	R/W	*: Bit Name GAM_G_LUT1: GAM_G_GAIN_00[10:0] GAM_G_LUT2: GAM_G_GAIN_02[10:0] GAM_G_LUT3: GAM_G_GAIN_04[10:0] GAM_G_LUT4: GAM_G_GAIN_06[10:0] GAM_G_LUT5: GAM_G_GAIN_08[10:0] GAM_G_LUT6: GAM_G_GAIN_10[10:0] GAM_G_LUT7: GAM_G_GAIN_12[10:0] GAM_G_LUT8: GAM_G_GAIN_14[10:0] GAM_G_LUT9: GAM_G_GAIN_16[10:0] GAM_G_LUT10: GAM_G_GAIN_18[10:0] GAM_G_LUT11: GAM_G_GAIN_20[10:0] GAM_G_LUT12: GAM_G_GAIN_22[10:0] GAM_G_LUT13: GAM_G_GAIN_24[10:0] GAM_G_LUT14: GAM_G_GAIN_26[10:0] GAM_G_LUT15: GAM_G_GAIN_28[10:0] GAM_G_LUT16: GAM_G_GAIN_30[10:0]

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	*	1024	R/W	GAM_G_LUT1: Gain Adjustment of Area 1 of G Signal GAM_G_LUT2: Gain Adjustment of Area 3 of G Signal GAM_G_LUT3: Gain Adjustment of Area 5 of G Signal GAM_G_LUT4: Gain Adjustment of Area 7 of G Signal GAM_G_LUT5: Gain Adjustment of Area 9 of G Signal GAM_G_LUT6: Gain Adjustment of Area 11 of G Signal GAM_G_LUT7: Gain Adjustment of Area 13 of G Signal GAM_G_LUT8: Gain Adjustment of Area 15 of G Signal GAM_G_LUT9: Gain Adjustment of Area 17 of G Signal GAM_G_LUT10: Gain Adjustment of Area 19 of G Signal GAM_G_LUT11: Gain Adjustment of Area 21 of G Signal GAM_G_LUT12: Gain Adjustment of Area 23 of G Signal GAM_G_LUT13: Gain Adjustment of Area 25 of G Signal GAM_G_LUT14: Gain Adjustment of Area 27 of G Signal GAM_G_LUT15: Gain Adjustment of Area 29 of G Signal GAM_G_LUT16: Gain Adjustment of Area 31 of G Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])
10 to 0	*	1024	R/W	*: Bit Name GAM_G_LUT1: GAM_G_GAIN_01[10:0] GAM_G_LUT2: GAM_G_GAIN_03[10:0] GAM_G_LUT3: GAM_G_GAIN_05[10:0] GAM_G_LUT4: GAM_G_GAIN_07[10:0] GAM_G_LUT5: GAM_G_GAIN_09[10:0] GAM_G_LUT6: GAM_G_GAIN_11[10:0] GAM_G_LUT7: GAM_G_GAIN_13[10:0] GAM_G_LUT8: GAM_G_GAIN_15[10:0] GAM_G_LUT9: GAM_G_GAIN_17[10:0] GAM_G_LUT10: GAM_G_GAIN_19[10:0] GAM_G_LUT11: GAM_G_GAIN_21[10:0] GAM_G_LUT12: GAM_G_GAIN_23[10:0] GAM_G_LUT13: GAM_G_GAIN_25[10:0] GAM_G_LUT14: GAM_G_GAIN_27[10:0] GAM_G_LUT15: GAM_G_GAIN_29[10:0] GAM_G_LUT16: GAM_G_GAIN_31[10:0]

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

38.8.2.4 Area Setting Register G1 in Gamma Correction Block (GAM_G_AREA1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GAM_G_TH_01[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_02[7:0]								GAM_G_TH_03[7:0]							
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GAM_G_TH_01[7:0]	8	R/W	Start Threshold of Area 1 of G Signal Unsigned (0 to 255 [LSB]) 0 < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_02[7:0]	16	R/W	Start Threshold of Area 2 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_03[7:0]	24	R/W	Start Threshold of Area 3 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

38.8.2.5 Area Setting Register G2 in Gamma Correction Block (GAM_G_AREA2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_04[7:0]								GAM_G_TH_05[7:0]							
Initial value:	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_06[7:0]								GAM_G_TH_07[7:0]							
Initial value:	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_04[7:0]	32	R/W	Start Threshold of Area 4 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_05[7:0]	40	R/W	Start Threshold of Area 5 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_06[7:0]	48	R/W	Start Threshold of Area 6 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_07[7:0]	56	R/W	Start Threshold of Area 7 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

38.8.2.6 Area Setting Register G3 in Gamma Correction Block (GAM_G_AREA3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_08[7:0]								GAM_G_TH_09[7:0]							
Initial value:	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_10[7:0]								GAM_G_TH_11[7:0]							
Initial value:	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_08[7:0]	64	R/W	Start Threshold of Area 8 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_09[7:0]	72	R/W	Start Threshold of Area 9 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_10[7:0]	80	R/W	Start Threshold of Area 10 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_11[7:0]	88	R/W	Start Threshold of Area 11 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

38.8.2.7 Area Setting Register G4 in Gamma Correction Block (GAM_G_AREA4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_12[7:0]								GAM_G_TH_13[7:0]							
Initial value:	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_14[7:0]								GAM_G_TH_15[7:0]							
Initial value:	0	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_12[7:0]	96	R/W	Start Threshold of Area 12 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_13[7:0]	104	R/W	Start Threshold of Area 13 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_14[7:0]	112	R/W	Start Threshold of Area 14 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_15[7:0]	120	R/W	Start Threshold of Area 15 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

38.8.2.8 Area Setting Register G5 in Gamma Correction Block (GAM_G_AREA5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_16[7:0]								GAM_G_TH_17[7:0]							
Initial value:	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_18[7:0]								GAM_G_TH_19[7:0]							
Initial value:	1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_16 [7:0]	128	R/W	Start Threshold of Area 16 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_17 [7:0]	136	R/W	Start Threshold of Area 17 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_18 [7:0]	144	R/W	Start Threshold of Area 18 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_19 [7:0]	152	R/W	Start Threshold of Area 19 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

38.8.2.9 Area Setting Register G6 in Gamma Correction Block (GAM_G_AREA6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_20[7:0]								GAM_G_TH_21[7:0]							
Initial value:	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_22[7:0]								GAM_G_TH_23[7:0]							
Initial value:	1	0	1	1	0	0	0	0	1	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_20 [7:0]	160	R/W	Start Threshold of Area 20 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_21 [7:0]	168	R/W	Start Threshold of Area 21 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_22 [7:0]	176	R/W	Start Threshold of Area 22 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_23 [7:0]	184	R/W	Start Threshold of Area 23 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

38.8.2.10 Area Setting Register G7 in Gamma Correction Block (GAM_G_AREA7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_24[7:0]								GAM_G_TH_25[7:0]							
Initial value:	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_26[7:0]								GAM_G_TH_27[7:0]							
Initial value:	1	1	0	1	0	0	0	0	1	1	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_24 [7:0]	192	R/W	Start Threshold of Area 24 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_25 [7:0]	200	R/W	Start Threshold of Area 25 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_26 [7:0]	208	R/W	Start Threshold of Area 26 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_27 [7:0]	216	R/W	Start Threshold of Area 27 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

38.8.2.11 Area Setting Register G8 in Gamma Correction Block (GAM_G_AREA8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_G_TH_28[7:0]								GAM_G_TH_29[7:0]							
Initial value:	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_G_TH_30[7:0]								GAM_G_TH_31[7:0]							
Initial value:	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_G_TH_28 [7:0]	224	R/W	Start Threshold of Area 28 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_G_TH_29 [7:0]	232	R/W	Start Threshold of Area 29 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_G_TH_30 [7:0]	240	R/W	Start Threshold of Area 30 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_G_TH_31 [7:0]	248	R/W	Start Threshold of Area 31 of G Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area ≤ 255

Note: This register is updated when GAM_G_VEN in GAM_G_UPDATE is 1.

38.8.2.12 Register Update Control Register B in Gamma Correction Block (GAM_B_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GAM_B_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GAM_B_VEN	0	R/WC1	Gamma Correction (B) Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.8.2.13 Table Setting Register B1 to B16 in Gamma Correction Block (GAM_B_LUT1 to GAM_B_LUT16)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GAM_B_GAIN_xx[10:0]										
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GAM_B_GAIN_yy[10:0]										
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	*	1024	R/W	GAM_B_LUT1: Gain Adjustment of Area 0 of B Signal GAM_B_LUT2: Gain Adjustment of Area 2 of B Signal GAM_B_LUT3: Gain Adjustment of Area 4 of B Signal GAM_B_LUT4: Gain Adjustment of Area 6 of B Signal GAM_B_LUT5: Gain Adjustment of Area 8 of B Signal GAM_B_LUT6: Gain Adjustment of Area 10 of B Signal GAM_B_LUT7: Gain Adjustment of Area 12 of B Signal GAM_B_LUT8: Gain Adjustment of Area 14 of B Signal GAM_B_LUT9: Gain Adjustment of Area 16 of B Signal GAM_B_LUT10: Gain Adjustment of Area 18 of B Signal GAM_B_LUT11: Gain Adjustment of Area 20 of B Signal GAM_B_LUT12: Gain Adjustment of Area 22 of B Signal GAM_B_LUT13: Gain Adjustment of Area 24 of B Signal GAM_B_LUT14: Gain Adjustment of Area 26 of B Signal GAM_B_LUT15: Gain Adjustment of Area 28 of B Signal GAM_B_LUT16: Gain Adjustment of Area 30 of B Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])
26 to 16	*	1024	R/W	*: Bit Name GAM_B_LUT1: GAM_B_GAIN_00[10:0] GAM_B_LUT2: GAM_B_GAIN_02[10:0] GAM_B_LUT3: GAM_B_GAIN_04[10:0] GAM_B_LUT4: GAM_B_GAIN_06[10:0] GAM_B_LUT5: GAM_B_GAIN_08[10:0] GAM_B_LUT6: GAM_B_GAIN_10[10:0] GAM_B_LUT7: GAM_B_GAIN_12[10:0] GAM_B_LUT8: GAM_B_GAIN_14[10:0] GAM_B_LUT9: GAM_B_GAIN_16[10:0] GAM_B_LUT10: GAM_B_GAIN_18[10:0] GAM_B_LUT11: GAM_B_GAIN_20[10:0] GAM_B_LUT12: GAM_B_GAIN_22[10:0] GAM_B_LUT13: GAM_B_GAIN_24[10:0] GAM_B_LUT14: GAM_B_GAIN_26[10:0] GAM_B_LUT15: GAM_B_GAIN_28[10:0] GAM_B_LUT16: GAM_B_GAIN_30[10:0]

Bit	Bit Name	Initial Value	R/W	Description
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	*	1024	R/W	GAM_B_LUT1: Gain Adjustment of Area 1 of B Signal GAM_B_LUT2: Gain Adjustment of Area 3 of B Signal GAM_B_LUT3: Gain Adjustment of Area 5 of B Signal GAM_B_LUT4: Gain Adjustment of Area 7 of B Signal GAM_B_LUT5: Gain Adjustment of Area 9 of B Signal GAM_B_LUT6: Gain Adjustment of Area 11 of B Signal GAM_B_LUT7: Gain Adjustment of Area 13 of B Signal GAM_B_LUT8: Gain Adjustment of Area 15 of B Signal GAM_B_LUT9: Gain Adjustment of Area 17 of B Signal GAM_B_LUT10: Gain Adjustment of Area 19 of B Signal GAM_B_LUT11: Gain Adjustment of Area 21 of B Signal GAM_B_LUT12: Gain Adjustment of Area 23 of B Signal GAM_B_LUT13: Gain Adjustment of Area 25 of B Signal GAM_B_LUT14: Gain Adjustment of Area 27 of B Signal GAM_B_LUT15: Gain Adjustment of Area 29 of B Signal GAM_B_LUT16: Gain Adjustment of Area 31 of B Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])
10 to 0	*	1024	R/W	*: Bit Name GAM_B_LUT1: GAM_B_GAIN_01[10:0] GAM_B_LUT2: GAM_B_GAIN_03[10:0] GAM_B_LUT3: GAM_B_GAIN_05[10:0] GAM_B_LUT4: GAM_B_GAIN_07[10:0] GAM_B_LUT5: GAM_B_GAIN_09[10:0] GAM_B_LUT6: GAM_B_GAIN_11[10:0] GAM_B_LUT7: GAM_B_GAIN_13[10:0] GAM_B_LUT8: GAM_B_GAIN_15[10:0] GAM_B_LUT9: GAM_B_GAIN_17[10:0] GAM_B_LUT10: GAM_B_GAIN_19[10:0] GAM_B_LUT11: GAM_B_GAIN_21[10:0] GAM_B_LUT12: GAM_B_GAIN_23[10:0] GAM_B_LUT13: GAM_B_GAIN_25[10:0] GAM_B_LUT14: GAM_B_GAIN_27[10:0] GAM_B_LUT15: GAM_B_GAIN_29[10:0] GAM_B_LUT16: GAM_B_GAIN_31[10:0]

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

38.8.2.14 Area Setting Register B1 in Gamma Correction Block (GAM_B_AREA1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GAM_B_TH_01[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_02[7:0]								GAM_B_TH_03[7:0]							
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GAM_B_TH_01[7:0]	8	R/W	Start Threshold of Area 1 of B Signal Unsigned (0 to 255 [LSB]) 0 < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_02[7:0]	16	R/W	Start Threshold of Area 2 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_03[7:0]	24	R/W	Start Threshold of Area 3 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

38.8.2.15 Area Setting Register B2 in Gamma Correction Block (GAM_B_AREA2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_04[7:0]								GAM_B_TH_05[7:0]							
Initial value:	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_06[7:0]								GAM_B_TH_07[7:0]							
Initial value:	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_04[7:0]	32	R/W	Start Threshold of Area 4 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_05[7:0]	40	R/W	Start Threshold of Area 5 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_06[7:0]	48	R/W	Start Threshold of Area 6 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_07[7:0]	56	R/W	Start Threshold of Area 7 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

38.8.2.16 Area Setting Register B3 in Gamma Correction Block (GAM_B_AREA3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_08[7:0]								GAM_B_TH_09[7:0]							
Initial value:	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_10[7:0]								GAM_B_TH_11[7:0]							
Initial value:	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_08[7:0]	64	R/W	Start Threshold of Area 8 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_09[7:0]	72	R/W	Start Threshold of Area 9 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_10[7:0]	80	R/W	Start Threshold of Area 10 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_11[7:0]	88	R/W	Start Threshold of Area 11 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

38.8.2.17 Area Setting Register B4 in Gamma Correction Block (GAM_B_AREA4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_12[7:0]								GAM_B_TH_13[7:0]							
Initial value:	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_14[7:0]								GAM_B_TH_15[7:0]							
Initial value:	0	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_12[7:0]	96	R/W	Start Threshold of Area 12 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_13[7:0]	104	R/W	Start Threshold of Area 13 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_14[7:0]	112	R/W	Start Threshold of Area 14 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_15[7:0]	120	R/W	Start Threshold of Area 15 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

38.8.2.18 Area Setting Register B5 in Gamma Correction Block (GAM_B_AREA5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_16[7:0]								GAM_B_TH_17[7:0]							
Initial value:	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_18[7:0]								GAM_B_TH_19[7:0]							
Initial value:	1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_16[7:0]	128	R/W	Start Threshold of Area 16 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_17[7:0]	136	R/W	Start Threshold of Area 17 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_18[7:0]	144	R/W	Start Threshold of Area 18 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_19[7:0]	152	R/W	Start Threshold of Area 19 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

38.8.2.19 Area Setting Register B6 in Gamma Correction Block (GAM_B_AREA6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_20[7:0]								GAM_B_TH_21[7:0]							
Initial value:	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_22[7:0]								GAM_B_TH_23[7:0]							
Initial value:	1	0	1	1	0	0	0	0	1	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_20[7:0]	160	R/W	Start Threshold of Area 20 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_21[7:0]	168	R/W	Start Threshold of Area 21 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_22[7:0]	176	R/W	Start Threshold of Area 22 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_23[7:0]	184	R/W	Start Threshold of Area 23 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

38.8.2.20 Area Setting Register B7 in Gamma Correction Block (GAM_B_AREA7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_24[7:0]								GAM_B_TH_25[7:0]							
Initial value:	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_26[7:0]								GAM_B_TH_27[7:0]							
Initial value:	1	1	0	1	0	0	0	0	1	1	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_24[7:0]	192	R/W	Start Threshold of Area 24 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_25[7:0]	200	R/W	Start Threshold of Area 25 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_26[7:0]	208	R/W	Start Threshold of Area 26 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_27[7:0]	216	R/W	Start Threshold of Area 27 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

38.8.2.21 Area Setting Register B8 in Gamma Correction Block (GAM_B_AREA8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_B_TH_28[7:0]								GAM_B_TH_29[7:0]							
Initial value:	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_B_TH_30[7:0]								GAM_B_TH_31[7:0]							
Initial value:	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_B_TH_28[7:0]	224	R/W	Start Threshold of Area 28 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_B_TH_29[7:0]	232	R/W	Start Threshold of Area 29 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_B_TH_30[7:0]	240	R/W	Start Threshold of Area 30 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_B_TH_31[7:0]	248	R/W	Start Threshold of Area 31 of B Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area ≤ 255

Note: This register is updated when GAM_B_VEN in GAM_B_UPDATE is 1.

38.8.2.22 Register Update Control Register R in Gamma Correction Block (GAM_R_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GAM_R_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GAM_R_VEN	0	R/WC1	Gamma Correction (R) Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.8.2.23 Table Setting Register R1 to R16 in Gamma Correction Block (GAM_R_LUT1 to GAM_R_LUT16)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	GAM_R_GAIN_xx[10:0]										
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	GAM_R_GAIN_yy[10:0]										
Initial value:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	*	1024	R/W	GAM_R_LUT1: Gain Adjustment of Area 0 of R Signal GAM_R_LUT2: Gain Adjustment of Area 2 of R Signal GAM_R_LUT3: Gain Adjustment of Area 4 of R Signal GAM_R_LUT4: Gain Adjustment of Area 6 of R Signal GAM_R_LUT5: Gain Adjustment of Area 8 of R Signal GAM_R_LUT6: Gain Adjustment of Area 10 of R Signal GAM_R_LUT7: Gain Adjustment of Area 12 of R Signal GAM_R_LUT8: Gain Adjustment of Area 14 of R Signal GAM_R_LUT9: Gain Adjustment of Area 16 of R Signal GAM_R_LUT10: Gain Adjustment of Area 18 of R Signal GAM_R_LUT11: Gain Adjustment of Area 20 of R Signal GAM_R_LUT12: Gain Adjustment of Area 22 of R Signal GAM_R_LUT13: Gain Adjustment of Area 24 of R Signal GAM_R_LUT14: Gain Adjustment of Area 26 of R Signal GAM_R_LUT15: Gain Adjustment of Area 28 of R Signal GAM_R_LUT16: Gain Adjustment of Area 30 of R Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])
26 to 16	*	1024	R/W	*: Bit Name GAM_R_LUT1: GAM_R_GAIN_00[10:0] GAM_R_LUT2: GAM_R_GAIN_02[10:0] GAM_R_LUT3: GAM_R_GAIN_04[10:0] GAM_R_LUT4: GAM_R_GAIN_06[10:0] GAM_R_LUT5: GAM_R_GAIN_08[10:0] GAM_R_LUT6: GAM_R_GAIN_10[10:0] GAM_R_LUT7: GAM_R_GAIN_12[10:0] GAM_R_LUT8: GAM_R_GAIN_14[10:0] GAM_R_LUT9: GAM_R_GAIN_16[10:0] GAM_R_LUT10: GAM_R_GAIN_18[10:0] GAM_R_LUT11: GAM_R_GAIN_20[10:0] GAM_R_LUT12: GAM_R_GAIN_22[10:0] GAM_R_LUT13: GAM_R_GAIN_24[10:0] GAM_R_LUT14: GAM_R_GAIN_26[10:0] GAM_R_LUT15: GAM_R_GAIN_28[10:0] GAM_R_LUT16: GAM_R_GAIN_30[10:0]
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
10 to 0	*	1024	R/W	GAM_R_LUT1: Gain Adjustment of Area 1 of R Signal GAM_R_LUT2: Gain Adjustment of Area 3 of R Signal GAM_R_LUT3: Gain Adjustment of Area 5 of R Signal GAM_R_LUT4: Gain Adjustment of Area 7 of R Signal GAM_R_LUT5: Gain Adjustment of Area 9 of R Signal GAM_R_LUT6: Gain Adjustment of Area 11 of R Signal GAM_R_LUT7: Gain Adjustment of Area 13 of R Signal GAM_R_LUT8: Gain Adjustment of Area 15 of R Signal GAM_R_LUT9: Gain Adjustment of Area 17 of R Signal GAM_R_LUT10: Gain Adjustment of Area 19 of R Signal GAM_R_LUT11: Gain Adjustment of Area 21 of R Signal GAM_R_LUT12: Gain Adjustment of Area 23 of R Signal GAM_R_LUT13: Gain Adjustment of Area 25 of R Signal GAM_R_LUT14: Gain Adjustment of Area 27 of R Signal GAM_R_LUT15: Gain Adjustment of Area 29 of R Signal GAM_R_LUT16: Gain Adjustment of Area 31 of R Signal Unsigned (0 to 2047 [LSB], 1024 [LSB] = 1.0 [times])
10 to 0	*	1024	R/W	*: Bit Name GAM_R_LUT1: GAM_R_GAIN_01[10:0] GAM_R_LUT2: GAM_R_GAIN_03[10:0] GAM_R_LUT3: GAM_R_GAIN_05[10:0] GAM_R_LUT4: GAM_R_GAIN_07[10:0] GAM_R_LUT5: GAM_R_GAIN_09[10:0] GAM_R_LUT6: GAM_R_GAIN_11[10:0] GAM_R_LUT7: GAM_R_GAIN_13[10:0] GAM_R_LUT8: GAM_R_GAIN_15[10:0] GAM_R_LUT9: GAM_R_GAIN_17[10:0] GAM_R_LUT10: GAM_R_GAIN_19[10:0] GAM_R_LUT11: GAM_R_GAIN_21[10:0] GAM_R_LUT12: GAM_R_GAIN_23[10:0] GAM_R_LUT13: GAM_R_GAIN_25[10:0] GAM_R_LUT14: GAM_R_GAIN_27[10:0] GAM_R_LUT15: GAM_R_GAIN_29[10:0] GAM_R_LUT16: GAM_R_GAIN_31[10:0]

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

38.8.2.24 Area Setting Register R1 in Gamma Correction Block (GAM_R_AREA1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	GAM_R_TH_01[7:0]							
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_02[7:0]								GAM_R_TH_03[7:0]							
Initial value:	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	GAM_R_TH_01[7:0]	8	R/W	Start Threshold of Area 1 of R Signal Unsigned (0 to 255 [LSB]) 0 < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_02[7:0]	16	R/W	Start Threshold of Area 2 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_03[7:0]	24	R/W	Start Threshold of Area 3 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

38.8.2.25 Area Setting Register R2 in Gamma Correction Block (GAM_R_AREA2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_04[7:0]								GAM_R_TH_05[7:0]							
Initial value:	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_06[7:0]								GAM_R_TH_07[7:0]							
Initial value:	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_04[7:0]	32	R/W	Start Threshold of Area 4 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_05[7:0]	40	R/W	Start Threshold of Area 5 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_06[7:0]	48	R/W	Start Threshold of Area 6 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_07[7:0]	56	R/W	Start Threshold of Area 7 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

38.8.2.26 Area Setting Register R3 in Gamma Correction Block (GAM_R_AREA3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_08[7:0]								GAM_R_TH_09[7:0]							
Initial value:	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_10[7:0]								GAM_R_TH_11[7:0]							
Initial value:	0	1	0	1	0	0	0	0	0	1	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_08[7:0]	64	R/W	Start Threshold of Area 8 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_09[7:0]	72	R/W	Start Threshold of Area 9 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_10[7:0]	80	R/W	Start Threshold of Area 10 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_11[7:0]	88	R/W	Start Threshold of Area 11 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

38.8.2.27 Area Setting Register R4 in Gamma Correction Block (GAM_R_AREA4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_12[7:0]								GAM_R_TH_13[7:0]							
Initial value:	0	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_14[7:0]								GAM_R_TH_15[7:0]							
Initial value:	0	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_12[7:0]	96	R/W	Start Threshold of Area 12 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_13[7:0]	104	R/W	Start Threshold of Area 13 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_14[7:0]	112	R/W	Start Threshold of Area 14 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_15[7:0]	120	R/W	Start Threshold of Area 15 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

38.8.2.28 Area Setting Register R5 in Gamma Correction Block (GAM_R_AREA5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_16[7:0]								GAM_R_TH_17[7:0]							
Initial value:	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_18[7:0]								GAM_R_TH_19[7:0]							
Initial value:	1	0	0	1	0	0	0	0	1	0	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_16[7:0]	128	R/W	Start Threshold of Area 16 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_17[7:0]	136	R/W	Start Threshold of Area 17 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_18[7:0]	144	R/W	Start Threshold of Area 18 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_19[7:0]	152	R/W	Start Threshold of Area 19 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

38.8.2.29 Area Setting Register R6 in Gamma Correction Block (GAM_R_AREA6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_20[7:0]								GAM_R_TH_21[7:0]							
Initial value:	1	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_22[7:0]								GAM_R_TH_23[7:0]							
Initial value:	1	0	1	1	0	0	0	0	1	0	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_20[7:0]	160	R/W	Start Threshold of Area 20 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_21[7:0]	168	R/W	Start Threshold of Area 21 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_22[7:0]	176	R/W	Start Threshold of Area 22 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_23[7:0]	184	R/W	Start Threshold of Area 23 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

38.8.2.30 Area Setting Register R7 in Gamma Correction Block (GAM_R_AREA7)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_24[7:0]								GAM_R_TH_25[7:0]							
Initial value:	1	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_26[7:0]								GAM_R_TH_27[7:0]							
Initial value:	1	1	0	1	0	0	0	0	1	1	0	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_24[7:0]	192	R/W	Start Threshold of Area 24 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_25[7:0]	200	R/W	Start Threshold of Area 25 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_26[7:0]	208	R/W	Start Threshold of Area 26 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_27[7:0]	216	R/W	Start Threshold of Area 27 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

38.8.2.31 Area Setting Register R8 in Gamma Correction Block (GAM_R_AREA8)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	GAM_R_TH_28[7:0]								GAM_R_TH_29[7:0]							
Initial value:	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	GAM_R_TH_30[7:0]								GAM_R_TH_31[7:0]							
Initial value:	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	GAM_R_TH_28[7:0]	224	R/W	Start Threshold of Area 28 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
23 to 16	GAM_R_TH_29[7:0]	232	R/W	Start Threshold of Area 29 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
15 to 8	GAM_R_TH_30[7:0]	240	R/W	Start Threshold of Area 30 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area < Threshold of next area
7 to 0	GAM_R_TH_31[7:0]	248	R/W	Start Threshold of Area 31 of R Signal Unsigned (0 to 255 [LSB]) Threshold of previous area < Threshold of current area ≤ 255

Note: This register is updated when GAM_R_VEN in GAM_R_UPDATE is 1.

38.8.2.32 TCON Register Update Control Register (TCON_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TCON_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 16	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
15 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TCON_VEN	0	R/WC1	LCD TCON Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.8.2.33 TCON Reference Timing Setting Register (TCON_TIM)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_HALF[10:0]										
Initial value:	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_OFFSET[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_HALF[10:0]	400	R/W	1/2fH Timing Specifies the clock count from the rising edge of the Hsync signal as the counting timing of horizontal counter.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_OFFSET[10:0]	0	R/W	Offset Hsync Signal Timing Sets the clock cycle count from the rising edge of the Hsync signal.

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.34 TCON Vertical Timing Setting Register A1 (TCON_TIM_STVA1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_STVA_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_STVA_VW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_STVA_VS [10:0]	0	R/W	STVA Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STVA_VS[10:0] from the rising edge of the Vsync signal (1/2fH cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_STVA_VW [10:0]	4	R/W	STVA Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STVA_VW[10:0] (1/2fH cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.35 TCON Vertical Timing Setting Register A2 (TCON_TIM_STVA2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TCON_STVA_INV	—	TCON_STVA_SEL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_STVA_INV	1	R/W	Polarity Inversion Control of STVA Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_STVA_SEL [2:0]	0	R/W	Output Signal Select for LCD_TCON0 Pin 0: STVA/VS all others: prohibited

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.36 TCON Vertical Timing Setting Register B1 (TCON_TIM_STVB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_STVB_VS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_STVB_VW[10:0]										
Initial value:	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_STVB_VS [10:0]	70	R/W	STVB Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STVB_VS[10:0] from the rising edge of the Vsync signal (1/2fH cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_STVB_VW [10:0]	960	R/W	STVB Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STVB_VW[10:0] (1/2fH cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.37 TCON Vertical Timing Setting Register B2 (TCON_TIM_STVB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	TCON_STVB_INV	—	TCON_STVB_SEL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W:	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_STVB_INV	0	R/W	Polarity Inversion Control of STVB Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_STVB_SEL [2:0]	1	R/W	Output Signal Select for LCD_TCON1 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.38 TCON Horizontal Timing Setting Register STH1 (TCON_TIM_STH1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_STH_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_STH_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_STH_HS [10:0]	0	R/W	STH Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STH_HS[10:0] from the rising edge of the Hsync signal (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_STH_HW [10:0]	96	R/W	STH Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STH_HW[10:0] (clock cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.39 TCON Horizontal Timing Setting Register STH2 (TCON_TIM_STH2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCON_STH_HS_SEL	—	—	—	TCON_STH_INV	—	TCON_STH_SEL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0..
8	TCON_STH_HS_SEL	0	R/W	STH Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_STH_INV	1	R/W	Polarity Inversion Control of STH Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_STH_SEL[2:0]	2	R/W	Output Signal Select for LCD_TCON2 Pin 2: STH/SP/HS all others: prohibited

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.40 TCON Horizontal Timing Setting Register STB1 (TCON_TIM_STB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_STB_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_STB_HW[10:0]										
Initial value:	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_STB_HS [10:0]	144	R/W	STB Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_STB_HS[10:0] from the rising edge of the Hsync signal (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_STB_HW [10:0]	640	R/W	STB Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_STB_HW[10:0] (clock cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.41 TCON Horizontal Timing Setting Register STB2 (TCON_TIM_STB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCON_STB_HS_SEL	—	—	—	TCON_STB_INV	—	TCON_STB_SEL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	TCON_STB_HS_SEL	0	R/W	STB Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_STB_INV	0	R/W	Polarity Inversion Control of STB Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_STB_SEL[2:0]	7	R/W	Output Signal Select for LCD_TCON3 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE*1

Note 1. Though the signal of the LCD_TCON3 pin can be freely selected from the list, it is recommended to select the DE signal (TCON_STB_SEL[2:0] = 7). This way the DE signal is located in the neighborhood of the other TCON signals at the device package, which might ease the PCB layout.

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.42 TCON Horizontal Timing Setting Register CPV1 (TCON_TIM_CPV1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_CPV_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_CPV_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_CPV_HS [10:0]	0	R/W	CPV Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_CPV_HS[10:0] from the rising edge of the Hsync signal (clock cycles).
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_CPV_HW [10:0]	0	R/W	CPV Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_CPV_HW[10:0] (clock cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.43 TCON Horizontal Timing Setting Register CPV2 (TCON_TIM_CPV2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	TCON_CPV_HS_SEL	—	—	—	TCON_CPV_INV	—	TCON_CPV_SEL[2:0]		
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	TCON_CPV_HS_SEL	0	R/W	CPV Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_CPV_INV	0	R/W	Polarity Inversion Control of CPV Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_CPV_SEL[2:0]	4	R/W	Output Signal Select for LCD_TCON4 Pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.44 TCON Horizontal Timing Setting Register POLA1 (TCON_TIM_POLA1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_POLA_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_POLA_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_POLA_HS [10:0]	0	R/W	POLA Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_POLA_HS[10:0] from the rising edge of the Hsync signal (clock cycles). Note: When 1 × 1, 1 × 2, or 2 × 2 reverse mode is selected, these bits should be set to 1 or greater.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_POLA_HW [10:0]	0	R/W	POLA Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_POLA_HW[10:0] (clock cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.45 TCON Horizontal Timing Setting Register POLA2 (TCON_TIM_POLA2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TCON_POLA_MD[1:0]		—	—	—	TCON_POLA_HS_SEL	—	—	—	TCON_POLA_INV	—	TCON_POLA_SEL[2:0]		
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	1
R/W:	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	TCON_POLA_MD [1:0]	1	R/W	POLA Signal Generation Mode Select 0: Normal mode Generates the signal that changes twice a horizontal period. 1: 1 × 1 reverse mode Generates the signal whose polarity is inverted every horizontal period. 2: 1 × 2 reverse mode Generates the signal whose polarity is inverted in the first horizontal period and is subsequently inverted every two horizontal periods. 3: 2 × 2 reverse mode Generates the signal whose polarity is inverted every two horizontal periods.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	TCON_POLA_HS _SEL	0	R/W	POLA Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_POLA_INV	0	R/W	Polarity Inversion Control of POLA Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_POLA_SEL [2:0]	5	R/W	Output Signal Select for LCD_TCON5 pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.46 TCON Horizontal Timing Setting Register POLB1 (TCON_TIM_POLB1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	TCON_POLB_HS[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	TCON_POLB_HW[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 27	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
26 to 16	TCON_POLB_HS [10:0]	0	R/W	POLB Signal Pulse Start Position (First Changing Timing) Starts pulse output after the time specified by the value of TCON_POLB_HS[10:0] from the rising edge of the Hsync signal (clock cycles). Note: When 1 × 1, 1 × 2, or 2 × 2 reverse mode is selected, these bits should be set to 1 or greater.
15 to 11	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	TCON_POLB_HW [10:0]	0	R/W	POLB Pulse Width (Second Changing Timing) Outputs a pulse of the duration of the value of TCON_POLB_HW[10:0] (clock cycles).

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.47 TCON Horizontal Timing Setting Register POLB2 (TCON_TIM_POLB2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TCON_POLB_MD[1:0]		—	—	—	TCON_POLB_HS_SEL	—	—	—	TCON_POLB_INV	—	TCON_POLB_SEL[2:0]		
Initial value:	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0
R/W:	R	R	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	TCON_POLB_MD [1:0]	1	R/W	POLB Signal Generation Mode Select 0: Normal mode Generates the signal that changes twice a horizontal period. 1: 1 × 1 reverse mode Generates the signal whose polarity is inverted every horizontal period. 2: 1 × 2 reverse mode Generates the signal whose polarity is inverted in the first horizontal period and is subsequently inverted every two horizontal periods. 3: 2 × 2 reverse mode Generates the signal whose polarity is inverted every two horizontal periods.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	TCON_POLB_HS _SEL	0	R/W	POLB Signal Operating Reference Select 0: Hsync signal reference 1: Offset Hsync signal reference
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	TCON_POLB_INV	0	R/W	Polarity Inversion Control of POLB Signal 0: Not inverted 1: Inverted
3	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
2 to 0	TCON_POLB_SEL [2:0]	6	R/W	Output Signal Select for LCD_TCON6 pin 0: STVA/VS 1: STVB/VE 2: STH/SP/HS 3: STB/LP/HE 4: CPV/GCK 5: POLA 6: POLB 7: DE

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.48 TCON Data Enable Polarity Setting Register (TCON_TIM_DE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TCON_DE_INV
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	TCON_DE_INV	0	R/W	Polarity Inversion Control of DE Signal 0: Not inverted 1: Inverted

Note: This register is updated when TCON_VEN in TCON_UPDATE is 1.

38.8.2.49 Register Update Control Register in Output Controller (OUT_UPDATE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OUTCNT_VEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/WC1

Bit	Bit Name	Initial Value	R/W	Description
31 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	OUTCNT_VEN	0	R/WC1	Brightness/Contrast, Dither Process, Output Interface Register Update 0: Registers are not updated. 1: Registers are updated at the rising edge of the Vsync.

38.8.2.50 Output Interface Register (OUT_SET)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	OUT_ENDIAN_ON	—	—	—	OUT_SWAP_ON	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	OUT_FORMAT[1:0]		—	—	OUT_FRQ_SEL[1:0]		—	—	OUT_SEQ_SEL	OUT_DIR_SEL	—	—	OUT_PHASE[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	OUT_ENDIAN_ON	0	R/W	Bit Endian Change On/Off Control 0: Off 1: On
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	OUT_SWAP_ON	0	R/W	B/R Signal Swap On/Off Control 0: Off 1: On
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20 to 16	—	All 1	R	Reserved These bits are always read as 1. The write value should always be 1.
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	OUT_FORMAT[1:0]	0	R/W	Output Format Select 0: RBG888 1: RGB666 2: RGB565 3: Serial RGB (D1M1A and D1M1-V2 only)
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	OUT_FRQ_SEL[1:0]	0	R/W	Clock Frequency Control 0: 100% speed — (parallel RGB) 1: Triple speed — (serial RGB) 2: Quadruple speed — (serial RGB) 3: Setting prohibited
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5	OUT_SEQ_SEL	0	R/W	Output Sequence Select in triple speed mode For the detail, please refer (5) Parallel to Serial Conversion in Section 38.8.1.8, Output Format Conversion 0: R/G/B output sequence is swapped at the next line 1: R/G/B output sequence is fixed at every line

Bit	Bit Name	Initial Value	R/W	Description
4	OUT_DIR_SEL	0	R/W	Scan Direction Select 0: Forward scan 1: Reverse scan
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	OUT_PHASE[1:0]	All 0	R/W	Clock Phase Adjustment During Serial RGB Output • Triple speed mode 0: 0 (clk) 1: 1 (clk) 2: 2 (clk) 3: 0 Setting prohibited • Quadruple speed mode 0: 0 (clk) 1: 1 (clk) 2: 2 (clk) 3: 3 (clk)

Note: This register is updated when OUTCNT_VEN in OUT_UPDATE is 1.

38.8.2.51 Brightness (DC) Correction Register 1 (OUT_BRIGHT1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PBRT_G[9:0]									
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	PBRT_G[9:0]	512	R/W	Brightness (DC) Adjustment of G Signal Unsigned (0 (-512) to 512 (0) to 1023 (+511) [LSB], 512 [LSB] with offset)

Note: This register is updated when OUTCNT_VEN in OUT_UPDATE is 1.

38.8.2.52 Brightness (DC) Correction Register 2 (OUT_BRIGHT2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	PBRT_B[9:0]									
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	PBRT_R[9:0]									
Initial value:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 26	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
25 to 16	PBRT_B[9:0]	512	R/W	Brightness (DC) Adjustment of B Signal Unsigned (0 (-512) to 512 (0) to 1023 (+511) [LSB], 512 [LSB] with offset)
15 to 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9 to 0	PBRT_R[9:0]	512	R/W	Brightness (DC) Adjustment of R Signal Unsigned (0 (-512) to 512 (0) to 1023 (+511) [LSB], 512 [LSB] with offset)

Note: This register is updated when OUTCNT_VEN in OUT_UPDATE is 1.

38.8.2.53 Contrast (Gain) Correction Register (OUT_CONTRAST)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CONT_G[7:0]							
Initial value:	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CONT_B[7:0]								CONT_R[7:0]							
Initial value:	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 24	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
23 to 16	CONT_G[7:0]	128	R/W	Contrast (Gain) Adjustment of G Signal 0/128 to 255/128 (approx.2 times)
15 to 8	CONT_B[7:0]	128	R/W	Contrast (Gain) Adjustment of B Signal 0/128 to 255/128 (approx.2 times)
7 to 0	CONT_R[7:0]	128	R/W	Contrast (Gain) Adjustment of R Signal 0/128 to 255/128 (approx.2 times)

Note: This register is updated when OUTCNT_VEN in OUT_UPDATE is 1.

38.8.2.54 Panel Dither Register (OUT_PDTHA)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	PDTH_SEL[1:0]		—	—	PDTH_FORMAT[1:0]	
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	PDTH_PA[1:0]		—	—	PDTH_PB[1:0]		—	—	PDTH_PC[1:0]		—	—	PDTH_PD[1:0]	
Initial value:	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1
R/W:	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 22	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
21, 20	PDTH_SEL[1:0]	0	R/W	Panel Dither Operation Mode 0: Truncate 1: Round-off 2: 2 × 2 pattern dither 3: Random pattern dither
19, 18	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
17, 16	PDTH_FORMAT [1:0]	0	R/W	Panel Dither Output Format Select 0: RGB888 1: RGB666 2: RGB565 3: Setting prohibited
15, 14	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
13, 12	PDTH_PA[1:0]	3	R/W	Pattern Value (A) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])
11, 10	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
9, 8	PDTH_PB[1:0]	0	R/W	Pattern Value (B) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])
7, 6	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
5, 4	PDTH_PC[1:0]	2	R/W	Pattern Value (C) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])
3, 2	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
1, 0	PDTH_PD[1:0]	1	R/W	Pattern Value (D) of 2 × 2 Pattern Dither Unsigned (0 to 3 [LSB])

Note: This register is updated when OUTCNT_VEN in OUT_UPDATE is 1.

38.8.2.55 Output Phase Control Register (OUT_CLK_PHASE)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	OUTCNT_FRONT_GAM	—	—	—	OUTCNT_LCD_EDGE	—	OUTCNT_STVA_EDGE	OUTCNT_STVB_EDGE	OUTCNT_STH_EDGE	OUTCNT_STB_EDGE	OUTCNT_CPV_EDGE	OUTCNT_POLA_EDGE	OUTCNT_POLB_EDGE
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	OUTCNT_FRONT_GAM	0	R/W	Correction Circuit Sequence Control 0: Brightness → contrast → gamma correction 1: Gamma correction → contrast → brightness
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	OUTCNT_LCD_EDGE	0	R/W	Output Phase Control of LCD_DATA23 to LCD_DATA0 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
7	—	0	R	Reserved This bit is always read as 0. The write value should always be 0.
6	OUTCNT_STVA_EDGE	0	R/W	Output Phase Control of LCD_TCON0 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
5	OUTCNT_STVB_EDGE	0	R/W	Output Phase Control of LCD_TCON1 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
4	OUTCNT_STH_EDGE	0	R/W	Output Phase Control of LCD_TCON2 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
3	OUTCNT_STB_EDGE	0	R/W	Output Phase Control of LCD_TCON3 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
2	OUTCNT_CPV_EDGE	0	R/W	Output Phase Control of LCD_TCON4 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
1	OUTCNT_POLA_EDGE	0	R/W	Output Phase Control of LCD_TCON5 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin
0	OUTCNT_POLB_EDGE	0	R/W	Output Phase Control of LCD_TCON6 Pin 0: Output at the rising edge of LCD_CLK pin 1: Output at the falling edge of LCD_CLK pin

Note: This register is updated when OUTCNT_VEN in OUT_UPDATE is 1.

38.8.3 Usage Methods

38.8.3.1 Gamma Correction Adjustment Method

The characteristics of G, B, R of each panel to be connected should be measured and gamma correction should be made to suit the panel.

Since the gamma correction adjustment depends on to the characteristics of the panel, there are no recommended setting values.

38.8.3.2 Dither Usage Method

Dither is used when pseudo contour is appeared on the display screen.

Table 38.168 Dither Settings

Bit Name	Setting Value
PDTH_FORMAT[1:0]	Selects the format. For RGB888: 0 For RGB666: 1 For RGB565: 2
PDTH_SEL[1:0]	When 2 × 2 pattern dither is to be used: 2
PDTH_PA[1:0]	Normally 3 (initial value)
PDTH_PB[1:0]	Normally 0 (initial value)
PDTH_PC[1:0]	Normally 2 (initial value)
PDTH_PD[1:0]	Normally 1 (initial value)

38.8.3.3 Output Format Adjustment Method

The setting example of the typical output format is shown in **Table 38.169** and **Table 38.170**.

It is necessary to carry out the setting of synchronization system signals of each output format similarly as the setting of output after scaling.

Table 38.169 Setting Example of Synchronizing Signal (1/2)

Register Name	Bit Name	VGA	SVGA	Description
TCON_TIM	TCON_HALF[10:0]	400	528	Sets the half value of 1H time period in clock units.
Vsync signal				
TCON_TIM_STVA1	TCON_STVA_VS[10:0]	0	0	Sets the pulse generation start position from the rising edge of the internal Vsync signal. A 1/2H time period is set as 1.
TCON_TIM_STVA1	TCON_STVA_VW[10:0]	4	8	Sets the changing point from the above pulse generation start position. A 1/2H time period is set as 1.
TCON_TIM_STVA2	TCON_STVA_INV	0	0	Sets the output polarity of the above pulse. For inverted output: 1
TCON_TIM_STVA2	TCON_STVA_SEL[2:0]	0	0	For STVA output selection: 0
Vertical enable signal				
TCON_TIM_STVB1	TCON_STVB_VS[10:0]	68	44	Sets the pulse generation start position from the rising edge of the internal Vsync signal. A 1/2H time period is set as 1.

Table 38.169 Setting Example of Synchronizing Signal (2/2)

Register Name	Bit Name	VGA	SVGA	Description
TCON_TIM_STVB1	TCON_STVB_VW[10:0]	960	1200	Sets the changing point from the above pulse generation start position. A 1/2H time period is set as 1.
TCON_TIM_STVB2	TCON_STVB_INV	0	0	Sets the output polarity of the above pulse. For inverted output: 1
TCON_TIM_STVB2	TCON_STVB_SEL[2:0]	1	1	For STVB output selection: 1

Register Name	Bit Name	VGA	SVGA	Description
Hsync signal				
TCON_TIM_STH1	TCON_STH_HS[10:0]	0	0	Sets the pulse generation start position from the rising edge of the internal Hsync signal.
TCON_TIM_STH1	TCON_STH_HW[10:0]	96	128	Sets the changing point from the above pulse generation start position.
TCON_TIM_STH2	TCON_STH_INV	0	0	Sets the output polarity of the above pulse. For inverted output: 1
TCON_TIM_STH2	TCON_STH_SEL[2:0]	2	2	For STH output selection: 2
Horizontal enable signal				
TCON_TIM_STB1	TCON_STB_HS[10:0]	128	192	Sets the pulse generation start position from the rising edge of the internal Hsync signal.
TCON_TIM_STB1	TCON_STB_HW[10:0]	640	800	Sets the changing point from the above pulse generation start position.
TCON_TIM_STB2	TCON_STB_INV	0	0	Sets the output polarity of the above pulse. For inverted output: 1
TCON_TIM_STB2	TCON_STB_SEL[2:0]	3	3	For STB output selection: 3

Table 38.170 Setting Example of Data System

Register Name	Bit Name	RGB888	RGB ()	Description
OUT_SET	OUT_ENDIAN_ON	0	0	When bit endian is changed: 1
OUT_SET	OUT_SWAP_ON	0	0	When B/R is to be swapped: 1
OUT_SET	OUT_FORMAT[1:0]	0	3	Sets the output format. For RGB888: 0 For RGB666: 1 For RGB565: 2 For serial RGB: 3
OUT_SET	OUT_FRQ_SEL[1:0]	0	1	Sets the output clock. For RGB888, RGB666, RGB565: 0 For triple speed serial RGB output: 1 For quadruple serial RGB output: 2
OUT_SET	OUT_DIR_SEL	0	0	When data arrangement of the serial RGB output is to be reversed: 1

38.9 System Controller

38.9.1 System Controller Functions

38.9.1.1 Overview of Functions

The system controller provides interrupt control, panel clock control, CLUT table read select signal status flag output functions.

38.9.1.2 Interrupt Control

Several interrupt signals are output from the scaler and image synthesizer. The system controller controls whether to output these interrupt signals.

One is written to the corresponding INT_STA* bit when an interrupt signal is to be accepted. After 1 has been written to the bit, however, its value is still read out as 0 until the interrupt signal is accepted. Once the interrupt signal has been accepted, the INT_STA* bit is read out as 1.

The bit for an accepted interrupt signal is cleared by writing 0 to the INT_STA* bit. If further interrupt signals are to be accepted after the INT_STA* bit has been cleared, 1 is again written to the bit.

NOTE

Interrupts settings should be set after clock supply to VDCE in order to avoid unexpected interrupts.

About VDCE clock, refer to Section 38.1.3, Clock supply.

Table 38.171 Interrupt Signals

Request Source Name	Bit Name	Function
S0_VI_VSYNC	INT_STA 0	Vsync signal input to scaler 0
S0_LO_VSYNC	INT_STA 1	Vsync signal output from scaler 0
S0_VSYNCERR	INT_STA 2	Missing Vsync signal for scaler 0
GR3_VLINE	INT_STA 3	Specified line signal for panel output in graphics 3
S0_VFIELD	INT_STA 4	Field end signal for recording function in scaler 0
IV1_VBUFERR	INT_STA 5	Frame buffer write overflow signal for scaler 0
IV3_VBUFERR	INT_STA 6	Frame buffer read underflow signal for graphics 0
IV5_VBUFERR	INT_STA 7	Frame buffer read underflow signal for graphics 2
IV6_VBUFERR	INT_STA 8	Frame buffer read underflow signal for graphics 3
S1_LO_VSYNC	INT_STA 11	Vsync signal output from scaler 1
S1_VSYNCERR	INT_STA 12	Missing Vsync signal for scaler 1
IV4_VBUFERR	INT_STA 15	Frame buffer read underflow signal for graphics 1
OIR_VI_VSYNC	INT_STA 17	Vsync signal input to output image generator
OIR_LO_VSYNC	INT_STA 18	Vsync signal output from output image generator
OIR_VLINE	INT_STA 19	Specified line signal for panel output in output image generator
IV8_VBUFERR	INT_STA 22	Frame buffer read underflow signal for graphics (OIR)

Table 38.172 Interrupt Clear/Hold Settings (1/2)

Register Name	Bit Name	Initial Value	Description
SYSCNT_INT1	INT_STA0	0	S0_VI_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA1	0	S0_LO_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA2	0	S0_VSYNCERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA3	0	GR3_VLINE Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA4	0	S0_VFIELD Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA5	0	IV1_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA6	0	IV3_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT1	INT_STA7	0	IV5_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT2	INT_STA8	0	IV6_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT2	INT_STA11	0	S1_LO_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT2	INT_STA12	0	S1_VSYNCERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT2	INT_STA15	0	IV4_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.

Table 38.172 Interrupt Clear/Hold Settings (2/2)

Register Name	Bit Name	Initial Value	Description
SYSCNT_INT3	INT_STA17	0	OIR_VI_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT3	INT_STA18	0	OIR_LO_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT3	INT_STA19	0	OIR_VLINE Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
SYSCNT_INT3	INT_STA22	0	IV8_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.

Table 38.173 Interrupt Output On/Off Settings (1/2)

Register Name	Bit Name	Initial Value	Description
SYSCNT_INT4	INT_OUT0_ON	0	S0_VI_VSYNC Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT1_ON	0	S0_LO_VSYNC Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT2_ON	0	S0_VSYNCERR Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT3_ON	0	GR3_VLINE Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT4_ON	0	S0_VFIELD Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT5_ON	0	IV1_VBUFERR Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT6_ON	0	IV3_VBUFERR Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT4	INT_OUT7_ON	0	IV5_VBUFERR Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT5	INT_OUT8_ON	0	IV6_VBUFERR Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT5	INT_OUT11_ON	0	S1_LO_VSYNC Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT5	INT_OUT12_ON	0	S1_VSYNCERR Interrupt Output On/Off 0: Off 1: On

Table 38.173 Interrupt Output On/Off Settings (2/2)

Register Name	Bit Name	Initial Value	Description
SYSCNT_INT5	INT_OUT15_ON	0	IV4_VBUFERR Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT6	INT_OUT18_ON	0	OIR_LO_VSYNC Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT6	INT_OUT19_ON	0	OIR_VLINE Interrupt Output On/Off 0: Off 1: On
SYSCNT_INT6	INT_OUT22_ON	0	IV8_VBUFERR Interrupt Output On/Off 0: Off 1: On

38.9.1.3 CLUT Table Read Select Signal Status Flag

The CLUT read select signal status can be read using the flags shown in **Table 38.174**.

Table 38.174 CLUT Table Read Select Signal Status Flags

Register Name	Bit Name	Initial Value	Description
SYSCNT_CLUT	GR0_CLT_SEL_ST	—	Graphics 0 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.
SYSCNT_CLUT	GR1_CLT_SEL_ST	—	Graphics 1 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.
SYSCNT_CLUT	GR2_CLT_SEL_ST	—	Graphics 2 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.
SYSCNT_CLUT	GR3_CLT_SEL_ST	—	Graphics 3 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.

38.9.2 Register Descriptions

The following register sets are allocated in the SH register map area.

- Symbols used in Register Descriptions

  Initial value: Register value after a reset

  —: Undefined value

  R/W: Readable/writable. The written value can be read.

  R/WC0: Readable/writable. Writing 0 initializes the bit. Writing 1 is ignored.

  R/WC1: Readable/writable. Writing 1 initializes the bit. Writing 0 is ignored.

  R: Read-only. The write value should always be 0.

  —/W: Write-only. The read value is undefined.

Table 38.175 Register Configuration of System Controller (CH0)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE0	Interrupt control register 1	SYSCNT_INT1	R/W	H'0000 0000	<VDCE0_base> + 1A80	32
VDCE0	Interrupt control register 2	SYSCNT_INT2	R/W	H'0000 0000	<VDCE0_base> + 1A84	32
VDCE0	Interrupt control register 3	SYSCNT_INT3	R/W	H'0000 0000	<VDCE0_base> + 1A88	32
VDCE0	Interrupt control register 4	SYSCNT_INT4	R/W	H'0000 0000	<VDCE0_base> + 1A8C	32
VDCE0	Interrupt control register 5	SYSCNT_INT5	R/W	H'0000 0000	<VDCE0_base> + 1A90	32
VDCE0	Interrupt control register 6	SYSCNT_INT6	R/W	H'0000 0000	<VDCE0_base> + 1A94	32
VDCE0	CLUT table read select signal status register	SYSCNT_CLUT	R	H'0000	<VDCE0_base> + 1A9A	16

Table 38.176 Register Configuration of System Controller (CH1)

Module Name	Register Name	Symbol	R/W	Initial Value	Address	Access Size
VDCE1	Interrupt control register 1	SYSCNT_INT1	R/W	H'0000 0000	<VDCE1_base> + 1A80	32
VDCE1	Interrupt control register 2	SYSCNT_INT2	R/W	H'0000 0000	<VDCE1_base> + 1A84	32
VDCE1	Interrupt control register 3	SYSCNT_INT3	R/W	H'0000 0000	<VDCE1_base> + 1A88	32
VDCE1	Interrupt control register 4	SYSCNT_INT4	R/W	H'0000 0000	<VDCE1_base> + 1A8C	32
VDCE1	Interrupt control register 5	SYSCNT_INT5	R/W	H'0000 0000	<VDCE1_base> + 1A90	32
VDCE1	Interrupt control register 6	SYSCNT_INT6	R/W	H'0000 0000	<VDCE1_base> + 1A94	32
VDCE1	CLUT table read select signal status register	SYSCNT_CLUT	R	H'0000	<VDCE1_base> + 1A9A	16

NOTE

Register access sizes other than defined in the table above are not supported.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

38.9.2.1 Interrupt Control Register 1 (SYSCNT_INT1)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	INT_STA7	—	—	—	INT_STA6	—	—	—	INT_STA5	—	—	—	INT_STA4
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	INT_STA3	—	—	—	INT_STA2	—	—	—	INT_STA1	—	—	—	INT_STA0
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	INT_STA7	0	R/W	IV5_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	INT_STA6	0	R/W	IV3_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	INT_STA5	0	R/W	IV1_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	INT_STA4	0	R/W	S0_VFIELD Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	INT_STA3	0	R/W	GR3_VLINE Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
8	INT_STA2	0	R/W	S0_VSYNCERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INT_STA1	0	R/W	S0_LO_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INT_STA0	0	R/W	S0_VI_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.

38.9.2.2 Interrupt Control Register 2 (SYSCNT_INT2)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	INT_STA15	—	—	—	—	—	—	—	—	—	—	—	INT_STA12
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	INT_STA11	—	—	—	—	—	—	—	—	—	—	—	INT_STA8
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	INT_STA15	0	R/W	IV4_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
27 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	INT_STA12	0	R/W	S1_VSYNCERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	INT_STA11	0	R/W	S1_LO_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
11 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INT_STA8	0	R/W	IV6_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.

38.9.2.3 Interrupt Control Register 3 (SYSCNT_INT3)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	INT_STA22	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	INT_STA19	—	—	—	INT_STA18	—	—	—	INT_STA17	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	INT_STA22	0	R/W	IV8_VBUFERR Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
23 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	INT_STA19	0	R/W	OIR_VLINE Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	INT_STA18	0	R/W	OIR_LO_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INT_STA17	0	R/W	OIR_VI_VSYNC Interrupt Clear/Hold 0(W): Clears the interrupt status. 1(W): Starts interrupt acceptance. 0(R): No interrupt has occurred. 1(R): An interrupt has occurred.
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

38.9.2.4 Interrupt Control Register 4 (SYSCNT_INT4)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	INT_OUT7_ON	—	—	—	INT_OUT6_ON	—	—	—	INT_OUT5_ON	—	—	—	INT_OUT4_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	INT_OUT3_ON	—	—	—	INT_OUT2_ON	—	—	—	INT_OUT1_ON	—	—	—	INT_OUT0_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	INT_OUT7_ON	0	R/W	IV5_VBUFERR Interrupt Output On/Off 0: Off 1: On
27 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	INT_OUT6_ON	0	R/W	IV3_VBUFERR Interrupt Output On/Off 0: Off 1: On
23 to 21	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
20	INT_OUT5_ON	0	R/W	IV1_VBUFERR Interrupt Output On/Off 0: Off 1: On
19 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	INT_OUT4_ON	0	R/W	S0_VFIELD Interrupt Output On/Off 0: Off 1: On
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	INT_OUT3_ON	0	R/W	GR3_VLINE Interrupt Output On/Off 0: Off 1: On
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	INT_OUT2_ON	0	R/W	S0_VSYNCERR Interrupt Output On/Off 0: Off 1: On
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

Bit	Bit Name	Initial Value	R/W	Description
4	INT_OUT1_ON	0	R/W	S0_LO_VSYNC Interrupt Output On/Off 0: Off 1: On
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INT_OUT0_ON	0	R/W	S0_VI_VSYNC Interrupt Output On/Off 0: Off 1: On

38.9.2.5 Interrupt Control Register 5 (SYSCNT_INT5)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	INT_OUT15_ON	—	—	—	—	—	—	—	—	—	—	—	INT_OUT12_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	INT_OUT11_ON	—	—	—	—	—	—	—	—	—	—	—	INT_OUT8_ON
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	Bit Name	Initial Value	R/W	Description
31 to 29	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
28	INT_OUT15_ON	0	R/W	IV4_VBUFERR Interrupt Output On/Off 0: Off 1: On
27 to 17	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
16	INT_OUT12_ON	0	R/W	S1_VSYNCERR Interrupt Output On/Off 0: Off 1: On
15 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	INT_OUT11_ON	0	R/W	S1_LO_VSYNC Interrupt Output On/Off 0: Off 1: On
11 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	INT_OUT8_ON	0	R/W	IV6_VBUFERR Interrupt Output On/Off 0: Off 1: On

38.9.2.6 Interrupt Control Register 6 (SYSCNT_INT6)

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	INT_OUT22_ON	—	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	INT_OUT19_ON	—	—	—	INT_OUT18_ON	—	—	—	INT_OUT17_ON	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
31 to 25	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
24	INT_OUT22_ON	0	R/W	IV8_VBUFERR Interrupt Output On/Off 0: Off 1: On
23 to 13	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
12	INT_OUT19_ON	0	R/W	OIR_VLINE Interrupt Output On/Off 0: Off 1: On
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	INT_OUT18_ON	0	R/W	OIR_LO_VSYNC Interrupt Output On/Off 0: Off 1: On
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	INT_OUT17_ON	0	R/W	OIR_VI_VSYNC Interrupt Output On/Off 0: Off 1: On
3 to 0	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.

38.9.2.7 CLUT Table Read Select Signal Status Register (SYSCNT_CLUT)

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	GR3_CLT_SEL_ST	—	—	—	GR2_CLT_SEL_ST	—	—	—	GR1_CLT_SEL_ST	—	—	—	GR0_CLT_SEL_ST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Bit Name	Initial Value	R/W	Description
15 to 13	—	All 0	R	Reserved This bit is always read as 0. The write value should always be 0.
12	GR3_CLT_SEL_ST	0	R	Graphics 3 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.
11 to 9	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
8	GR2_CLT_SEL_ST	0	R	Graphics 2 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.
7 to 5	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
4	GR1_CLT_SEL_ST	0	R	Graphics 1 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.
3 to 1	—	All 0	R	Reserved These bits are always read as 0. The write value should always be 0.
0	GR0_CLT_SEL_ST	0	R	Graphics 0 CLUT Table Read Select Signal Status Flag 0: CLUT table 0 is read out. 1: CLUT table 1 is read out.

Section 39 Video Output Warping Engine (VOWE)

This section contains a description of the RH850/D1L/D1M specific properties of the Video Output Warping Engine, such as the number of units, register base addresses, etc.

The functional description of the Video Output Warping Engine is available under non-disclosure agreement.

For details, contact your local sales representatives.

39.1 Features of RH850/D1L/D1M VOWE

39.1.1 Number of Units

This microcontroller has the following number of units of the VOWE.

Table 39.1 Units

Video Data Controller E (VDCE)	D1L1	D1L2(H)	D1M1(H) D1M1-V2 D1M1A	D1M2(H)
Units	–	–	1	1
Names	–	–	VOWE	VOWE

39.1.2 Register Base Address

VOWE base addresses are listed in the following table.

VOWE register addresses are given as offsets from the base addresses in general.

Table 39.2 Register Base Address

Base Address Name	Base Address
<VOWE_base>	F200 0000_H

39.1.3 Clock Supply

The VOWE clock supply is shown in the following table.

Table 39.3 Clock Supply

Unit Name	Unit Clock Name	Supply Clock Name
VOWE	VOWE operation clock	C_ISO_CPUCLK
	Cross-connect bus clock	C_ISO_XCCLK
	PCLK	C_ISO_XCCLK/2
	Video Out Pixel Clock	Video channels clock generator C_ISO_VDCE0CLK

39.1.4 Interrupt Request

VOWE interrupt requests are listed in the following table.

Table 39.4 Interrupt Requests

VOWE signals	Function	Connected to
VOWEINT	Common interrupt	Interrupt Controller INTVOWE0

39.1.5 Reset Sources

VOWE reset sources are listed in the following table. VOWE is initialized by these reset sources.

Table 39.5 Reset Sources

Unit Name	Reset Source
VOWE	• Reset Controller SYSRES • Reset upon wake-up from DEEPSTOP mode

39.1.6 Bus master ID

The Video Output Warping Engine bus master interfaces are connected to the cross-connect system XC2, which forms an own master ID domain.

Since Video Output Warping Engine accesses the XC0 and XC1 cross-connects through the Sprite Engine master interfaces, the master ID of the respective Sprite Engine master interface appears on the XC0/XC1 cross-connects.

The following table shows the Video Output Warping Engine master interfaces IDs, their connection to an XC2 layer and the Sprite Engine slave interface.

Table 39.6 Video Output Warping Engine master ID

		XC2 cross-connect		Sprite Engine	
Master I/F	Access	Master ID	XC2 layer	XC2 slave	XC0/XC1 master ID
Video Output Warping Engine	W	6	XC2_2	Sprite Unit 1	MSTID10
	R				

For detailed information about the cross-connection systems and the usage of the master IDs refer to Section 14.2, Cross-connect systems.

Section 40 Video Output Checker A (VOCA)

This section contains a generic description of the Video Output Checker.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

40.1 Overview of the RH850/D1L/D1M Video Output Checkers

40.1.1 Units

This microcontroller has the following number of units of the Video Output Checker.

Table 40.1 Units

Video Output Checker (VOCA)	D1L1	D1L2(H)	D1M1(H) D1M1-V2 D1M1A	D1M2(H)
Units	0	0	1	1
Names	–	–	VOCA0	VOCA0

Unit index n

Throughout this section, the individual units of the Video Output Checkers are identified by the index “n” (n = 0), for example VOCAnEN for the VOCAn enable register.

Monitor index m

Throughout this section, the individual monitors of the Video Output Checkers are identified by the index “m” (m = 0 to 15), for example VOCAnMmCFG0 for the VOCAn monitor channel m start point register.

Video channel index j

Throughout this section, the individual video channels are identified by the index “j”.

Table 40.2 Video channels index

Video channel index	D1M1(H), D1M1-V2	D1M1A, D1M2(H)
Video channels	1	2
Video channel index	j = 0	j = 0, 1

40.1.2 Register addresses

All Video Output Checker register addresses are given as address offsets from the individual base addresses <VOCAn_base>.
The <VOCAn_base> addresses of each VOCAn are listed in the following table:

Table 40.3 Register base addresses <VOCAn_base>

VOCAn unit	<VOCAn_base> address
VOCA0	FFFD 8000_H

40.1.3 Clock supply

All Video Output Checkers provide two clock inputs.

Table 40.4 Clock supply

VOCAn unit	VOCAn clock	Connected to
VOCA0	PCLK	Clock Controller C_ISO_PCLK
	Activity Monitor operation clock	Clock Controller f_{RL}

40.1.4 Interrupts

The Video Output Checkers can generate the following interrupt requests:

Table 40.5 VOCAn interrupt requests

VOCA0 signals	Function	Connected to
VOC_INT	Video Output Monitor error interrupt	Both interrupts are logically OR combined and input to the Error Control Module INTVOCAERR
ACTMON_INT	Activity Monitor error interrupt	

40.1.5 Reset sources

The Video Output Checkers and their registers are initialized by the following reset signal:

Table 40.6 Reset sources

Reset target	Reset signal
VOCAn module	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

40.1.6 Internal signals

The internal signal connections of the Video Output Checkers are listed in the following table.

Table 40.7 VOCAn internal signal connections

VOCA0 signals	Function	Connected to
Video Output Checker connection to Video Data Controller VDCE0:		
VCLK0	Pixel clock	Video channels clock generator C_ISO_VDCE0CLK
VSYNC0	Vertical synchronization	Port VDCE0_VO_TCON0
HSYNC0	Horizontal synchronization	Port VDCE0_VO_TCON2
RGB444	Most significant 4 bit of each RGB color component	R[7:4], G[7:4], B[7:4]
Video Output Checker connection to Video Data Controller VDCE1 (for D1M2(H) and D1M1A only):		
VCLK1	Pixel clock	Video channels clock generator C_ISO_VDCE1CLK
VSYNC1	Vertical synchronization	Port VDCE1_VO_TCON0
HSYNC1	Horizontal synchronization	Port VDCE1_VO_TCON2
RGB444	Most significant 4 bit of each RGB color component	R[7:4], G[7:4], B[7:4]

40.2 Functions Overview

The Video Output Checker allows to check whether the display content is correctly output by the Video Output.

Main purpose of the Video Output Checker is to secure perceivable appearance of warning symbols on the display. Such warning symbols are normally clearly distinctive in terms of shape and color against other display content.

Features summary

- Total of up to 16 monitored areas
- Up to 32,768 pixels in total over all monitored areas
- Four programmable reference color ranges for each monitored area
- Programmable discriminator threshold for fail/pass decision
- Interrupt generation upon
  - symbol detection mismatch
  - Activity Monitor
- Support both video channels of D1M2H and D1M1A devices, that means in particular:
  - Total of up to 16 monitored areas on both video output screens supported
  - Different number of monitored areas can be assigned to the video outputs
  - Activity Monitor supports two separate time windows for the video output channels.

The interrupts are input to the Error Control Module and thus can generate interrupts or a reset.

Restriction

VOCA doesn't support the use of the Serial RGB I/F. For the detail of Serial RGB, please refer (5) Parallel to Serial Conversion in Section 38.8.1.8, Output Format Conversion.

Block Diagram

The block diagram shows the main components of the VOCA.

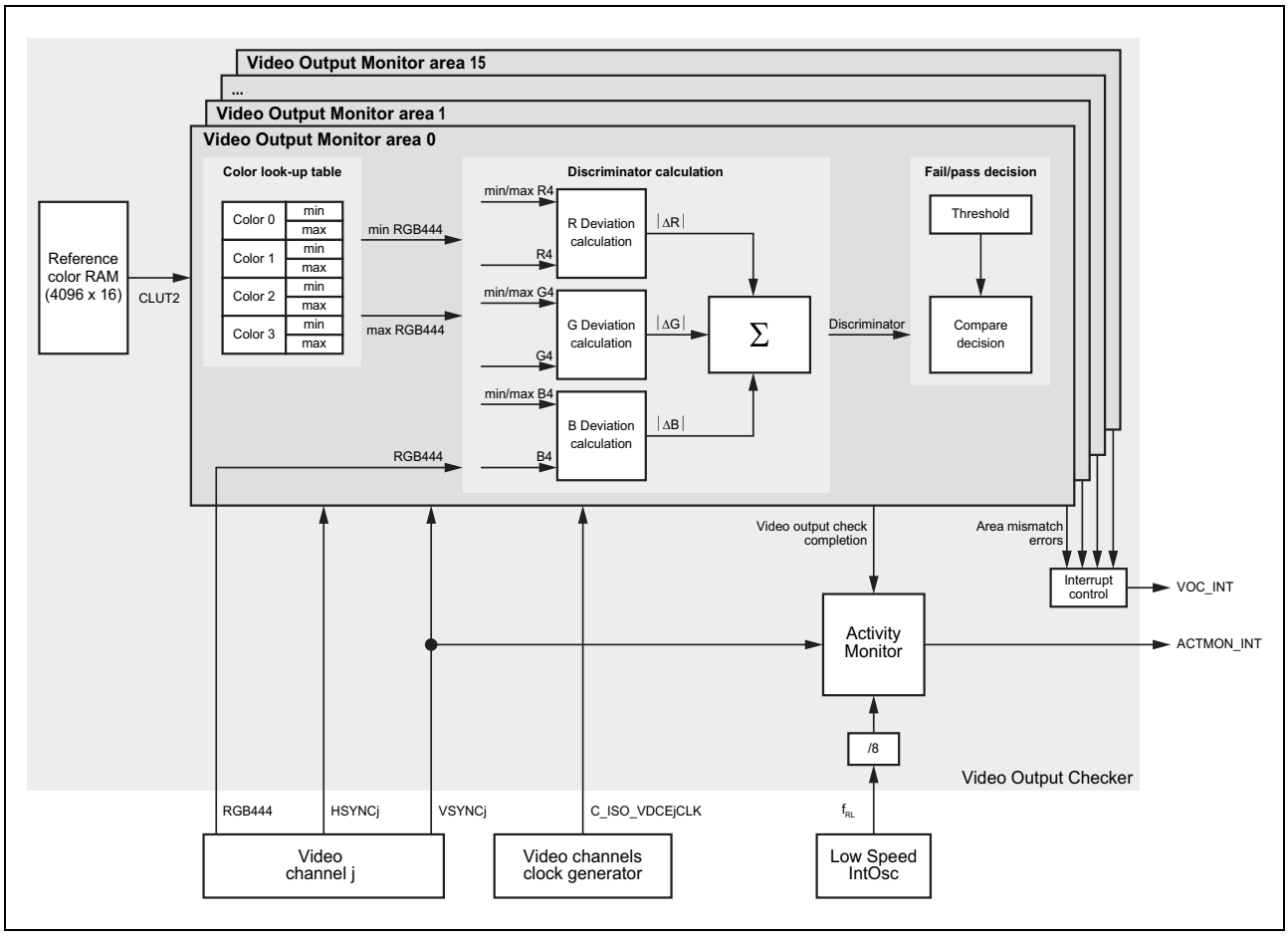


Figure 40.1 Block diagram of the Video Output Checker A

40.3 Functional Description

40.3.1 Video Output Monitoring Basic Procedure

The video output pixels of a rectangle area, that is supposed to show, for unit, a warning symbol, are tested against a reference image.
For that purpose a discriminator is calculated, which is a measure of the deviation between the output rectangle and its reference. If the discriminator exceeds a certain threshold the output is judged as incorrect.

In order to cope with unpredictable symbol background pattern (for unit because of dithering during of video output format expansion, life video background, multi-layer blending effects, gamma correction, etc.) the discriminator calculation

- is performed on reduced 12-bit RGB444 color format instead of 16-/18-/24-bit color
- compares against a reference color range (minimum/maximum reference color component) rather than on a particular 12-bit color reference value
- forms an integral value over all pixels of the entire rectangular area.

This way the calculation algorithm yields a robust correlation measure between the reference and video output content.

Discriminator calculation

The discriminator value D is a sum of deviations ΔR, ΔG, ΔB of the video output pixel color components from the allowed minimum/maximum reference color component range.

$$D = \Sigma (\Delta R + \Delta G + \Delta B)$$

In the following C stands for the color components R, G and B.

The video data is compared pixel- and color-component-wise with the reference range:

- A video output pixel is judged as correct if its color component values C_{VO} are in the minimum/maximum range (C_{REFmin}/C_{REFmax}) of the reference color component values, i.e.
  - if $C_{REFmin} \leq C_{VO} \leq C_{REFmax}$: $\Delta C = 0$
- If a video output color component value C_{VO} is outside the reference range, its contribution ΔC to the discriminator D is:
  - – if $C_{VO} < C_{REFmin}$: $\Delta C = C_{REFmin} - C_{VO}$
  - – if $C_{VO} > C_{REFmax}$: $\Delta C = C_{VO} - C_{REFmax}$

The following figure gives an example how a discriminator is calculated:

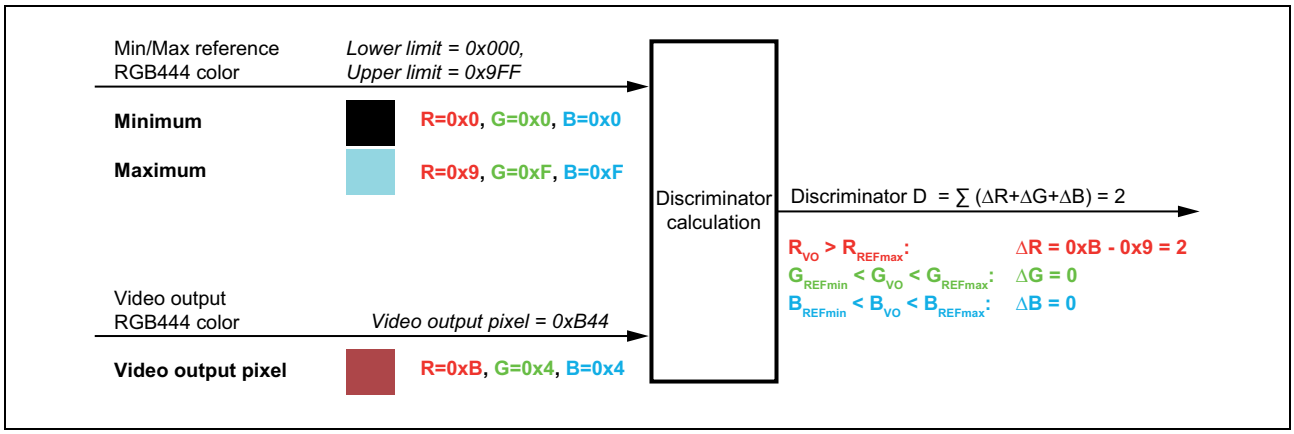


Figure 40.2 Deviation calculation example

Error judgement

The discriminator value D of the monitored area is compared against an acceptance threshold THS:

- if D ≥ THS: content of output rectangle area is erroneous
- if D < THS: content of output rectangle area is not erroneous

40.3.2 Video Channels selection

Before any Video Output Monitor or Activity Monitor is used, the availability of a certain video channel j must be selected:

- VOCAnACTj.VOCAnVOUTj = 0: Video channel j is not available
- VOCAnACTj.VOCAnVOUTj = 1: Video channel j is available

Refer to "Overview of the RH850/D1L/D1M Video Output Checkers" in the first subsection to check which video channels are available in a particular device.

40.3.3 Video Output Monitor implementation

The Video Output Monitor can supervise up to 16 (m = 0 to 15) areas of the video channel's screen.

Each Monitor can be enabled and disabled separately:

- VOCAnEN.VOCAnENm = 0: Monitor m is disabled
- VOCAnEN.VOCAnENm = 1: Monitor m is enabled

Each Monitor can be assigned to monitor an area of the video channel j (j = 0, 1). The assignment is done via the Video Output Monitor channel assignment register:

- VOCAnCH.VOCAnCHm = 0: Monitor m is assigned to video channel 0
- VOCAnCH.VOCAnCHm = 1: Monitor m is assigned to video channel 1

40.3.3.1 Video Monitor area definition

The area m to be monitored is defined relative to the HSYNC and VSYNC signals of the video output channel j:

- VOCAnOFFSj.VOCAnHOFFSj/.VOCAnVOFFSj:
  start of the active video output screen in horizontal/vertical direction

  The back porch offset values specify the number of pixels between the falling edge of the HSYNC or VSYNC pulse and the beginning of the active screen output. Thus the offset values to specify depend on the polarity of the HSYNC and VSYNC signals:

  – If the SYNC signal is active high (refer to **Figure 40.3**):
    offset value = back porch
    = time between the end of the SYNC pulse and the active screen output
  – If the SYNC signal is active low (refer to **Figure 40.4**):
    offset value = SYNC pulse width + back porch
    = time between the beginning of SYNC pulse and the active screen output
- VOCAnMmCFG0.VOCAnMmHSTRT/.VOCAnMmVSTRT:
  horizontal/vertical coordination of the first pixel of the monitor area m
- VOCAnMmCFG1.VOCAnMmHSIZE/.VOCAnMmVSIZE:
  horizontal/vertical size of the monitor area m (max. horizontal/vertical size is 128)

All values in horizontal direction (VOCAnHOFFSj, VOCAnMmHSTRT, VOCAnMmHSIZE) are defined in pixel units.
All values in vertical direction (VOCAnVOFFSj, VOCAnMmVSTRT, VOCAnMmVSIZE) are defined in line units.

CAUTION

The back and front porch of the video signal must have at least a width of one pixel.

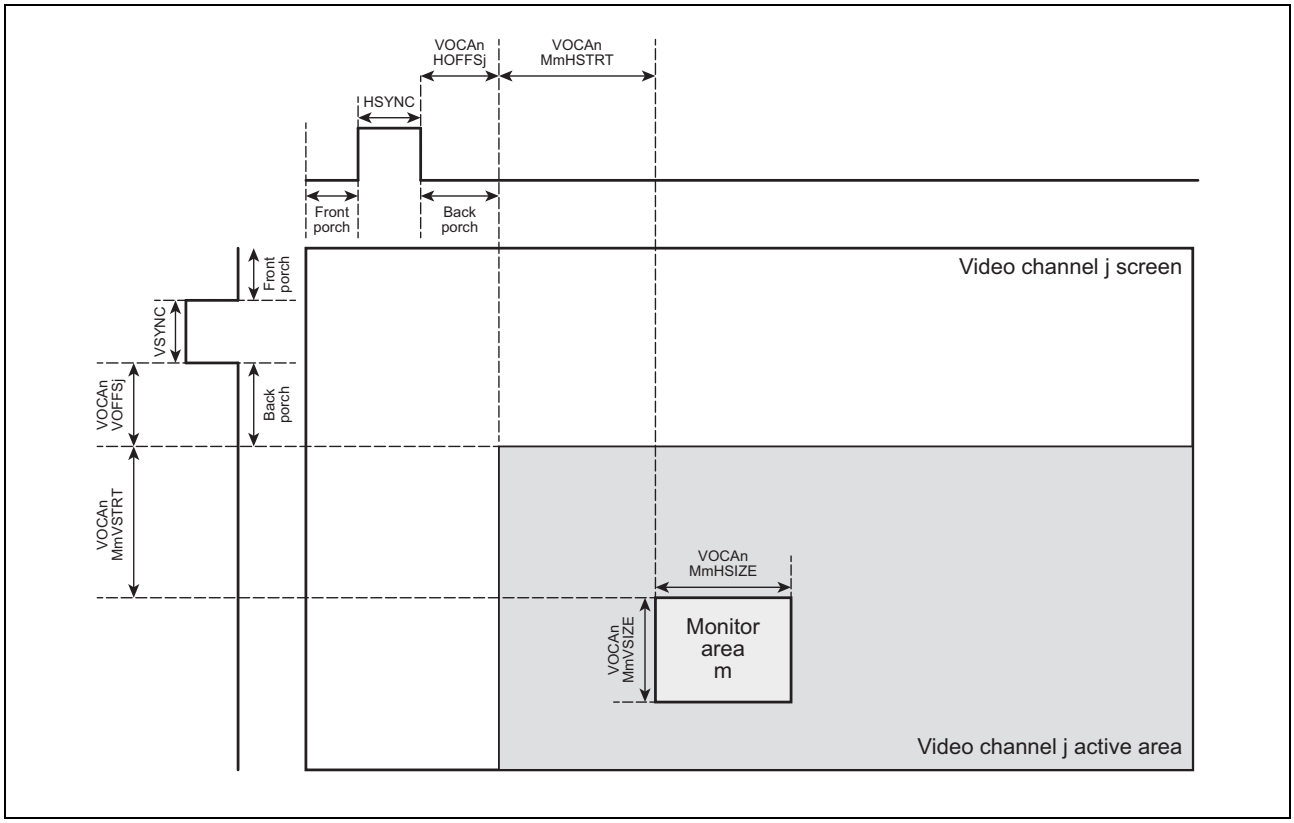


Figure 40.3 Monitor area definition with active high HSYNC and VSYNC

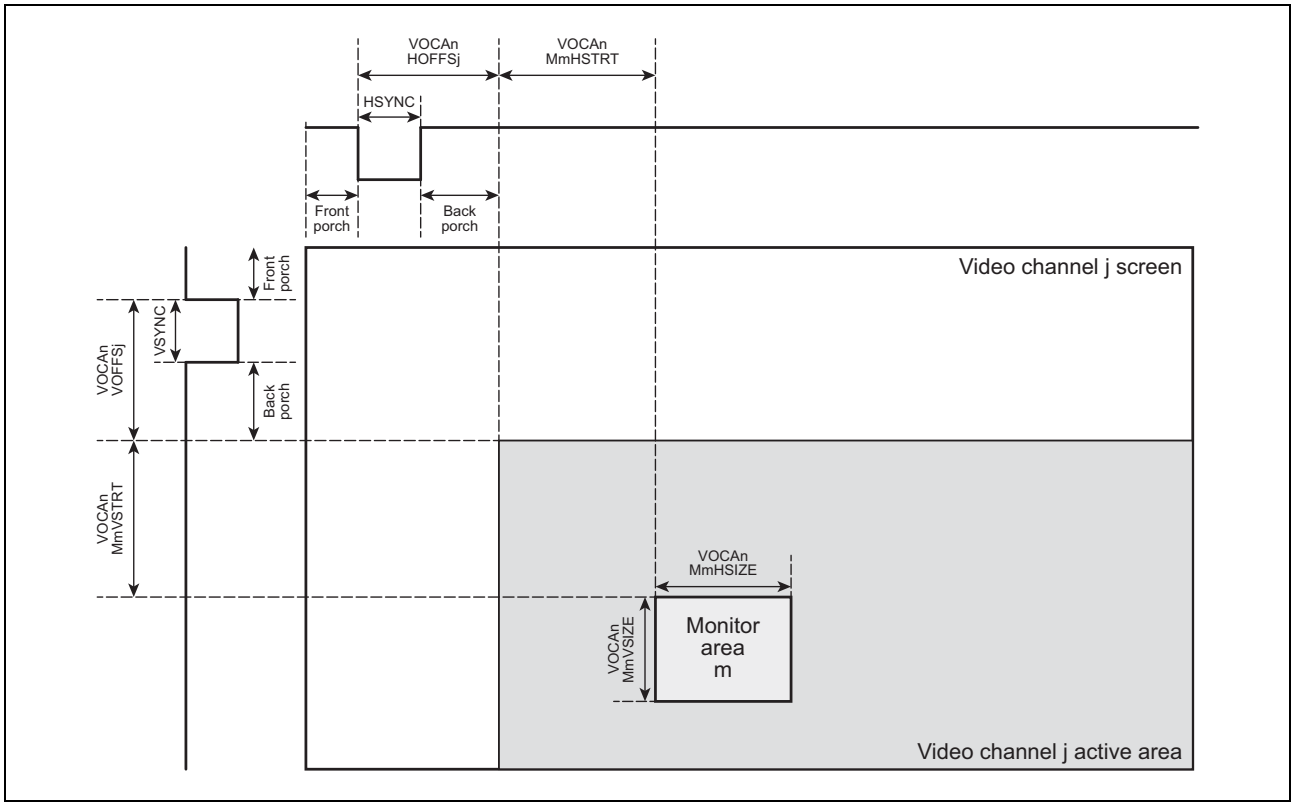


Figure 40.4 Monitor area definition with active low HSYNC and VSYNC

40.3.3.2 Reference color

The reference colors for the discriminator calculation are stored as 2-bit CLUT2 indices, that select one out of four 12-bpp RGB444 colors from a color look-up table.

Reference color definition

Each RGB444 reference color describes an acceptance range by an upper and lower limit for each of the RGB components.
The limit values for each of the four reference colors for each monitor area m are defined by the registers VOCAnMmCFG4 to VOCAnMmCFG7 with following content:

- VOCAnMmRUP[3:0]/RLO[3:0]: upper/lower limit of red component
- VOCAnMmGUP[3:0]/GLO[3:0]: upper/lower limit of green component
- VOCAnMmBUP[3:0]/BLO[3:0]: upper/lower limit of blue component

Reference color selection

The 2-bpp CLUT2 indices are stored in the reference color RAM. The RAM is organized as 4096 x 16 bit.

The address of the reference color CLUT2 index of the first pixel of the monitor area m is defined in the Video Output Monitor m reference color RAM address register VOCAnMmCFG2m.VOCAnMmADDR[11:0] in the range of 0 to 4095, i.e. 000_H to FFF_H.

The 2-bpp CLUT2 indices are stored in the color RAM via the registers VOCAnEXPD0000 to VOCAnEXPD4095.

The addresses of the monitor areas reference color indices can be freely assigned in the RAM. The table below gives an examples for three monitored areas (m = 0, 2, 10) of different size.

NOTES

1. The total number of monitored pixel is limited to 32,768 pixels.
2. The RAM areas, assigned to different monitor areas, must not overlap.

Table 40.8 Reference color RAM example

Monitor area	VOCAnMmCFG2	RAM address	Bit*1 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
not used		000_H																
		001_H																
		...																
m = 0 32 x 32 pixel	VOCAnM0ADDR	010_H	I0		I1		I2		I3		I4		I5		I6		I7	
	VOCAnM0ADDR + 1	011_H	I8		I9		I10		I11		I12		I13		I14		I15	
	...	...	...		...		...		...		...		...		...		...	
	VOCAnM0ADDR + 127	$08F_H$	I1016		I1017		I1018		I1019		I1020		I1021		I1022		I1023	
m = 10 60 x 35 pixel	VOCAnM10ADDR	090_H	I0		I1		I2		I3		I4		I5		I6		I7	
	VOCAnM10ADDR + 1	091_H	I8		I9		I10		I11		I12		I13		I14		I15	
	...	...	...		...		...		...		...		...		...		...	
	VOCAnM10ADDR + 262	196_H	I2096		I2097		I2098		I2099		–		–		–		–	
not used		...																
m = 2 128 x 128 pixel	VOCAnM2ADDR	800_H	I0		I1		I2		I3		I4		I5		I6		I7	
	VOCAnM2ADDR + 1	801_H	I8		I9		I10		I11		I12		I13		I14		I15	
	...	...	...		...		...		...		...		...		...		...	
	VOCAnM2ADDR + 2047	FFF_H	I16376		I16377		I16378		I16379		I16380		I16381		I16382		I16383	

Note 1. “I” stands for a 2-bpp CLUT2 color index.

40.3.3.3 Video Output Monitor error notification

An error during check of the Video Output Monitor area m can be detected for two reasons:

Monitor area m content mismatch error

Such error is detected, if the content check (with respect to discriminator vs. acceptance threshold) fails.
Upon detection of such error

- the monitor m area error bit VOCAnSTR.VOCAnEm is set
- the Video Output Monitor interrupt VOC_INT is asserted.

Monitor area m parameter error

Such error is detected, if the definition of a monitored area m (area m start point in VOCAnMmCFG0 and size in VOCAnMmCFG1) does not fit to the active screen size of the related video channel.
Upon detection of such error

- the parameter error VOCAnSTR.VOCAnE18 flag is set
- the monitor m area error bit VOCAnSTR.VOCAnEm is set
- the Video Output Monitor interrupt VOC_INT is asserted.

NOTES

1. The error flags in the status register VOCAnSTR remain set until they are cleared via the control register VOCAnCTL. Refer to “Video Output Monitor error flag clearing procedure” below for details.

2. The VOC_INT remains asserted as long as any of the Video Output Monitor error flags VOCAnE18 and VOCAnEm are set.

Video Output Monitor error flag clearing procedure

In order to clear a Video Output Monitor error flag, that has been set due to a content mismatch or parameter error detection, proceed as follows:

1. Evaluate the error source via the error flags in the status register VOCAnSTR.
2. Disable INTVOCAERR in the Error Control Module (by setting its error input mask in the Error Control Module).
3. Disable all Video Output Monitors by VOCAnEN.VOCAnENm = 0.
4. Wait until Video Output Monitors are inactive, i.e. until VOCAnSTR.VOCAnVOCSTR = 0.
5. Clear all Video Output Monitor error flags by
   - VOCAnCTL.VOCAnCLm = 1 for content mismatch errors
   - VOCAnCTL.VOCAnCL18 = 1 and VOCAnCTL.VOCAnCLm = 1 for parameter error
6. Release all Video Output Monitor error flags clear by
   - VOCAnCTL.VOCAnCLm = 0 for content mismatch errors
   - VOCAnCTL.VOCAnCL18 = 0 and VOCAnCTL.VOCAnCLm = 0 for parameter error
7. Verify that all Video Output Monitor error flags are reset, i.e. that
   - VOCAnSTR.VOCAnEm = 0: no content mismatch error
   - VOCAnSTR.VOCAnE18 = 0 and VOCAnSTR.VOCAnEm = 0: no parameter error

   If any of the above error flags are not cleared, return to **5.**
8. Enable INTVOCAERR in the Error Control Module (by resetting its error input mask in Error Control Module).
9. Enable required Video Output Monitor m by VOCAnEN.VOCAnENm = 1.

Interrupt masking

Assertion of an error interrupt VOC_INT can be disabled:

- VOCAnCTL.VOCAnMKINT = 0:
  VOC_INT is generated in case of a Video Output Monitor error detection
- VOCAnCTL.VOCAnMKINT = 1:
  no VOC_INT is generated in case of a Video Output Monitor error detection

Note that VOCAnCTL.VOCAnMKINT affects also the Activity Monitor interrupt ACTMON_INT.

40.3.3.4 Monitoring sequence

In each video output frame only a single Video Output Monitor area m is examined.

Priority of monitor areas

If several monitor areas are enabled the monitoring order follows a fixed priority:

priority of m = 0 > m = 1 > ... > m = 15

Detection timing

The time for a content mismatch detection depends on the total number of monitored areas.

The maximum detection time of an erroneous area is calculated by

(number of monitored areas x 2) / framerate

NOTE

The maximum detection time of an erroneous frame assumes that the VOCA needs to switch between the video channels after each Video Output Monitor frame and that the switching to the other video channel takes one frame for synchronization.

It is recommended to setup the Video Output Monitors in a way, that video channel switching is minimized. i.e. set up Video Output Monitors related to the same video channel consecutively. (example: M0 to M7: video channel 0, M8 to M15: video channel 1)

This will shorten the detection time to

(number of monitoring area + 2) / frame rate

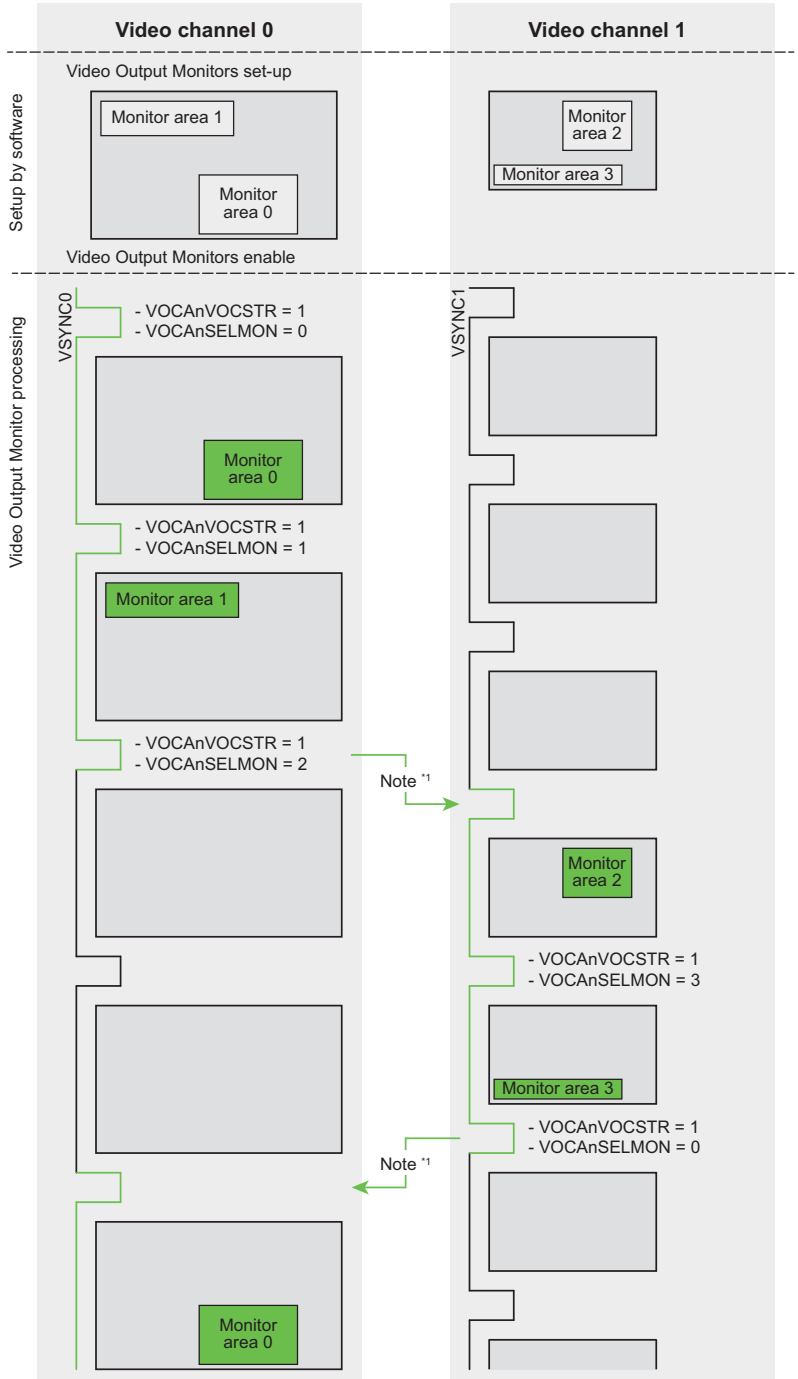
Monitor process status

The current Video Output Monitor process status is reflected in the status register VOCAnSTR:

- VOCAnVOCSTR:
  Video Output Monitor is active in the current video channel frame.
- VOCAnSELMON[3:0]:
  - If the vertical synchronizing signal is active low, VOCAnSELMON[3:0] indicate the number of the current video channel frame, that is monitored.
  - If the vertical synchronizing signal is active high, VOCAnSELMON[3:0] indicate the number of the previous video channel frame.

These status bits are valid for the current video channel frame.

The following diagram shows an example for sequence, with monitor areas for both video channels.



Note 1. During switching between the video channels the Video Output Monitor status indicator VOCAnVOCSTR becomes 0 meanwhile.

Figure 40.5 Video Output Monitor processing sequence example

40.3.4 Activity Monitor

The Activity Monitor controls whether

- the Video Output Monitor has completed the check of a monitor area before the next VSYNC signal occurs and
- the next VSYNC signal is asserted within a certain time window.

If any of the above two conditions is not fulfilled the Activity Monitor notifies an error.

NOTE

Since the Activity Monitor requires active Video Output Monitors on the video channel to check activity, at least one Video Output Monitor must be enabled for the respective video channel.

That means at least one Video Output Monitor with

VOCAnEN.VOCAnENm = 1 and VOCAnCH.VOCAnCHm = 0 (assigned to video channel 0)

VOCAnEN.VOCAnENm = 1 and VOCAnCH.VOCAnCHm = 1 (assigned to video channel 1)

must be set up, if the activity of both video channels shall be monitored.

If only the Activity Monitor is to be used, but no Video Output Monitor on a certain video channel, enable a Video Output Monitor for that video channel and set it up in such a way that it does not generate any Video Output Monitor errors.

This can be achieved by selecting the range of all reference colors to maximum, i.e.

VOCAnMmCFG4m = VOCAnMmCFG5m =
VOCAnMmCFG6m = VOCAnMmCFG7m = 0FFF 0000_H

If the Activity Monitor for only channel 1 is to be enabled, channel 0 also is monitored at first because channel 0 is selected by default.

Activity Monitors activation

The Activity Monitors for both video channels can be enabled and disabled all together:

- VOCAnEN.VOCAnEN16 = 0: Activity Monitors inactive
- VOCAnEN.VOCAnEN16 = 1: Activity Monitors active

40.3.4.1 Activity Monitors set-up

The time window for a valid VSYNC signal is determined by a counter, that is operated with the clock of the Low Speed IntOsc $f_{RL}/8$, i.e. with nominal 30 kHz. The time window's duration and location on the time axis is set up via the timing register VOCAnTIMEj for each of both video channels j = 0, 1:

- VOCAnTIMEj.VOCAnMINj: minimum detection time of Activity Monitors for video channel j
- VOCAnTIMEj.VOCAnMAXj: maximum detection time of Activity Monitors for video channel j

Both values are defined as a multiple of 8 x $T_{RL} = 8/f_{RL}$ = 0.033 ms up to

- VOCAnTIMEj.VOCAnMINj: 136.467 ms
- VOCAnTIMEj.VOCAnMAXj: 136.5 ms

NOTES

1. Subject to the Activity Monitor check in a certain video channel frame is always the current monitor area for that frame, i.e. the area m, indicated by VOCAnSTR.VOCAnSELMON[3:0].
2. The time window definitions VOCAnTIMEj.VOCAnMINj/MAXj are automatically selected for the video channel j, that is assigned to the current monitor area via the Video Output Monitor channel assignment VOCAnCHm.

40.3.4.2 Activity Monitors error notification

The Activity Monitor notifies an error

- at the end of the current video channel frame, if the active Video Output Monitor does not complete area m processing or
- at the end of the time window, if the VSYNC signal does not occur within the allowed time window.

Upon detection of such error

- the Activity Monitor error bit
  - VOCAnSTR.VOCAnE16 for video channel 0 error
  - VOCAnSTR.VOCAnE17 for video channel 1 error

  is set
- the Activity Monitor error interrupt ACTMON_INT asserted

NOTES

1. The error flags in the status register VOCAnSTR remain set until they are cleared via the control register VOCAnCTL. Refer to “Activity Monitors error flag clearing procedure” below for details.
2. The ACTMON_INT remains asserted as long as any of the Activity Monitor error flags VOCAnE16 and VOCAnE17 are set.
3. When the Activity Monitor error interrupt ACTMON_INT occur with Activity Monitor channel x error, the software must handle the cause of this error first. I.e. resolve the cause for the missing or wrong pixel clock condition of the video output channel x.
   In case that the pixel clock of video output channel x is missing at all, the continuation of VOCA operation else might cause the error for the other Activity Monitor channel y not to be generated.

Activity Monitors error flag clearing procedure

In order to clear an Activity Monitor error flag proceed as follows:

1. Evaluate the error source via the error flags in the status register VOCAnSTR.
2. Disable INTVOCAERR in the Error Control Module (by setting its error input mask in the Error Control Module).
3. Disable the Activity Monitors by VOCAnEN.VOCAnEN16 = 0.

4. Wait until Activity Monitors are inactive, i.e. until VOCAnSTR.VOCAnACTSTR = 0.
5. Clear all Activity Monitor error flags by VOCAnCTL.VOCAnCL[17:16] = 1.
6. Release all Activity Monitor error flags clear by VOCAnCTL.VOCAnCL[17:16] = 0.
7. Verify that all Activity Monitor error flags are reset, i.e. that VOCAnSTR.VOCAnE[17:16] = 0. If any of the above error flags are not cleared, return to **5.**
8. Enable INTVOCAERR in the Error Control Module (by resetting its error input mask in Error Control Module).
9. Enable the Activity Monitors by VOCAnEN.VOCAnEN16 = 1.

Interrupt masking

Assertion of an error interrupt ACTMON_INT can be disabled:

- VOCAnCTL.VOCAnMKINT = 0:
  ACTMON_INT is generated in case of an Activity Monitor error detection
- VOCAnCTL.VOCAnMKINT = 1:
  no ACTMON_INT is generated in case of an Activity Monitor error detection

Note that VOCAnCTL.VOCAnMKINT affects also the Video Output Monitor interrupt VOC_INT.

Figure 40.6 shows an example with correct detection of the Video Output Monitor completion and VSYNC signal. Thus no error is reported.

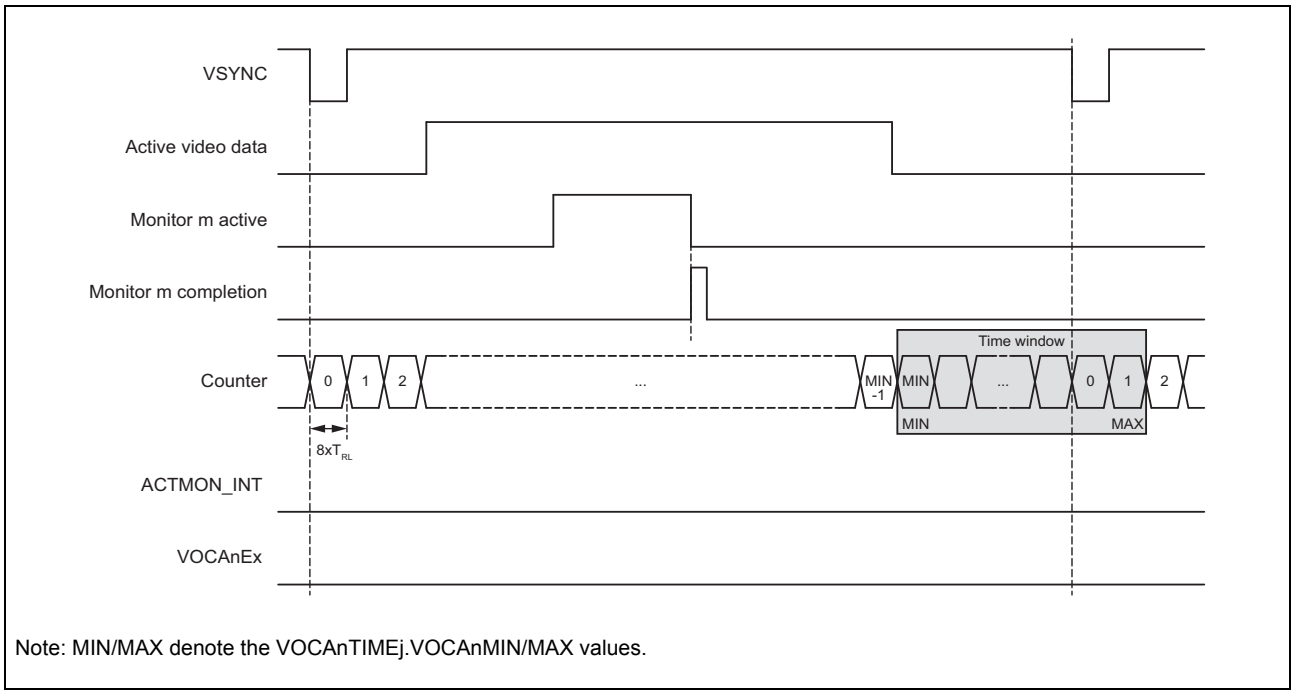
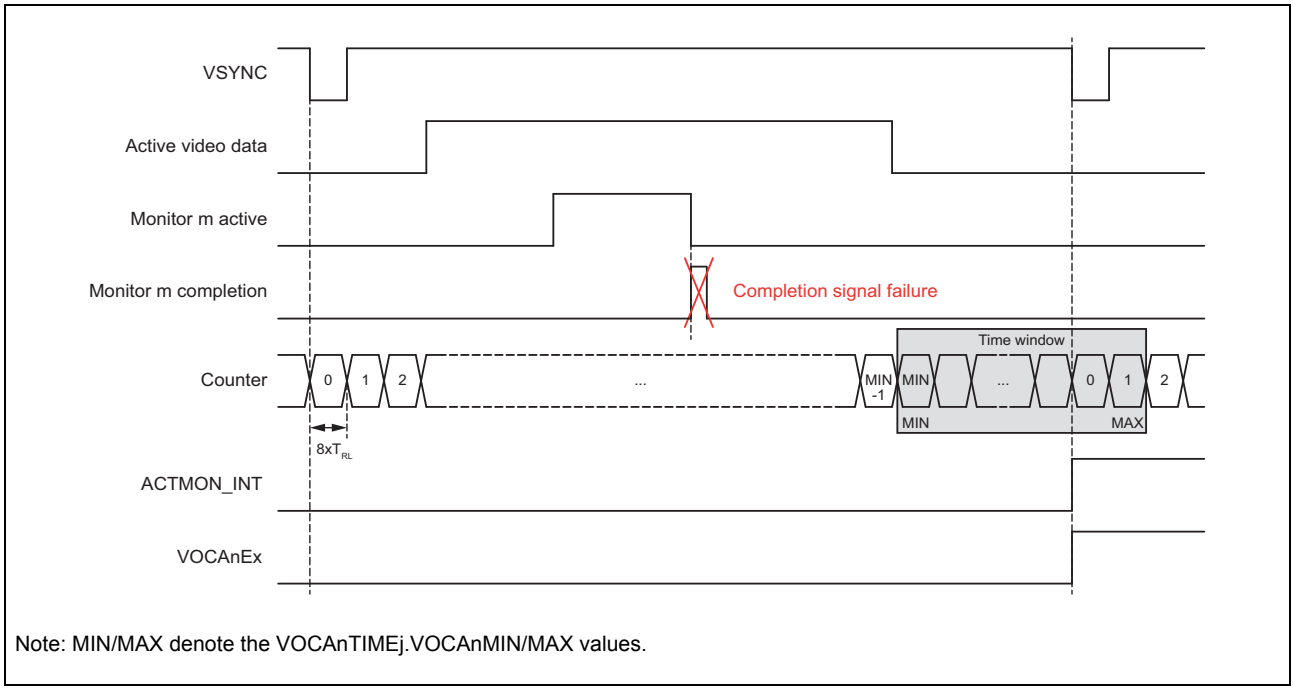


Figure 40.6 Activity Monitor process without error detection

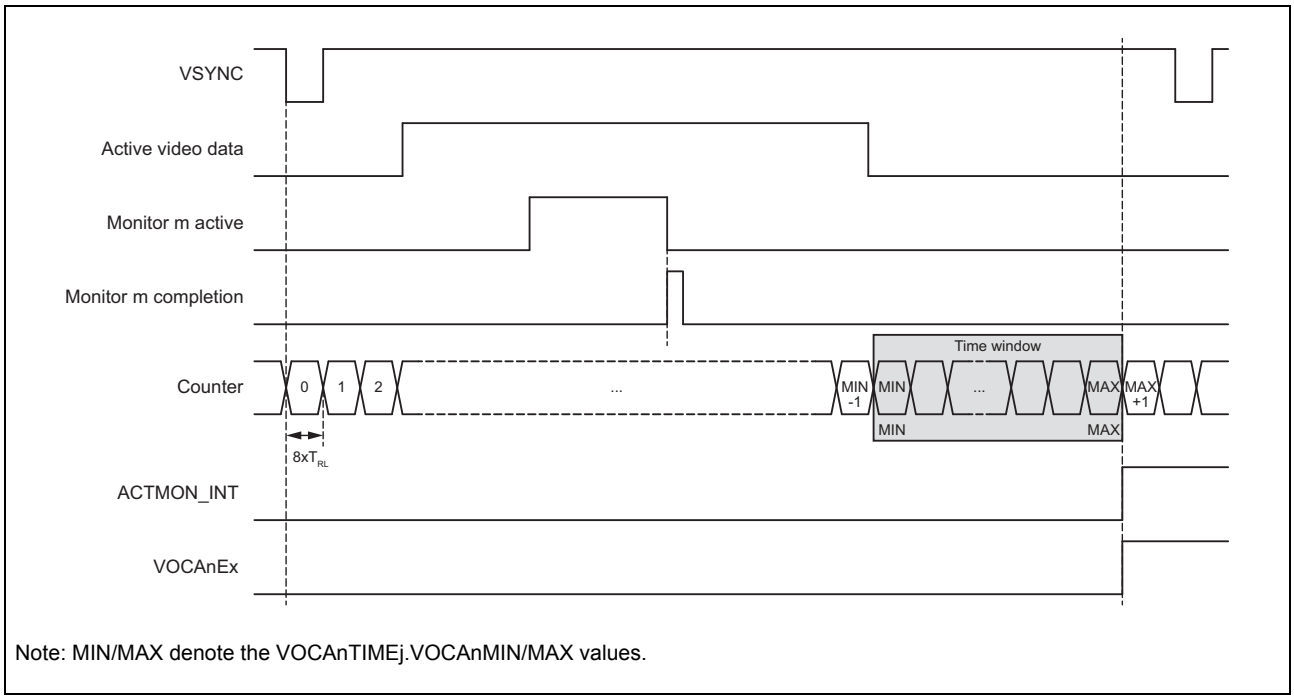
Figure 40.7 shows an example with detection of a missing Video Output Monitor completion signal. The Activity Monitor error flag VOCAnSTR.VOCAnE16/E17 for video channel 0/1 is set and the ACTMON_INT interrupt is asserted.



Note: MIN/MAX denote the VOCAnTIMEj.VOCAnMIN/MAX values.

Figure 40.7 Activity Monitor process with error detection due to missing Video Output Monitor completion signal

Figure 40.8 shows an example with detection of a missing VSYNC signal within the time window. The Activity Monitor error flag VOCAnSTR.VOCAnE16/E17 for video channel 0/1 is set and the ACTMON_INT interrupt is asserted.



Note: MIN/MAX denote the VOCAnTIMEj.VOCAnMIN/MAX values.

Figure 40.8 Activity Monitor process with error detection due to missing VSYNC in the time window

40.3.4.3 Safety aspects

By use of the Activity Monitor the operation of several video channel functions are indirectly supervised:

- VSYNC/HSYNC signals of the video channel
- video channel pixel clock
- operation of the Video Output Monitor

If any of the above modules operations fail, the Activity Monitor is able to detect such malfunction and issues error notifications.

40.4 Registers

This section contains a description of all registers of the VOCA.

40.4.1 VOCA registers overview

The VOCA is controlled and operated by the following registers:

Table 40.9 VOCA register overview

Register Name	Symbol	Address
Status register	VOCAnSTR	<VOCAn_base>
Control register	VOCAnCTL	<VOCAn_base> + 04_H
Video Output and Activity Monitor enable register	VOCAnEN	<VOCAn_base> + 08_H
Video Output Monitor channel assignment register	VOCAnCH	<VOCAn_base> + $0C_H$
Video channel 0 Activity Monitor detection time register	VOCAnTIME0	<VOCAn_base> + 10_H
Video channel 1 Activity Monitor detection time register	VOCAnTIME1	<VOCAn_base> + 14_H
Video channel 0 back porch offset register	VOCAnOFFS0	<VOCAn_base> + 20_H
Video channel 1 back porch offset register	VOCAnOFFS1	<VOCAn_base> + 24_H
Video channel 0 image size register	VOCAnDISP0	<VOCAn_base> + 28_H
Video channel 1 image size register	VOCAnDISP1	<VOCAn_base> + $2C_H$
Video channel 0 selection register	VOCAnACT0	<VOCAn_base> + 30_H
Video channel 1 selection register	VOCAnACT1	<VOCAn_base> + 34_H
Video Output Monitor discriminator register	VOCAnDIFF	<VOCAn_base> + 40_H
Video output monitor m area start point	VOCAnMmCFG0	<VOCAn_base> + 100_H + m x 20_H
Video output monitor m area size	VOCAnMmCFG1	<VOCAn_base> + 104_H + m x 20_H
Video output monitor m reference color RAM start address	VOCAnMmCFG2	<VOCAn_base> + 108_H + m x 20_H
Video output monitor m acceptance threshold	VOCAnMmCFG3	<VOCAn_base> + $10C_H$ + m x 20_H
Video output monitor m reference color 0	VOCAnMmCFG4	<VOCAn_base> + 110_H + m x 20_H
Video output monitor m reference color 1	VOCAnMmCFG5	<VOCAn_base> + 114_H + m x 20_H
Video output monitor m reference color 2	VOCAnMmCFG6	<VOCAn_base> + 118_H + m x 20_H
Video output monitor m reference color 3	VOCAnMmCFG7	<VOCAn_base> + $11C_H$ + m x 20_H
Video output monitor reference RAM register 0	VOCAnEXPD0000	<VOCAn_base> + 300_H
...	...	...
Video output monitor reference RAM register 4095	VOCAnEXPD4095	<VOCAn_base> + $42FC_H$

<VOCAn_base>

The base addresses <VOCAn_base> of the VOCAn is defined in "Register base addresses <VOCAn_base>" in the section above.

NOTES

1. The index m is representing a double-digit number for these registers.
2. The index k is representing a 4-digit number for these registers.

40.4.2 VOCAn control registers details

40.4.2.1 VOCAnSTR – Status register

This register holds various status and error information.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base>

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	VOCAn ACTSTR	VOCAn E18	VOCAn VOCSTR	VOCAnSELMON[3:0]				VOCAn E17	VOCAn E16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOCAnEm															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40.10 VOCAnSTR register contents

Bit position	Bit name	Function
31 to 25	Reserved	When read, the value after reset is returned.
24	VOCAnACTSTR	Activity Monitor status 0: not active 1: active
23	VOCAnE18	Parameter error Area of Video Output Monitor surpasses the size of video channel image If this error occurs, the related Video Output Monitor error (VOCAnEm) flag is also set. 0: no error occurred 1: error occurred*1*3
22	VOCAn VOCSTR	VOC monitor status 0: Video Output Monitor not active in current video channel frame*4 1: Video Output Monitor active in current video channel frame
21 to 18	VOCAn SELMON[3:0]	• If the vertical synchronizing signal is active low, VOCAnSELMON[3:0] indicate the number of the current video channel frame, that is monitored. • If the vertical synchronizing signal is active high, VOCAnSELMON[3:0] indicate the number of the previous video channel frame.
17	VOCAnE17	Activity Monitor of video channel 1 error 0: no error occurred 1: error occurred*1*2 **NOTE** This bit must be kept 0 in D1M1(H) and D1M1-V2 devices.
16	VOCAnE16	Activity Monitor of video channel 0 error 0: no error occurred 1: error occurred*1*2
15 to 0	VOCAnEm	Video Output Monitor area m error 0: no error occurred 1: error occurred*1*3

Note 1. These error flags are cleared via the corresponding clear bits in the control register VOCAnCTL.

Note 2. If any Activity Monitor error occurred, the Activity Monitor error interrupt ACTMON_INT is asserted.
Note 3. If any Video Output Monitor occurred, the Video Output Monitor error interrupt VOC_INT is asserted.
Note 4. As there is no Video Output Monitor checked during the switching of the video channel, the status of VOCAnVOCSTR will be 0 in this case, even if one (or more) VOC Monitor is enabled (VOCAnENm = 1).

40.4.2.2 VOCAnCTL – Control register

This register is used to mask interrupts, apply a software reset and to clear the error flags in the status register VOCAnSTR.

Access: This register can be accessed in 32-bit units.
Address: <VOCAn_base> + 04_H
Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	VOCAn MKVOC	VOCAn MKINT	VOCAn SRST	–	–	–	–	–	–	–	–	–	–	VOCAn CL18	VOCAn CL17	VOCAn CL16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOCAnCLm															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.11 VOCAnCTL register contents (1/2)

Bit position	Bit name	Function
31	VOCAnMKVOC *3	Video Output Monitor and Activity Monitor stop control 0: Video Output Monitor and Activity Monitor operating 1: Video Output Monitor and Activity Monitor stopped
28 to 19	Reserved	When read, the value after reset is returned. When written, write the value after reset.
30	VOCAnMKINT	Interrupt masking 0: VOC_INT and ACTMON_INT interrupts are generated upon error detections 1: VOC_INT and ACTMON_INT interrupts are not generated upon error detections
29	VOCAnSRST	Software reset Write: 0: no function 1: apply software reset to VOCA module Read: 0: software reset released 1: software reset active Refer also to Section (1), Software reset procedure.
18	VOCAnCL18	Parameter error clear 0: sets VOCAnCL18 = 0 1: clear VOCAnSTR.VOCAnE18 Clearing the error flag has to be done by a certain procedure. Refer to “Video Output Monitor error flag clearing procedure” in **Section 40.3.3.3** for details.

Table 40.11 VOCAnCTL register contents (2/2)

Bit position	Bit name	Function
17	VOCAnCL17	Activity Monitor of video channel 1 error flag clear 0: sets VOCAnCL17 = 0 1: clear VOCAnSTR.VOCAnE17*1 Clearing the error flag has to be done by a certain procedure. Refer to “Activity Monitors error flag clearing procedure” in Section 40.3.4.2, Activity Monitors error notification for details.
16	VOCAnCL16	Activity Monitor of video channel 0 error flag clear 0: sets VOCAnCL16 = 0 1: clear VOCAnSTR.VOCAnE16*1 Clearing the error flag has to be done by a certain procedure. Refer to “Activity Monitors error flag clearing procedure” in Section 40.3.4.2, Activity Monitors error notification for details.
15 to 0	VOCAnCLm	Video Output Monitor m error flag clear 0: sets VOCAnCLm = 0 1: clear VOCAnSTR.VOCAnEm*2 Clearing the error flag has to be done by a certain procedure. Refer to “Video Output Monitor error flag clearing procedure” in Section 40.3.3.3, Video Output Monitor error notification for details.

Note 1. If both Activity Monitor errors are cleared, the Activity Monitor error interrupt ACTMON_INT is de-asserted.

Note 2. If all Video Output Monitor errors are cleared, the Video Output Monitor error interrupt VOC_INT is de-asserted.

Note 3. The stop mode may be activated by setting VOCAnCTL.VOCAnMKVOC = 1 during Video Output Monitor error interrupt VOC_INT that indicates the detection of an area error. By this the detection of the first area m error will cause the Video Output Monitor to stop.
When Video Output monitor is reopened, VOCAnCTL.VOCAnMKVOC must be set to 0 after error handling operation.

(1) Software reset procedure

The following sequence describes the procedure to issue a software reset of the VOCA unit.

1. Apply software reset by setting VOCAnCTL.VOCAnSRST = 1.
2. Wait until software reset released state reached by polling until VOCAnCTL.VOCAnSRST = 0.
3. Wait for occurrence of next Video Data Controller interrupt INTVDCE0S0LOVSYNC.

The VOCA software reset has been completed.

40.4.2.3 VOCAnEN – Video Output and Activity Monitor enable register

This register is enable/disable each Video Output Monitor.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base> + 08_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	VOCAnEN16
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOCAnENm															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.12 VOCAnEN register contents

Bit position	Bit name	Function
31 to 17	Reserved	When read, the value after reset is returned. When written, write the value after reset.
16	VOCAnEN16	Enable/disable Activity Monitors 0: Activity Monitors disabled 1: Activity Monitors enabled
15 to 0	VOCAnENm	Enable/disable Video Output Monitor m 0: Video Output Monitor m disabled 1: Video Output Monitor m enabled

Enabled or disabling of the Activity Monitors or of a Video Output Monitor m becomes effective with the next frame of the assigned video channel, i.e. with its next VSYNC.

CAUTION

If the Video Output Monitor or the Activity Monitor are used, VOCAnEN.VOCAnENm or VOCAnEN.VOCAnEN16 must be set after above one frame period has elapsed since enabling video output by OUT_UPDATE.OUTCNT_VEN.

40.4.2.4 VOCAnCH – Video Output Monitor channel assignment register

This register is used to assign one of both video channels to each Video Output Monitor.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base> + $0C_H$

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOCAnCHm															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.13 VOCAnCH register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, write the value after reset.
15 to 0	VOCAnCHm	Video channel assignment to Video Output Monitor m 0: video channel 0 assigned to Video Output Monitor m 1: video channel 1 assigned to Video Output Monitor m

NOTE

This register value must be kept 0000 0000_H in D1M1(H) and D1M1-V2 devices.

CAUTION

Changing the VOCAnCHm bit is only allowed, when the respective Video Monitor m is disabled, i.e. VOCAnEN.VOCAnENm = 0.

40.4.2.5 VOCAnTIMEj – Video channel j Activity Monitor detection time register

These registers specify the upper and lower detection time for Activity Monitor for both video channels.

The timing values in this register are specified in units of nominal $8/f_{RL}$ = 0.033 ms, where f_{RL} is the nominal frequency 240 kHz of the Low Speed IntOsc.

Access: This register can be accessed in 32-bit units.

Address: VOCAnTIME0: <VOCAn_base> + 10_H
VOCAnTIME1: <VOCAn_base> + 14_H *1

Initial value: 0000 0000_H

Note 1. This register is only available in D1M2(H) and D1M1A devices.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	VOCAnMAXj[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	VOCAnMINj[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.14 VOCAnTIMEj register contents

Bit position	Bit name	Function
31 to 28	Reserved	When read, the value after reset is returned. When written, write the value after reset.
27 to 16	VOCAn MAXj[11:0]	Upper detection time for Activity Monitor in 0.033 ms units 0: 0 ms 1: 0.033 ms 2: 0.067 ms 3: 0.1 ms ... 4094: 136.467 ms 4095: Reserved
15 to 12	Reserved	When read, the value after reset is returned. When written, write the value after reset.
11 to 0	VOCAn MINj[11:0]	Lower detection time for Activity Monitor in 0.033 ms units 0: 0 ms 1: 0.033 ms 2: 0.067 ms 3: 0.1 ms ... 4094: 136.467 ms 4095: 136.5 ms

CAUTION

Writing to these registers is only allowed, when the Activity Monitors are disabled, i.e. VOCAnEN.VOCAnEN16 = 0.

40.4.2.6 VOCAnOFFSj – Video channel j back porch offset register

These registers specify the horizontal and vertical back porch offsets of video channel's timing signals.

Access: This register can be accessed in 32-bit units.

Address: VOCAnOFFS0: <VOCAn_base> + 20_H
VOCAnOFFS1: <VOCAn_base> + 24_H *1

Initial value: $0001\ 0001_H$

Note 1. This register is only available in D1M2(H) and D1M1A devices.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	VOCAnHOFFSj[10:0]										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	VOCAnVOFFSj[10:0]										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.15 VOCAnOFFSj register contents

Bit position	Bit name	Function
31 to 27	Reserved	When read, the value after reset is returned. When written, write the value after reset.
26 to 16	VOCAn HOFFSj[10:0]	Horizontal back porch offset of video channel j in pixels
15 to 11	Reserved	When read, the value after reset is returned. When written, write the value after reset.
10 to 0	VOCAn VOFFSj[10:0]	Vertical back porch offset of video channel j in lines

Refer to Section 40.3.3.1, Video Monitor area definition for details about the offset definition.

CAUTION

This register value needs to be set before it can be used for any monitoring purposes.

40.4.2.7 VOCAnDISPj – Video channel j image size register

These registers specify size of the image of the video channels.

Access: This register can be accessed in 32-bit units.

Address: VOCAnDISP0: <VOCAn_base> + 28_H
VOCAnDISP1: <VOCAn_base> + $2C_H$ *1

Initial value: $0001\ 0001_H$

Note 1. This register is only available in D1M2(H) and D1M1A devices.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	VOCAnHSIZEj[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	VOCAnVSIZEj[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.16 VOCAnDISPj register contents

Bit position	Bit name	Function
31 to 28	Reserved	When read, the value after reset is returned. When written, write the value after reset.
27 to 16	VOCAn HSIZEj[11:0]	Horizontal size of the video channel j image in pixels
15 to 12	Reserved	When read, the value after reset is returned. When written, write the value after reset.
11 to 0	VOCAn VSIZEj[11:0]	Vertical size of the video channel j image in pixels

CAUTION

This register value needs to be set before it can be used for any monitoring purposes.

40.4.2.8 VOCAnACTj – Video channel j selection register

These registers specify which video channel is available in the device.

Access: This register can be accessed in 32-bit units.

Address: VOCAnACT0: <VOCAn_base> + 30_H
VOCAnACT1: <VOCAn_base> + 34_H

Initial value: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	VOCAn VOUTj
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 40.17 VOCAnACTj register contents

Bit position	Bit name	Function
31 to 1	Reserved	When read, the value after reset is returned. When written, write the value after reset.
0	VOCAn VOUTj	Video channel j availability selection 0: Video channel j is not available 1: Video channel j is available

NOTES

1. Video channel j needs to be selected as available (VOCAnVOUTj = 1) before it can be used for any monitoring purposes.
2. In D1M1(H) devices, video channel 0 only needs to be selected as available (VOCAnVOUT0=1, VOCAnVOUT1=0) before it can be used for any monitoring purposes.

40.4.2.9 VOCAnDIFF – Video Output Monitor discriminator register

This register shows the calculated discriminator value D for a single Video Output Monitor area. Refer to Section 40.3.1, Video Output Monitoring Basic Procedure for details about the discriminator calculation.

This register is to be used only for debugging purposes and in the way shown below:

1. Disable all Video Output Monitor areas m (VOCAnEN.VOCAnENm = 0).
2. Enable the single area m to be evaluated (VOCAnEN.VOCAnENm = 1).
3. Disable the evaluated single area m (VOCAnEN.VOCAnENm = 0).
4. Wait until monitoring becomes inactive, i.e. until VOCAnSTR.VOCAnVOCSTR = 0.
5. The last calculated discriminator can be read from the VOCAnDIFF register.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base> + 40_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	VOCAn DIFF0[17:15]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOCAnDIFF0[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 40.18 VOCAnDIFF register contents

Bit position	Bit name	Function
31 to 18	Reserved	When read, the value after reset is returned. When written, write the value after reset.
17 to 0	VOCAn DIFF0[17:0]	Last discriminator value

40.4.2.10 VOCAnMmCFG0 – Video Output Monitor m area register

These registers specify the screen area to be monitored by the Video Output Monitor m.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base> + 100_H + m x 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	VOCAnMmHSTRT[10:0]										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	VOCAnMmVSTRT[10:0]										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.19 VOCAnMmCFG0 register contents

Bit position	Bit name	Function
31 to 27	Reserved	When read, the value after reset is returned. When written, write the value after reset.
26 to 16	VOCAn MmHSTRT[10:0]	Video Output Monitor m area horizontal start point in pixels
15 to 11	Reserved	When read, the value after reset is returned. When written, write the value after reset.
10 to 0	VOCAn MmVSTRT[10:0]	Video Output Monitor m area vertical start point in pixels

CAUTION

Changing this register is only allowed, when the respective Video Monitor m is disabled, i.e. VOCAnEN.VOCAnENm = 0.

40.4.2.11 VOCAnMmCFG1 – Video Output Monitor m area size register

These registers specify the area size to be monitored by the Video Output Monitor m.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base> + 104_H + m x 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	VOCAnMmHSIZE[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–	VOCAnMmVSIZE[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.20 VOCAnMmCFG1 register contents

Bit position	Bit name	Function
31 to 24	Reserved	When read, the value after reset is returned. When written, write the value after reset.
23 to 16	VOCAn MmHSIZE[7:0]	Video Output Monitor m area horizontal size in pixels **Note:** Setting the value more than 128 is not allowed.
15 to 8	Reserved	When read, the value after reset is returned. When written, write the value after reset.
7 to 0	VOCAn MmVSIZE[7:0]	Video Output Monitor m area vertical size in pixels **Note:** Setting the value more than 128 is not allowed.

CAUTION

Changing this register is only allowed, when the respective Video Monitor m is disabled, i.e. VOCAnEN.VOCAnENm = 0.

40.4.2.12 VOCAnMmCFG2 – Video Output Monitor m reference color RAM address register

These registers specify the first address of the 2-bit color index in the reference RAM for Video Output Monitor m reference colors.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base> + 108_H + m x 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	VOCAnMmADDR[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.21 VOCAnMmCFG2 register contents

Bit position	Bit name	Function
31 to 12	Reserved	When read, the value after reset is returned. When written, write the value after reset.
11 to 0	VOCAn MmADDR[11:0]	Video Output Monitor m reference RAM start address.

CAUTION

Changing this register is only allowed, when the respective Video Monitor m is disabled, i.e. VOCAnEN.VOCAnENm = 0.

40.4.2.13 VOCAnMmCFG3 – Video Output Monitor m acceptance threshold register

These registers specify the threshold to be compared against the calculated discriminator.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base> + $10C_H$ + m x 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	VOCAnMm THSH[17:16]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	VOCAnMmTHSH[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.22 VOCAnMmCFG3 register contents

Bit position	Bit name	Function
31 to 18	Reserved	When read, the value after reset is returned. When written, write the value after reset.
17 to 0	VOCAn MmTHSH[17:0]	Video Output Monitor m acceptance threshold Setting VOCAnMmTHSH[17:0] = 0 means, the Video Output Monitor always reports an error.

CAUTION

Changing this register is only allowed, when the respective Video Monitor m is disabled, i.e. VOCAnEN.VOCAnENm = 0.

NOTE

In the use case the worst threshold is assumed for a 25 % difference per pixel color on a 128 x 128 pixel monitor area. With this case the full threshold range of VOCAnMmTHSH[17:0] which is limited to 18 bit is sufficient.

When using a monitoring size of over 64 x 64 pixel please pay attention that the maximum value of the threshold (and discriminator) range given by VOCAnMmTHSH[17:0] is saturated with 0x3FFFF.

40.4.2.14 VOCAnMmCFG4 – Video Output Monitor m reference color 0 register

These registers specify the minimum/maximum values of reference color 0.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base> + 110_H + m x 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	VOCAnMmRUP0[3:0]				VOCAnMmGUP0[3:0]				VOCAnMmBUP0[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	VOCAnMmRLO0[3:0]				VOCAnMmGLO0[3:0]				VOCAnMmBLO0[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.23 VOCAnMmCFG4 register contents

Bit position	Bit name	Function
31 to 28	Reserved	When read, the value after reset is returned. When written, write the value after reset.
27 to 24	VOCAn MmRUP0[3:0]	Video Output Monitor m reference color 0: red upper limit
23 to 20	VOCAn MmGUP0[3:0]	Video Output Monitor m reference color 0: green upper limit
19 to 16	VOCAn MmBUP0[3:0]	Video Output Monitor m reference color 0: blue upper limit
15 to 12	Reserved	When read, the value after reset is returned. When written, write the value after reset.
11 to 8	VOCAn MmRLO0[3:0]	Video Output Monitor m reference color 0: red lower limit
7 to 4	VOCAn MmGLO0[3:0]	Video Output Monitor m reference color 0: green lower limit
3 to 0	VOCAn MmBLO0[3:0]	Video Output Monitor m reference color 0: blue lower limit

CAUTION

Changing this register is only allowed, when the respective Video Monitor m is disabled, i.e. VOCAnEN.VOCAnENm = 0.

40.4.2.15 VOCAnMmCFG5 – Video Output Monitor m reference color 1 register

These registers specify the minimum/maximum values of reference color 1.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base> + 114_H + m x 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	VOCAnMmRUP1[3:0]				VOCAnMmGUP1[3:0]				VOCAnMmBUP1[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	VOCAnMmRLO1[3:0]				VOCAnMmGLO1[3:0]				VOCAnMmBLO1[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.24 VOCAnMmCFG5 register contents

Bit position	Bit name	Function
31 to 28	Reserved	When read, the value after reset is returned. When written, write the value after reset.
27 to 24	VOCAn MmRUP1[3:0]	Video Output Monitor m reference color 1: red upper limit
23 to 20	VOCAn MmGUP1[3:0]	Video Output Monitor m reference color 1: green upper limit
19 to 16	VOCAn MmBUP1[3:0]	Video Output Monitor m reference color 1: blue upper limit
15 to 12	Reserved	When read, the value after reset is returned. When written, write the value after reset.
11 to 8	VOCAn MmRLO1[3:0]	Video Output Monitor m reference color 1: red lower limit
7 to 4	VOCAn MmGLO1[3:0]	Video Output Monitor m reference color 1: green lower limit
3 to 0	VOCAn MmBLO1[3:0]	Video Output Monitor m reference color 1: blue lower limit

CAUTION

Changing this register is only allowed, when the respective Video Monitor m is disabled, i.e. VOCAnEN.VOCAnENm = 0.

40.4.2.16 VOCAnMmCFG6 – Video Output Monitor m reference color 2 register

These registers specify the minimum/maximum values of reference color 2.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base> + 118_H + m x 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	VOCAnMmRUP2[3:0]				VOCAnMmGUP2[3:0]				VOCAnMmBUP2[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	VOCAnMmRLO2[3:0]				VOCAnMmGLO2[3:0]				VOCAnMmBLO2[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.25 VOCAnMmCFG6 register contents

Bit position	Bit name	Function
31 to 28	Reserved	When read, the value after reset is returned. When written, write the value after reset.
27 to 24	VOCAn MmRUP2[3:0]	Video Output Monitor m reference color 2: red upper limit
23 to 20	VOCAn MmGUP2[3:0]	Video Output Monitor m reference color 2: green upper limit
19 to 16	VOCAn MmBUP2[3:0]	Video Output Monitor m reference color 2: blue upper limit
15 to 12	Reserved	When read, the value after reset is returned. When written, write the value after reset.
11 to 8	VOCAn MmRLO2[3:0]	Video Output Monitor m reference color 2: red lower limit
7 to 4	VOCAn MmGLO2[3:0]	Video Output Monitor m reference color 2: green lower limit
3 to 0	VOCAn MmBLO2[3:0]	Video Output Monitor m reference color 2: blue lower limit

CAUTION

Changing this register is only allowed, when the respective Video Monitor m is disabled, i.e. VOCAnEN.VOCAnENm = 0.

40.4.2.17 VOCAnMmCFG7 – Video Output Monitor m reference color 3 register

These registers specify the minimum/maximum values of reference color 3.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base> + $11C_H$ + m x 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	VOCAnMmRUP3[3:0]				VOCAnMmGUP3[3:0]				VOCAnMmBUP3[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	VOCAnMmRLO3[3:0]				VOCAnMmGLO3[3:0]				VOCAnMmBLO3[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.26 VOCAnMmCFG7 register contents

Bit position	Bit name	Function
31 to 28	Reserved	When read, the value after reset is returned. When written, write the value after reset.
27 to 24	VOCAn MmRUP3[3:0]	Video Output Monitor m reference color 3: red upper limit
23 to 20	VOCAn MmGUP3[3:0]	Video Output Monitor m reference color 3: green upper limit
19 to 16	VOCAn MmBUP3[3:0]	Video Output Monitor m reference color 3: blue upper limit
15 to 12	Reserved	When read, the value after reset is returned. When written, write the value after reset.
11 to 8	VOCAn MmRLO3[3:0]	Video Output Monitor m reference color 3: red lower limit
7 to 4	VOCAn MmGLO3[3:0]	Video Output Monitor m reference color 3: green lower limit
3 to 0	VOCAn MmBLO3[3:0]	Video Output Monitor m reference color 3: blue lower limit

CAUTION

Changing this register is only allowed, when the respective Video Monitor m is disabled, i.e. VOCAnEN.VOCAnENm = 0.

40.4.2.18 VOCAnEXPDk – Video Output Monitor reference RAM register k (k = 0 to 4095)

These registers define the 2-bpp CLUT2 indices of the reference colors.
For further details refer to Section 40.3.3.2, Reference color.

Access: This register can be accessed in 32-bit units.

Address: <VOCAn_base> + 300_H + k x 4_H

Initial value: 0000 $XXXX_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	I0[1:0]		I1[1:0]		I2[1:0]		I3[1:0]		I4[1:0]		I5[1:0]		I6[1:0]		I7[1:0]	
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 40.27 VOCAnEXPDk register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, write the value after reset.
15 to 14	I0[1:0]	CLUT2 index of pixel x
13 to 12	I1[1:0]	CLUT2 index of pixel x+1
11 to 10	I2[1:0]	CLUT2 index of pixel x+2
9 to 8	I3[1:0]	CLUT2 index of pixel x+3
7 to 6	I4[1:0]	CLUT2 index of pixel x+4
5 to 4	I5[1:0]	CLUT2 index of pixel x+5
3 to 2	I6[1:0]	CLUT2 index of pixel x+6
1 to 0	I7[1:0]	CLUT2 index of pixel x+7

CAUTION

Changing this register is only allowed, when the respective Video Monitor m is disabled, i.e. VOCAnEN.VOCAnENm = 0.

40.5 Usage Notes

40.5.1 Procedure for entering and resuming DEEPSTOP

If VOCA and all DISCOMn are not used, always mask INTVOCAERR.
In other case, mask error signals of INTVOCAERR on ECM, before entering DEEPSTOP.
And after resuming from DEEPSTOP, clear the error status flag of INTVOCAERR, clear error status flags of VOCA and set expected mask setting of ECM again.

(a) in case VOCA is used

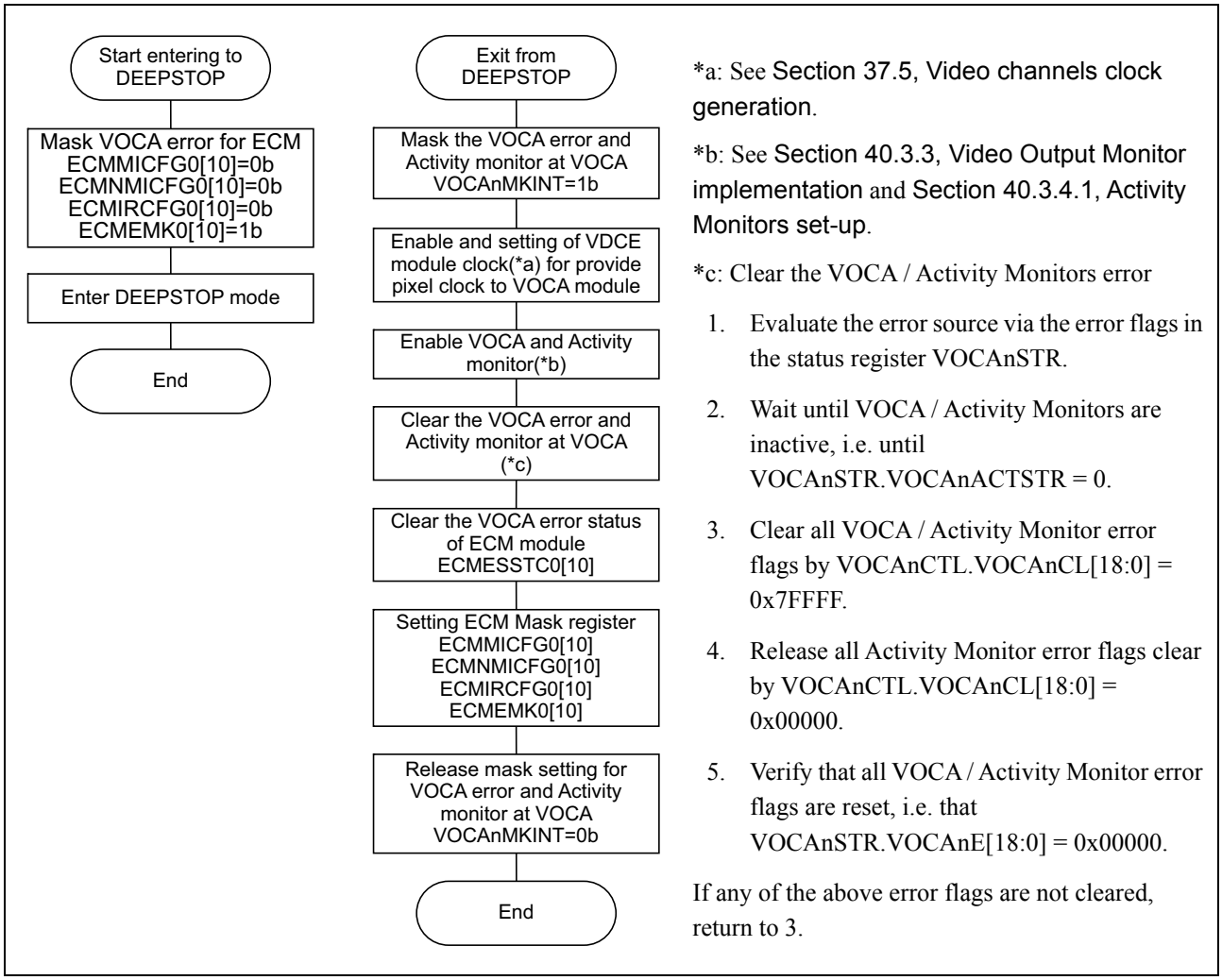


Figure 40.9 Workaround of unintended VOCA error (if VOCA is used)

(b) in case VOCA is not used

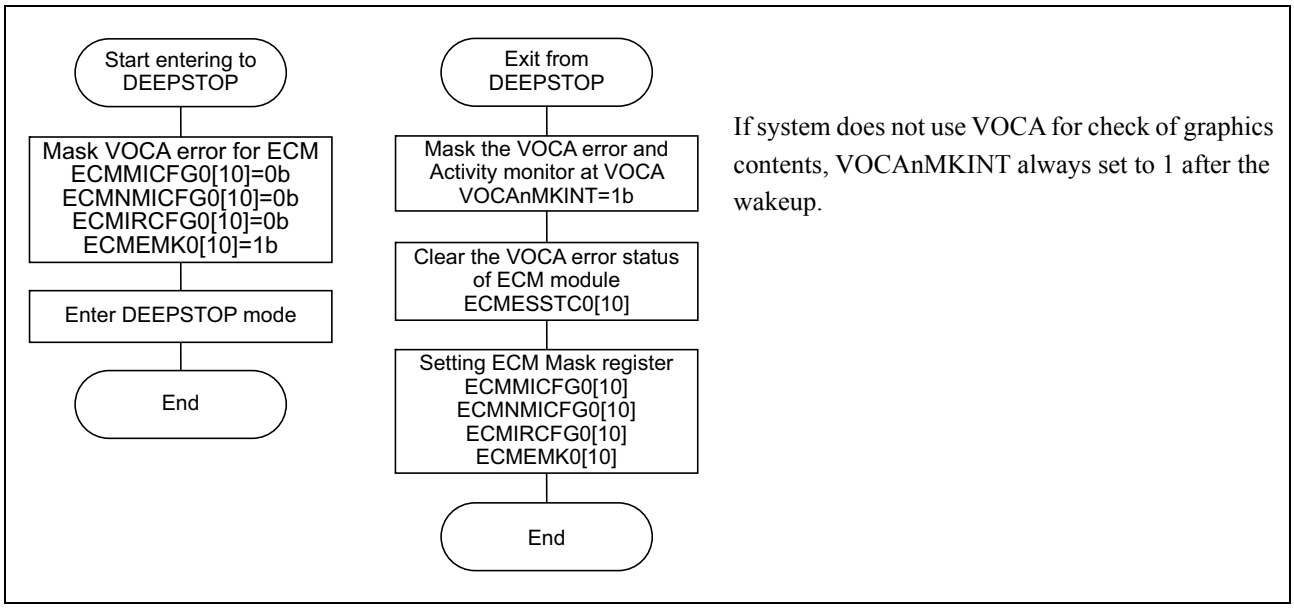


Figure 40.10 Workaround of unintended VOCA error (if VOCA is not used)

Section 41 Display Output Comparator (DISCOM)

This section contains a generic description of the Display Output Comparator (DISCOM).

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

41.1 Overview of the RH850/D1L/D1M Display Output Comparator

41.1.1 Units

This microcontroller has the following number of units of the Display Output Comparator.

Table 41.1 Units

Display Output Comparator (DISCOM)	D1L1	D1L2(H)	D1M1(H) D1M1-V2	D1M1A D1M2(H)
Units	0	0	2	4
Names	–	–	DISCOM0 to DISCOM1	DISCOM0 to DISCOM3

Unit index n

Throughout this section, the individual units of the Display Output Comparators are identified by the index “n” (n = 0 to 3).

41.1.2 Register addresses

All Display Output Comparator register addresses are given as address offsets from the individual base addresses <DISCOMn_base>.
The <DISCOMn_base> addresses of each DISCOMn are listed in the following table:

Table 41.2 Register base addresses <DISCOMn_base>

DISCOMn unit	<DISCOMn_base> address
DISCOM0	F200 9000_H
DISCOM1	F200 9400_H
DISCOM2	F200 9800_H
DISCOM3	F200 $9C00_H$

41.1.3 Clock supply

All Display Output Comparators provide two clock inputs.

Table 41.3 Clock supply

DISCOMn unit	DISCOMn clock	Connected to
DISCOM0 DISCOM1	Register access clock PCLK	Clock Controller C_ISO_PCLK
	Video channel 0 pixel clock	Video channel clock generators C_ISO_VDCE0CLK
DISCOM2 DISCOM3	Register access clock PCLK	Clock Controller C_ISO_PCLK
	Video channel 1 pixel clock	Video channel clock generators C_ISO_VDCE1CLK

41.1.4 Interrupts

The Display Output Comparators can generate the following interrupt requests:

Table 41.4 DISCOMn interrupt requests

DISCOMn signals	Function	Connected to
CMPI	Comparison error interrupts	All DISCOM interrupts are logically OR combined with the interrupt outputs of the Video Output Checker (VOCA) and input to the Error Control Module INTVOCAERR

41.1.5 Reset sources

The Display Output Comparators and their registers are initialized by the following reset signal:

Table 41.5 Reset sources

DISCOMn unit	Reset signal
DISCOMn	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

41.1.6 Connection of the DISCOM units to the Video Channels

The connection of the Display Output Comparator modules to the video channels of the Video Display Controller is shown in the following figure.

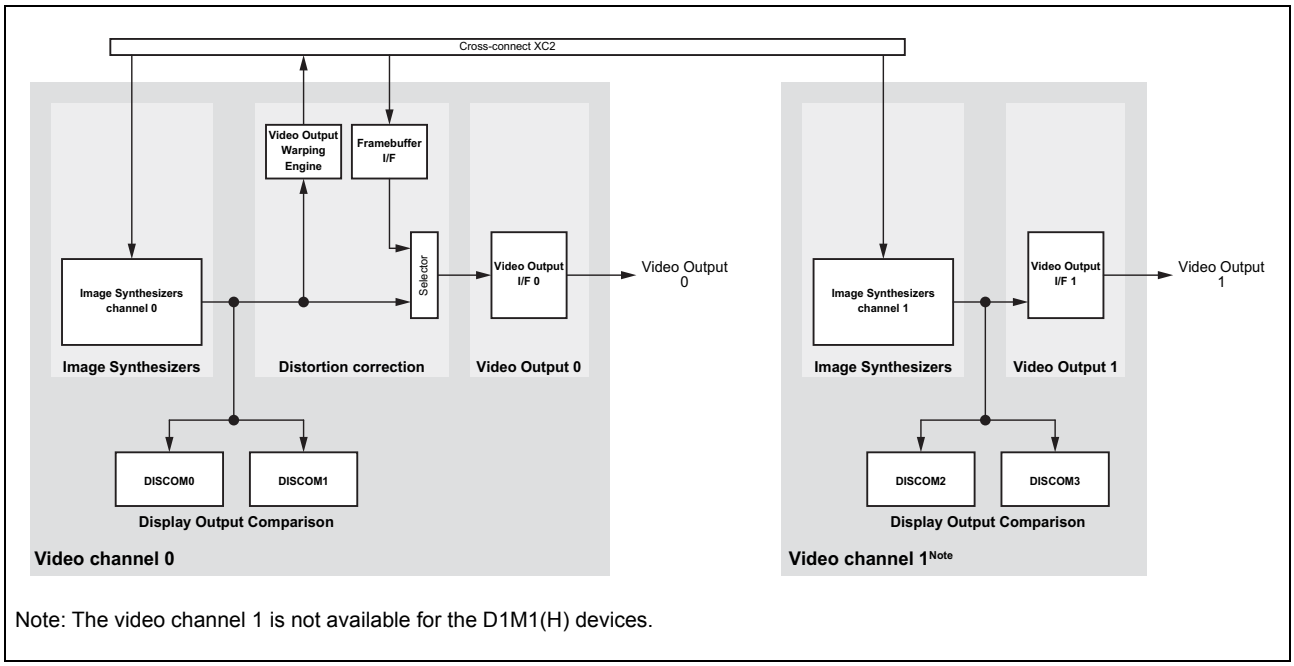


Figure 41.1 DISCOM connections to the video channels

Two rectangular areas of each video channel can be monitored.

41.2 Features

The Display Out Comparison Unit (DISCOM) checks whether the data output from the Video Display Controller (VDCE) agrees with the expected graphics data. This checking is accomplished by comparing the CRC code of the data output from the graphics display module with the pre-calculated CRC code of the expected graphics data.

This module has the following features.

- Comparison of Data after α Blending
  The CRC code of the graphics data obtained after alpha blending in the graphics display module can be compared with the expected CRC code.
- Rectangular Area Specification
  The rectangular area can be specified based on the graphics data output from the graphics display module and its CRC code can be compared with the expected CRC code.
- Pixel format: 32-bit αRGB888 with fixed a = FF_H
- CRC is calculated from most significant byte (MSB) to least significant byte (LSB)

41.3 Block Diagram

An overall block diagram of this module is shown in figure below.

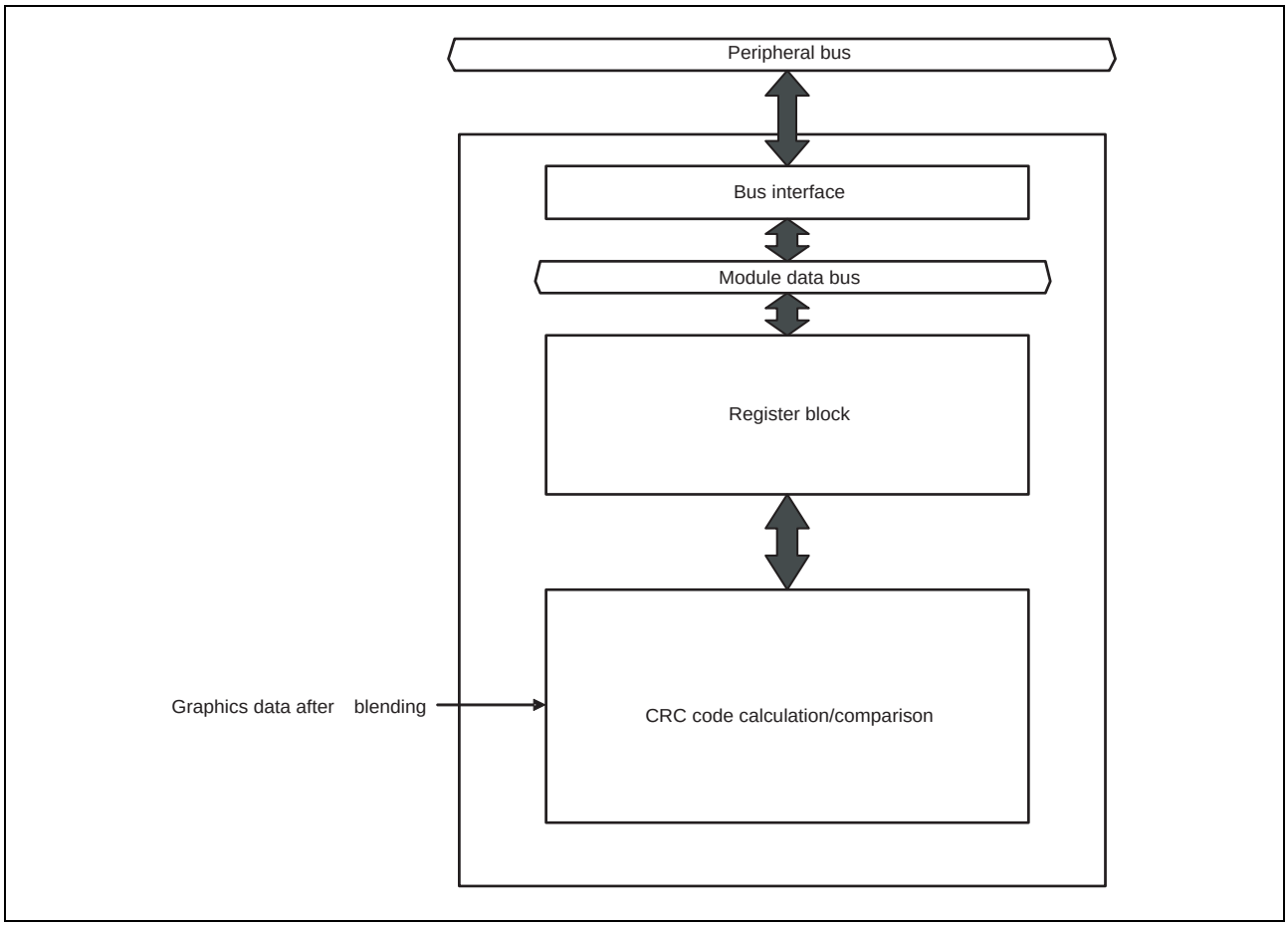


Figure 41.2 Block Diagram

41.4 Register Descriptions

The following table shows the register configuration.

Table 41.6 Register Configuration

Module Name	Register Name	Symbol	R/W	Address	Access Size
DISCOMn	Control register	DOCMCR	R/W	<DISCOMn_base> + 00_H	32
DISCOMn	Status register	DOCMSTR	R	<DISCOMn_base> + 04_H	32
DISCOMn	Status clear register	DOCMCLSTR	R/W	<DISCOMn_base> + 08_H	32
DISCOMn	Interrupt enable register	DOCMIENR	R/W	<DISCOMn_base> + $0C_H$	32
DISCOMn	Operation parameter setting register	DOCMPMR	R/W	<DISCOMn_base> + 14_H	32
DISCOMn	Expected CRC code register	DOCMECRCR	R/W	<DISCOMn_base> + 18_H	32
DISCOMn	Calculated CRC code value register	DOCMCCRCR	R	<DISCOMn_base> + $1C_H$	32
DISCOMn	Horizontal start position setting register	DOCMSPXR	R/W	<DISCOMn_base> + 20_H	32
DISCOMn	Vertical start position setting register	DOCMSPYR	R/W	<DISCOMn_base> + 24_H	32
DISCOMn	Horizontal size setting register	DOCMSZXR	R/W	<DISCOMn_base> + 28_H	32
DISCOMn	Vertical size setting register	DOCMSZYR	R/W	<DISCOMn_base> + $2C_H$	32
DISCOMn	CRC code initialization register	DOCMCRCIR	R/W	<DISCOMn_base> + 30_H	32

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

41.4.1 Control Register (DOCMCR)

DOCMCR turns CRC code comparison on or off.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMPRU
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMPR
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 41.7 DOCMCR register contents

Bit	Bit Name	Description
31 to 17	–	Reserved These bits are always read as 0. The write value should always be 0.
16	CMPRU	Display Out Comparison Update Value Reflects the internal update of the CMPR bit. It should be checked that this bit is 0 before updating the registers other than the CMPR bit in DOCMCR, DOCMCLSTR, and DOCMIENR. For details, see Section 41.5.7, Register Update Timing.
15 to 1	-	Reserved These bits are always read as 0. The write value should always be 0.
0	CMPR	Display Out Comparison Execution Executes display out comparison. This bit is loaded inside when the start of the valid period of the graphics data is detected. For details, see Section 41.5.7, Register Update Timing. 0: Stops display out comparison. 1: Executes display out comparison.

41.4.2 Status Register (DOCMSTR)

DOCMSTR returns the comparison result of the CRC code. The result is reflected in this register when the end of the valid period of the graphics data is detected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMPST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 41.8 DOCMSTR register contents

Bit	Bit Name	Description
31, 30	–	Reserved Values read from these bits are undefined. The write value should always be 0.
29 to 1	–	Reserved These bits are always read as 0. The write value should always be 0.
0	CMPST	Display Out Comparison Status 0: Compared CRC codes match. 1: Compared CRC codes do not match.

41.4.3 Status Clear Register (DOCMCLSTR)

Writing 1 to the CMPCLST bit causes clearing of the CMPST bit in DOCMSTR to 0. However, clearing of the CMPST bit in DOCMSTR after 1 is written to the CMPCLST bit takes a fixed amount of time. Confirm that the CMPST bit in DOCMSTR is actually cleared after writing 1 to the CMPCLST bit.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMP CLST
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 41.9 DOCMCLSTR register contents

Bit	Bit Name	Description
31 to 1	–	Reserved These bits are always read as 0. The write value should always be 0.
0	CMPCLST	Display Out Comparison Status Clear Setting this bit to 1 clears the CMPST bit in DOCMSTR to 0. This bit is always read as 0.

41.4.4 Interrupt Enable Register (DOCMIENR)

DOCMIENR enables interrupt of the corresponding status bits in DOCMSTR.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMPIEN
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 41.10 DOCMIENR register contents

Bit	Bit Name	Description
31 to 1	–	Reserved These bits are always read as 0. The write value should always be 0.
0	CMPIEN	Display Out Comparison Mismatch Detection Interrupt Enable Enables/disables the display out comparison mismatch detection interrupt (CMPI) when the CMPST bit in DOCMSTR is set to 1. 0: Disables the display out comparison mismatch detection interrupt (CMPI). 1: Enables the display out comparison mismatch detection interrupt (CMPI).

41.4.5 Operation Parameter Setting Register (DOCMPMR)

DOCMPMR selects the graphics data and sets the pixel format.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	CMPBT
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMPDFA[7:0]								CMP DAUF	-	-	-	CMPSELP[3:0]			
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W

Table 41.11 DOCMPMR register contents

Bit	Bit Name	Description
31 to 17	–	Reserved These bits are always read as 0. The write value should always be 0.
16	CMPBT	Pixel Format Data Width 0: 32-bit 1: prohibited **CAUTION** The default value “0” must not be changed.
15 to 8	CMPDFA [7:0]	Display Out Comparison Default α Value FF_H: α = FF_H all others: prohibited **CAUTION** The default value “00_H” must be changed to “FF_H”.
7	CMPDAUF	Display Out Comparison Default α Value Use 0: prohibited 1: use default α value CMPDFA[7:0] = FF_H **CAUTION** The default value “0” must be changed to “1”.
6 to 4	–	Reserved These bits are always read as 0. The write value should always be 0.
3 to 0	CMPSELP [3:0]	Display Out Comparison input selection Selects the graphics data for CRC code comparison. 9: Selects graphics data after α blending. Other than above: Setting prohibited **CAUTION** The default value “0” must be changed to “9”.

41.4.6 Expected CRC Code Register (DOCMECRCR)

DOCMECRCR specifies the CRC code of the expected graphics data.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CMPECRC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMPECRC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.12 DOCMECRCR register contents

Bit	Bit Name	Description
31 to 0	CMPECRC [31:0]	Expected Display Out Comparison CRC Code The expected CRC code value of the selected graphics data or rectangular area

41.4.7 Calculated CRC Code Value Register (DOCMCCRCR)

The CRC code calculation result can be read from this register. The calculation result is reflected in this register when the end of the valid period of the graphics data is detected.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CMPCCRC[31:16]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMPCCRC[15:0]															
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 41.13 DOCMCCRCR register contents

Bit	Bit Name	Description
31 to 0	CMPCCRC [31:0]	Calculated Display Out Comparison CRC Code Value The calculated CRC code value of the selected graphics data or rectangular area

41.4.8 Horizontal Start Position Setting Register (DOCMSPXR)

DOCMSPXR specifies the horizontal start position of the rectangular area for which the CRC code is calculated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMPSPX[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.14 DOCMSPXR register contents

Bit	Bit Name	Description
31 to 11	–	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	CMPSPX [10:0]	Display Out Comparison Horizontal Start Position Specifies the horizontal start position of the rectangular area for which the CRC code is calculated. The set value should be smaller than or equal to the horizontal size of the graphics data.

41.4.9 Vertical Start Position Setting Register (DOCMSPYR)

DOCMSPYR specifies the vertical start position of the rectangular area for which the CRC code is calculated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMPSPY[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.15 DOCMSPYR register contents

Bit	Bit Name	Description
31 to 11	–	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	CMPSPY [10:0]	Display Out Comparison Vertical Start Position Specifies the vertical start position of the rectangular area for which the CRC code is calculated. The set value should be smaller than or equal to the vertical size of the graphics data.

41.4.10 Horizontal Size Setting Register (DOCMSZXR)

DOCMSZXR specifies the horizontal size of the rectangular area for which the CRC code is calculated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMPSZX[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.16 DOCMSZXR register contents

Bit	Bit Name	Description
31 to 11	–	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	CMPSZX [10:0]	Display Out Comparison Horizontal Size Specifies the horizontal size of the rectangular area for which the CRC code is calculated. The value should be set as follows: Horizontal size of the graphics data ≥ Horizontal start position (CMPSPX[10:0]) + Horizontal size (CMPSZX[10:0]).

41.4.11 Vertical Size Setting Register (DOCMSZYR)

DOCMSZYR specifies the vertical size of the rectangular area for which the CRC code is calculated.

Bit:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	-	-	-	-	-	CMPSZY[10:0]										
Initial value:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 41.17 DOCMSZYR register contents

Bit	Bit Name	Description
31 to 11	–	Reserved These bits are always read as 0. The write value should always be 0.
10 to 0	CMPSZY [10:0]	Display Out Comparison Vertical Size Specifies the vertical size of the rectangular area for which the CRC code is calculated. The value should be set as follows: Vertical size of the graphics data ≥ Vertical start position (CMPSPY[10:0]) + Vertical size (CMPSZY[10:0]).

41.4.12 CRC Code Initialization Register (DOCMCRCIR)

DOCMCRCIR is used to specify the initial value of the CRC code.

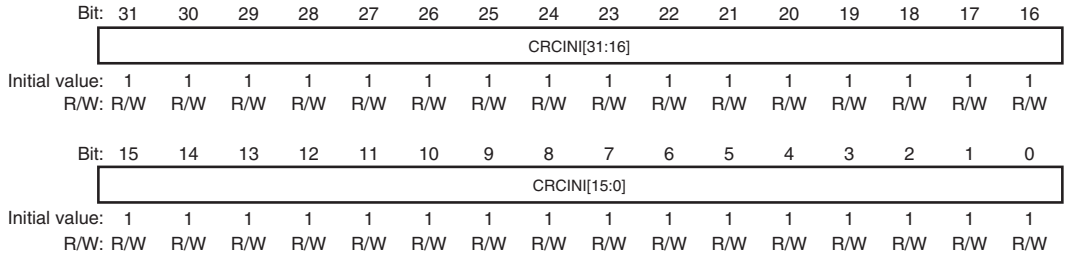


Table 41.18 DOCMCRCIR register contents

Bit	Bit Name	Description
31 to 0	CRCINI [31:0]	Display Output CRC Comparison Initial Value These bits specify the initial value of the CRC for the rectangular area of the selected graphics data.

41.5 Operation

41.5.1 Overview of Operations

This module is capable of calculating the CRC code of the arbitrary rectangular area of graphics data. By comparing the CRC code with the pre-calculated expected CRC code value, this module can detect whether the display output is obtained as expected.

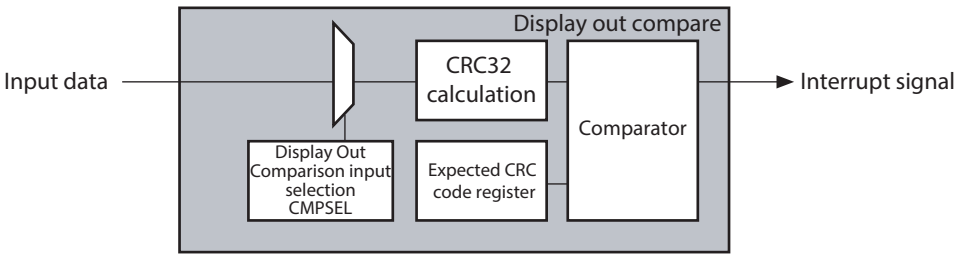
Main features of this module are as follows.

- Arbitrary rectangular area of the selected graphics data can be specified
- Pixel format: 32-bit αRGB8888 with fixed a = FF_H
- Interrupt is generated when the compared CRC codes do not match.

41.5.2 System Configuration

This module is configured as shown in the figure below.

The CRC code is calculated after receiving the graphics data output from the graphics display module. The calculated CRC code is then compared with the pre-calculated expected CRC code value.



Note: The input data selection DOCMPMR.CMPSEL[3:0] must be set to 9.

Figure 41.3 System Configuration

41.5.3 CRC Calculation Method

The display out comparison unit generates a 32-bit CRC code by using the following CRC polynomial (IEEE802.3).

$$x^{32}+ x^{26}+ x^{23}+ x^{22}+ x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^{8}+ x^{7}+ x^{5}+ x^{4}+ x^{2}+ x+1$$

CRC is sequentially calculated beginning with LSB in pixel units.

In other words it is calculated in units of 32 bits. During CRC calculation of data, pixel data is input when the graphics data is output (top left to bottom right).

41.5.4 Pixel Format

41.5.4.1 Pixel Format Specification

DOCMPMR specifies the pixel format. The pixel formats are given in the following table.

Table 41.19 Pixel Format

Bits in DOCMPMR		CMPBT	CMPDFA[7:0]	CMPDAUF
32 bits/pixel	αRGB8888 (fixed a = FF_H)	0	FF_H	1

Data arrangement for each pixel format is given below.

- αRGB8888 (32 bits/pixel)

1 1 1 1 1 1 1 1 b23 b16	b15 b8	b7 b0
α = FF_H, 8 bits	Red 8 bits	Green 8 bits	Blue 8 bits

41.5.5 Rectangular Area Settings

Based on the selected graphics data, the start position and size of the rectangular area for which the CRC code is calculated can be set with the registers. The following figure and table show such a rectangular area and the register settings for the area.

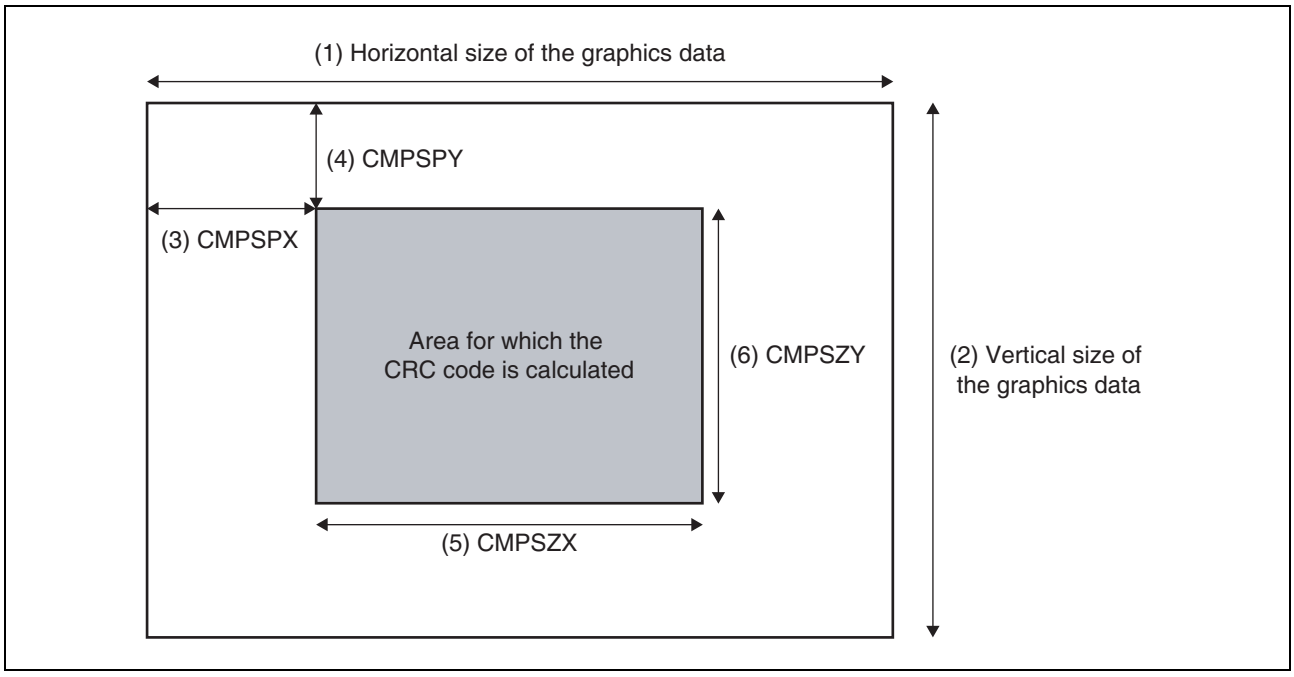


Figure 41.4 Rectangular Area for which the CRC Code is Calculated

Table 41.20 Register Settings for the Rectangular Area for which the CRC Code is Calculated

No.	Symbol in the Figure	Register Used for Setting	Description
(1)	Horizontal size of the graphics data		Horizontal size of the graphics data. Set the size using the graphics display module.
(2)	Vertical size of the graphics data		Vertical size of the graphics data. Set the size using the graphics display module.
(3)	CMPSPX (horizontal start position)	DOCMSPXR	Set the horizontal distance from the upper left origin of the graphics data to the rectangular area for which the CRC code is calculated in pixel units.
(4)	CMPSPY (vertical start position)	DOCMSPYR	Set the vertical distance from the upper left origin of the graphics data to the rectangular area for which the CRC code is calculated in line units.
(5)	CMPSZX (horizontal size)	DOCMSZXR	Set the horizontal size of the rectangular area for which the CRC code is calculated in pixel units. The value should be set as follows: Horizontal size of the graphics data ≥ CMPSPX + CMPSZX.
(6)	CMPSZY (vertical size)	DOCMSZYR	Set the vertical size of the rectangular area for which the CRC code is calculated in line units. The value should be set as follows: Vertical size of the graphics data ≥ CMPSPY + CMPSZY.

41.5.6 CRC Calculation Time Period and Comparison Timing

Figure 41.5 shows the CRC calculation time period and timing of comparing the calculated result with the expected value.

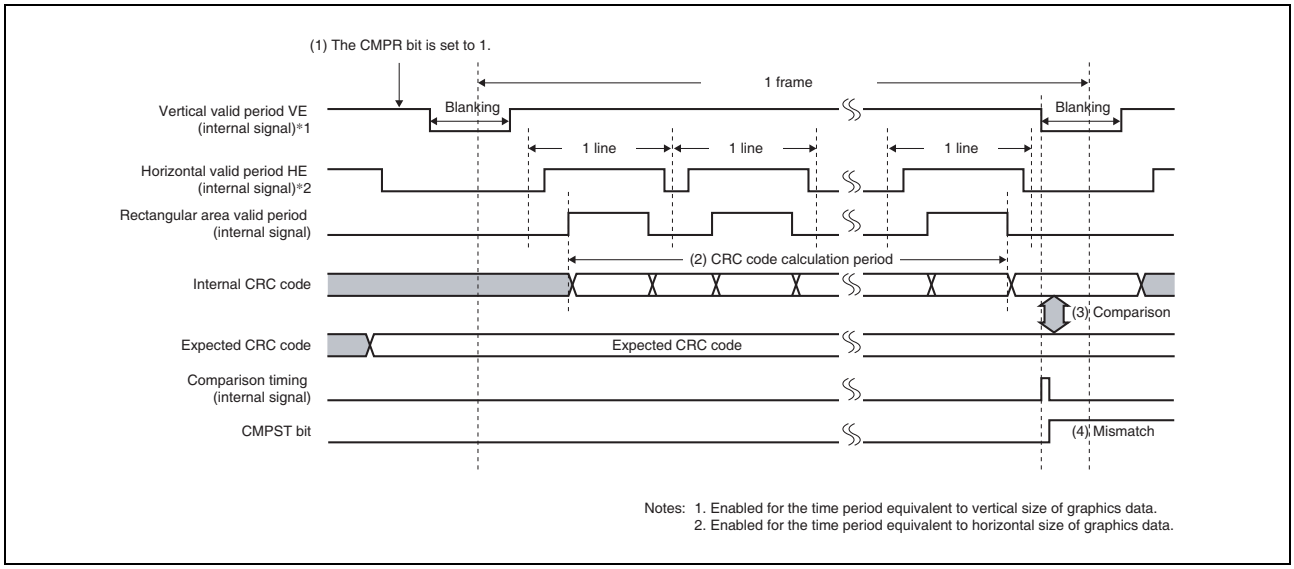


Figure 41.5 CRC Calculation Time Period and Timing of Comparing the Calculated Result with the Expected Value

[Operation]

(1) The operation starts at the next frame after the CMPR bit in DOCMCR is set to 1. For the register update timing, see Section 41.5.7, Register Update Timing.

(2) CRC code is calculated in the set rectangular area.

(3) The CRC code calculation result is compared with the expected CRC code value (DOCMECRCR) at the end of the valid period of the graphics data.

(4) If the compared CRC codes do not match, the CMPST bit in DOCMSTR is set.

41.5.7 Register Update Timing

41.5.7.1 Timing when Register Values are Loaded Inside

All the register bits except the CMPR bit in DOCMCR are loaded inside immediately after the registers are written to. Thus, if the register is updated with the CMPRU bit in DOCMCR as 1, an unexpected result is likely to be obtained in the CRC code calculation. The registers that affect the CRC code calculation (i.e., registers other than DOCMCLSTR and DOCMIENR) should be updated after confirming that the CMPRU bit in DOCMCR is 0.

The CMPR bit in DOCMCR is loaded inside upon detection of the start of the valid period of the graphics data. Thus, even if the register is rewritten to in the middle of a frame, the CRC code calculation of the frame does not get affected.

41.5.7.2 Timing when Internal State is Reflected in Registers

The internal state is reflected in DOCMSTR and DOCMCCRCR at the end of the valid period of the graphics data. The internal state of the CMPR bit is reflected in the CMPRU bit in DOCMCR at the start of the valid period of the graphics data.

The figure below shows the register update timing.

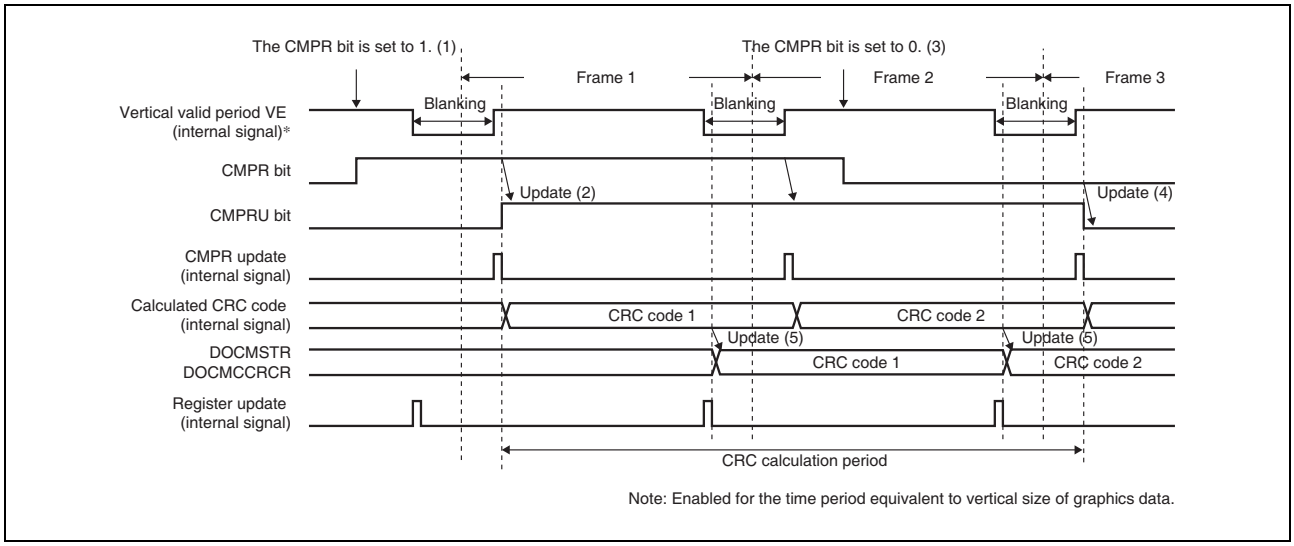


Figure 41.6 Register Update Timing

[Operation]

(1) The CMPR bit is set to 1. This bit is not immediately loaded inside.

(2) CRC calculation is carried out after the CMPR bit value is loaded inside upon detection of the start of the valid period of the graphics data.

(3) To suspend the CRC comparison, the CMPR bit is set to 0. Similarly to (1), this bit is not immediately loaded inside.

(4) The CRC calculation is suspended after the CMPR bit value is loaded inside upon detection of the start of the valid period of the graphics data.

(5) The internal state is reflected in the register at the end of the valid period of the graphics data.

41.5.8 Operation Flow

41.5.8.1 Procedure for Starting Display Out Comparison

The following figure shows a sample procedure for starting the display out comparison.

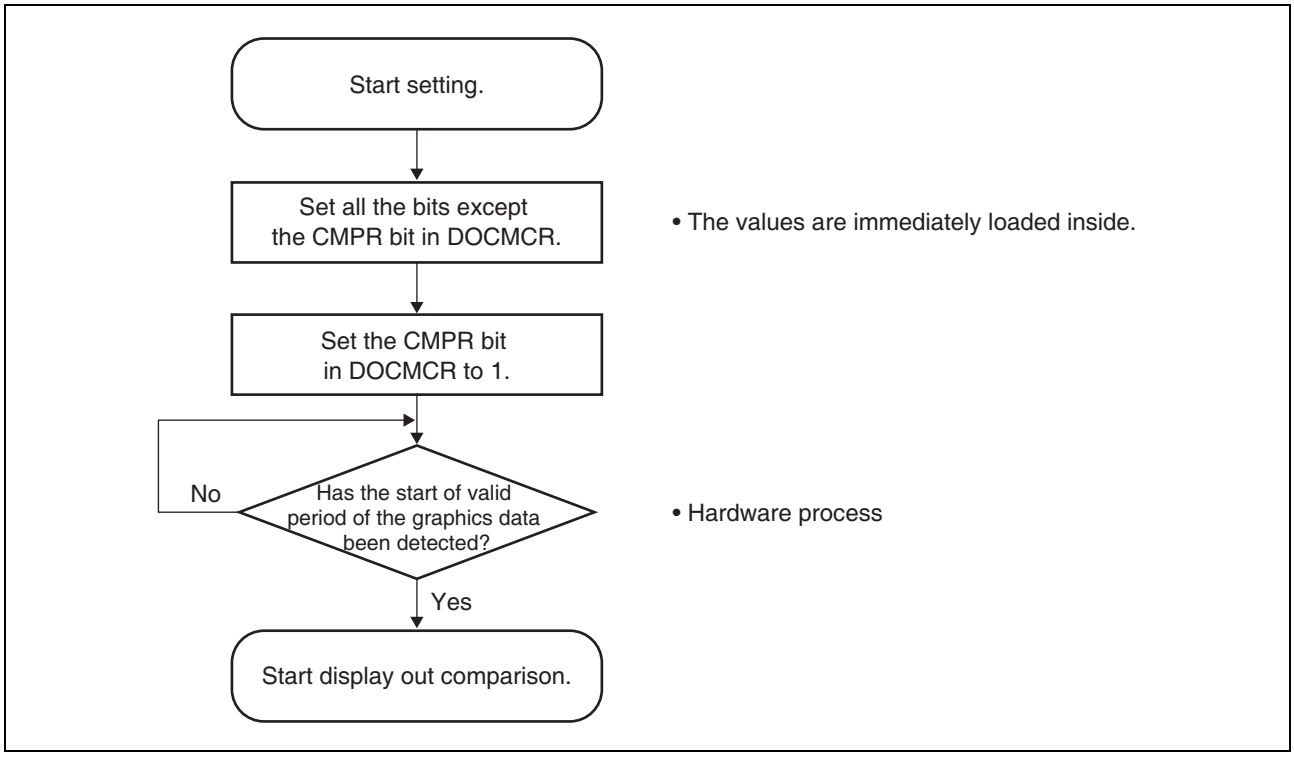


Figure 41.7 Sample Procedure for Starting the Display Out Comparison

41.5.8.2 Procedure for Changing Register Setting with stopping display output comparison

The following figure shows a sample procedure for changing the register setting during display out comparison.

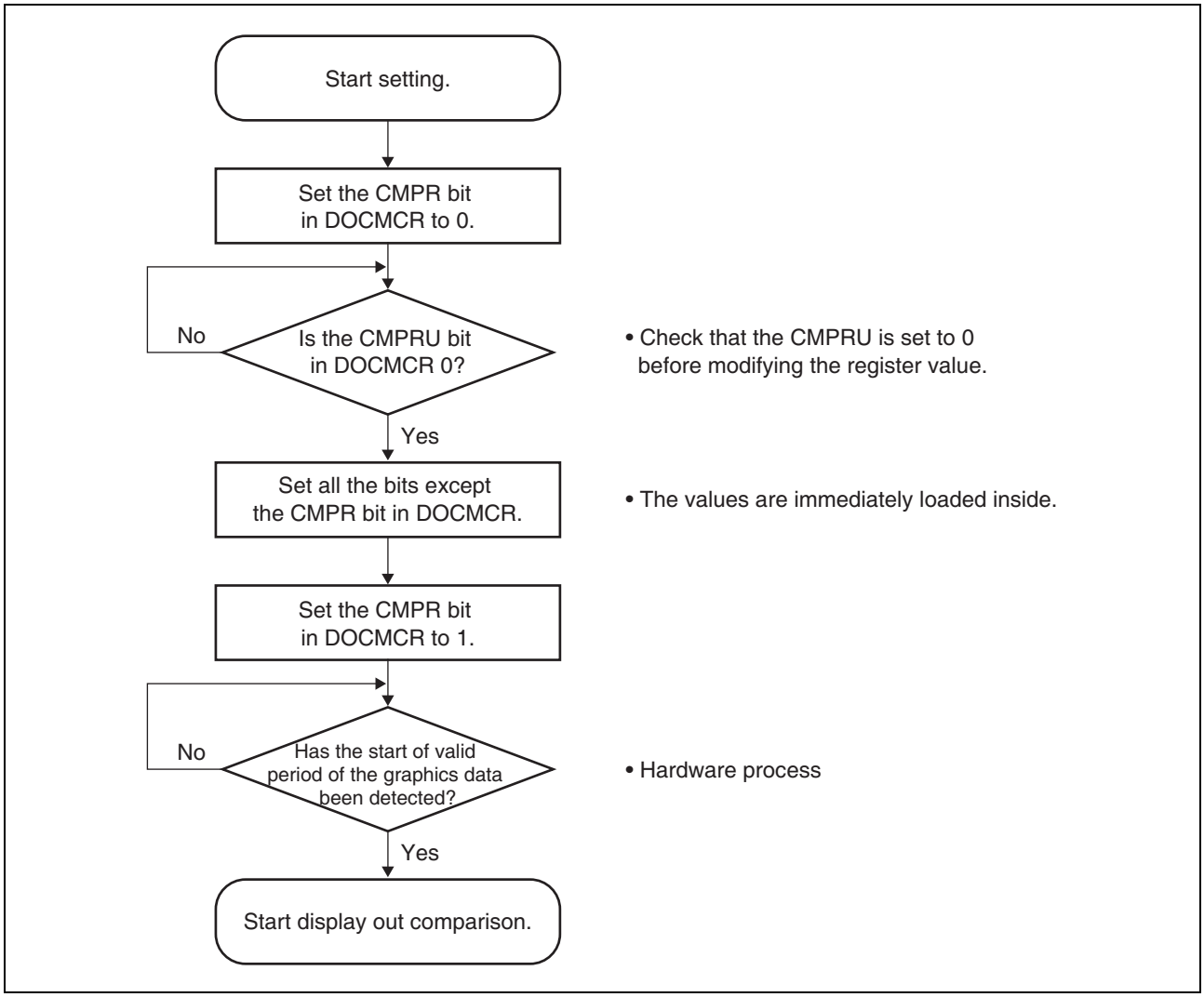


Figure 41.8 Sample Procedure for Changing the Register Setting with stopping Display Out Comparison

41.5.8.3 Procedure for Changing Register Setting without stopping display output comparison

Since the register settings for the next CRC calculation must be valid at the end of the vertical blanking period, it is also possible to change all registers within the vertical blanking period without stopping the CRC calculation.

That means it's not necessary to set the CMPR bit of the DOCMCR register to 0 and to wait until the CMPRU bit of the DOCMCR register has become 0, provided it is guaranteed that all register modifications are completed before the end of the vertical blanking period.

The start of the vertical blanking period is indicated by the interrupt INTVDCEnGR3VBLANK of the respective video channel n (n = 0, 1).

The following diagram outlines the procedure of a register update without stopping the display output comparison.

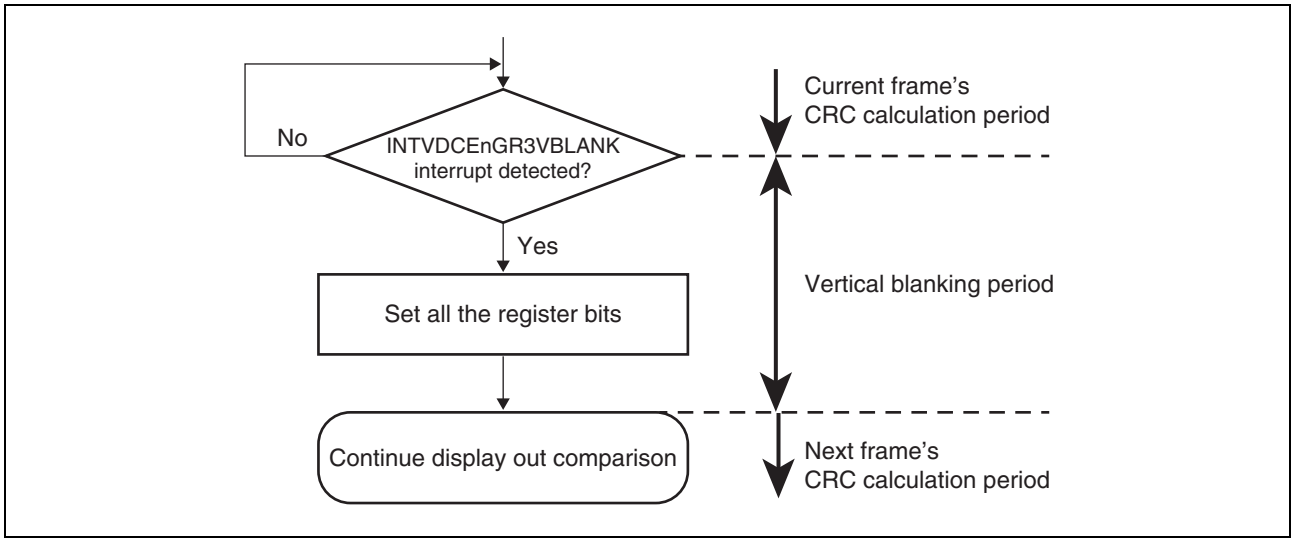


Figure 41.9 Sample Procedure for Changing the Register Setting without stopping Display Out Comparison

41.6 Interrupt

The display out comparison mismatch detection interrupt is provided as an interrupt source.

The interrupt request is generated when the CMPIEN bit in DOCMIENR and the CMPST bit in DOCMSTR are both set to 1.

41.7 Usage Note

41.7.1 Expected CRC Value

The graphics data after α blending should be selected at the time of debugging and the calculated CRC code value read from DOCMCCRCR should be used as the expected CRC code value. This is because there is a possibility of mismatching of the result of superimposition of the graphics data by the software and the result of superimposition by the graphics display module due to an error in calculation when the graphics data after α blending is selected.

41.7.2 Expansion Control Functionality

When scaling settings are applied for expansion processing by the Video Display Controller, this module becomes incapable of generating CRC codes. Stop this module while expansion processing is being applied.

Section 42 2D Graphics Processing Unit (GPU2D)

This section describes briefly the features of the 2D Graphics Processing Unit and contains a list of its registers.

42.1 Overview of the RH850/D1L/D1M 2D Graphics Processing Unit

42.1.1 Units

This microcontroller has the following number of units of the 2D Graphics Processing Unit.

Table 42.1 Units

2D Graphics Processing Unit (GPU2D)	D1L1	D1L2(H)	D1M1(H) D1M1-V2 D1M1A	D1M2(H)
Units	0	0	1	1
Names	–	–	GPU2D0	GPU2D0

42.1.2 Register addresses

All 2D Graphics Processing Unit register addresses are given as address offsets from the individual base addresses <GPU2Dn_base>.
The <GPU2Dn_base> addresses of each GPU2Dn are listed in the following table:

Table 42.2 Register base addresses <GPU2Dn_base>

GPU2Dn unit	<GPU2Dn_base> address
GPU2D0	FFFD 0000_H

42.1.3 Clock supply

All 2D Graphics Processing Units provide two clock inputs.

Table 42.3 Clock supply

GPU2Dn unit	GPU2Dn clock	Connected to
GPU2D0	PBUS clock	Clock Controller C_ISO_PCLK
	GPU2D operation clock	• D1M1(H), D1M1-V2, D1M1A: Clock Controller C_ISO_XCCLK • D1M2(H): Clock Controller C_ISO_CPUCLK

42.1.4 Interrupts

The 2D Graphics Processing Units can generate the following interrupt requests:

Table 42.4 GPU2Dn interrupt requests

GPU2D0 signals	Function	Connected to
pause_irq_o	Pause interrupt	Interrupt Controller INTGPU2D0PAUSE
sync_irq_o	SYNC interrupt	Interrupt Controller INTGPU2D0SYNC
special_irq_o	Stop / Stall / MBI Error interrupt	Interrupt Controller INTGPU2D0SP

42.1.5 Reset sources

The 2D Graphics Processing Units and their registers are initialized by the following reset signal:

Table 42.5 Reset sources

GPU2Dn unit	Reset signal
GPU2D0	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

42.1.6 Bus master ID

The 2D Graphics Processing Unit bus master interface is connected to the XC1cross-connect system. The master interface has the following master ID:

GPU2D: MSTID4

42.2 Functional Overview

GPU2D is a 2D render core targeting high resolution displays. The most important aspect to be considered for such displays is the necessary pixel fill rate to provide high rendering frame rates even when writing multiple MPixels per frame.

Compared to 2D Drawing Engine (DRW) from Dx4 dashboard controller series, the primary differences are:

- Completely newly developed low-overhead enumeration
- Bézier lines and Bézier segments
- Multi-Texturing (two textures)
- Flexible colorization using multiple passes (three colorization passes)
- Prefetching of textures to hide texture read latency
- Texture address swizzling and tiling for higher texturing performance
- Dedicated clear unit for fill of memory regions with maximum bandwidth
- completely overworked display list handling and device context switch support

The GPU2D driver targets to provide a slim, small-footprint interface to the hardware features while hiding the complexity of efficiently using the hardware module from the user.

- GPU2D provides high quality 2D rendering
  - general sub pixel accuracy
  - direct edge anti-aliasing support

Compatibility

- Software compatible to the 2D Drawing Engine (DRW) of the Dx4 series, but delivers about four times performance
- GPU2D supports OpenVG
  - direct Bézier curve rendering
  - multi-texturing to support images and gradients
  - support for an unlimited number of scissor rectangles
  - extended blending modes
  - global alpha mask

BitBLT engine

- GPU2D supports the typical BitBLT functions
  - Direct blit and stretch blit
  - Transparency and color keying
  - Bilinear filtering
  - Per pixel alpha blending
  - Coloring
  - RLE decompression
  - RLE decompression formats: 8 bpp, 16 bpp, 24 bpp, 32 bpp
- Texture mapping
  - Perspective 3D texturing, supporting cover flow effects
  - Multi-texturing for sophisticated shading effects
  - Dot3 bump mapping allows 3D per-pixel lighting effects
  - Very suitable for dashboard needles

Direct RLE texture input

- Textures can be RLE compressed
  - Targa tga file format
  - Restart from any texture position (fast forward)
    RLE textures shall be mapped completely from texture start for best performance
- Input color formats
  - αRGB8888, RGBα8888, αRGB4444, RGBα4444, αRGB1555, RGBα5551, RGB565
  - Color Look-Up (CLUT) or luminance formats αL88, αL44, αL8, αL4, αL2, αL1
- Typical size compression of factor 4 possible
  - Fitting smaller color format = factor 2, RLE = factor 2
  - Most effective for larger monochrome parts as in GUI elements
  - GPU2D cannot generate RLE textures on the fly, but they are stored pre-rendered in texture memory

Various texturing options

- Perspective correct texturing for 3D effects
- Multi texturing for illumination effects
- Bump mapping for surface modelling
- Freely programmable 7x7 filter matrix, e.g. can be used to calculate blurring effects on the fly

Vector drawing engine

- OpenVG compatible
- Lines
  - Arbitrary width
  - Round endpoints
  - Truncated endpoints
  - Alpha gradients
  - Soft edges (blurring)
- Polygons
  - Triangles and Quadrangles
  - Alpha gradients
  - Soft edges (blurring)
  - Per edge controls for anti aliasing

Direct Bézier Rendering

- Bézier curves describe smooth outlines
  - No sharp corners
  - Few curve control points
- Compatible with Truetype
- Very effective for OpenVG support

General features

- Command Stream execution from ring-buffer
  - No stream size limit due to memory availability
  - Reduced latency from API call to displayed frame
    - Start frame rendering already with first object instead after last
    - Faster interaction through HMI
- Drawing command lists support
  - Checksums in drawing command stream for safety
  - Automatically stop on error in command stream and interrupt generation towards the CPU
  - CPU can recover the error and drawing engine can continue without artefacts
- Performance Verification
  - Performance Counters
  - Visualize enumeration efficiency
  - Visualize cache burst access lengths
- Readout or active dumping of full internal state
  - All registers, call stack, CLUT content
- Forced stop of execution

– Fast and efficient task switch
– Let core dump internal state for later resume

42.3 GPU2D registers overview

This section contains a list of all registers of the 2D Graphics Processing Unit.

42.3.1 Register lists by function

The 2D Graphics Processing Unit is controlled and operated by the following registers:

Table 42.6 STC registers

Register name	Shortcut	Address
STC version	GPU2D0STC_VERSION	<GPU2Dn_base> + 000_H
STC configuration 1	GPU2D0STC_CONFIG_1	<GPU2Dn_base> + 004_H
STC configuration 2	GPU2D0STC_CONFIG_2	<GPU2Dn_base> + 008_H
STC configuration 3	GPU2D0STC_CONFIG_3	<GPU2Dn_base> + $00C_H$
Modules busy bits	GPU2D0STC_BUSY_STATUS	<GPU2Dn_base> + 010_H
STC control	GPU2D0STC_CONTROL	<GPU2Dn_base> + 014_H
STC interrupt status/clear	GPU2D0STC_INTERRUPT_STATUS_CLEAR	<GPU2Dn_base> + 018_H
Stream start address	GPU2D0STC_STREAM_ADDRESS	<GPU2Dn_base> + $01C_H$
Ring pause address	GPU2D0STC_RING_PAUSE_ADDRESS	<GPU2Dn_base> + 020_H
Current ring address	GPU2D0STC_CURRENT_RING_ADDRESS	<GPU2Dn_base> + 024_H
Current stream address	GPU2D0STC_CURRENT_STREAM_ADDRESS	<GPU2Dn_base> + 028_H
Checksum	GPU2D0STC_CHECKSUM	<GPU2Dn_base> + $02C_H$
Synchronization ID	GPU2D0STC_SYNCHRONIZATION_ID_0	<GPU2Dn_base> + 030_H
Synchronization ID	GPU2D0STC_SYNCHRONIZATION_ID_1	<GPU2Dn_base> + 034_H
Synchronization ID	GPU2D0STC_SYNCHRONIZATION_ID_2	<GPU2Dn_base> + 038_H
Call Stack Pointer	GPU2D0STC_CALL_STACK_POINTER	<GPU2Dn_base> + $03C_H$
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_0	<GPU2Dn_base> + 040_H
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_1	<GPU2Dn_base> + 044_H
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_2	<GPU2Dn_base> + 048_H
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_3	<GPU2Dn_base> + $04C_H$
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_4	<GPU2Dn_base> + 050_H
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_5	<GPU2Dn_base> + 054_H
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_6	<GPU2Dn_base> + 058_H
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_7	<GPU2Dn_base> + $05C_H$
Performance Counter Values	GPU2D0STC_PERFORMANCE_COUNTER_VALUES_0	<GPU2Dn_base> + 080_H
Performance Counter Values	GPU2D0STC_PERFORMANCE_COUNTER_VALUES_1	<GPU2Dn_base> + 084_H
Performance Counter Values	GPU2D0STC_PERFORMANCE_COUNTER_VALUES_2	<GPU2Dn_base> + 088_H
Performance Counter Values	GPU2D0STC_PERFORMANCE_COUNTER_VALUES_3	<GPU2Dn_base> + $08C_H$

Table 42.7 PSU Pixel Selection registers

Register name	Shortcut	Address
Bounding box upper left corner	GPU2D0PSU_LIM_BBOX_MIN	<GPU2Dn_base> + 200_H
Bounding box lower right corner	GPU2D0PSU_LIM_BBOX_MAX	<GPU2Dn_base> + 204_H
Start position for enumeration	GPU2D0PSU_LIM_START	<GPU2Dn_base> + 208_H
Pixel selection unit Limiter control register	GPU2D0PSU_LIM_CTRL	<GPU2Dn_base> + $20C_H$
Enumeration stripes width and offset	GPU2D0PSU_LIM_STRIPE	<GPU2Dn_base> + 210_H
Bézier control	GPU2D0PSU_BEZ_CTRL	<GPU2Dn_base> + 214_H
Offset to Limiter value for Bézier function (14.18 signed fixed point)	GPU2D0PSU_BEZ_VOFF_0	<GPU2Dn_base> + 218_H
Offset to Limiter value for Bézier function (14.18 signed fixed point)	GPU2D0PSU_BEZ_VOFF_1	<GPU2Dn_base> + $21C_H$
Bézier antialiasing and width control	GPU2D0PSU_BEZ_AA_CTRL	<GPU2Dn_base> + 220_H
Initial Limiter value of enumeration start position (14.18 signed fixed point)	GPU2D0PSU_LIM_VSTART_0	<GPU2Dn_base> + 224_H
Fractional part of Limiter x step (2.18 signed fixed point)	GPU2D0PSU_LIM_DX_0	<GPU2Dn_base> + 228_H
Fractional part of Limiter y step (2.18 signed fixed point)	GPU2D0PSU_LIM_DY_0	<GPU2Dn_base> + $22C_H$
Initial Limiter value of enumeration start position (14.18 signed fixed point)	GPU2D0PSU_LIM_VSTART_1	<GPU2Dn_base> + 230_H
Fractional part of Limiter x step (2.18 signed fixed point)	GPU2D0PSU_LIM_DX_1	<GPU2Dn_base> + 234_H
Fractional part of Limiter y step (2.18 signed fixed point)	GPU2D0PSU_LIM_DY_1	<GPU2Dn_base> + 238_H
Initial Limiter value of enumeration start position (14.18 signed fixed point)	GPU2D0PSU_LIM_VSTART_2	<GPU2Dn_base> + $23C_H$
Fractional part of Limiter x step (2.18 signed fixed point)	GPU2D0PSU_LIM_DX_2	<GPU2Dn_base> + 240_H
Fractional part of Limiter y step (2.18 signed fixed point)	GPU2D0PSU_LIM_DY_2	<GPU2Dn_base> + 244_H
Initial Limiter value of enumeration start position (14.18 signed fixed point)	GPU2D0PSU_LIM_VSTART_3	<GPU2Dn_base> + 248_H
Fractional part of Limiter x step (2.18 signed fixed point)	GPU2D0PSU_LIM_DX_3	<GPU2Dn_base> + $24C_H$
Fractional part of Limiter y step (2.18 signed fixed point)	GPU2D0PSU_LIM_DY_3	<GPU2Dn_base> + 250_H
Initial Limiter value of enumeration start position (14.18 signed fixed point)	GPU2D0PSU_LIM_VSTART_4	<GPU2Dn_base> + 254_H
Fractional part of Limiter x step (2.18 signed fixed point)	GPU2D0PSU_LIM_DX_4	<GPU2Dn_base> + 258_H
Fractional part of Limiter y step (2.18 signed fixed point)	GPU2D0PSU_LIM_DY_4	<GPU2Dn_base> + $25C_H$
Initial Limiter value of enumeration start position (14.18 signed fixed point)	GPU2D0PSU_LIM_VSTART_5	<GPU2Dn_base> + 260_H
Fractional part of Limiter x step (2.18 signed fixed point)	GPU2D0PSU_LIM_DX_5	<GPU2Dn_base> + 264_H
Fractional part of Limiter y step (2.18 signed fixed point)	GPU2D0PSU_LIM_DY_5	<GPU2Dn_base> + 268_H
Debug control	GPU2D0PSU_DEBUG_CTRL	<GPU2Dn_base> + $26C_H$
Maximum coordinates generated by PSU, clipping limits for bounding box	GPU2D0PSU_LIM_MAX_COORD	<GPU2Dn_base> + 270_H

Table 42.8 Texture U/V Interpolation registers

Register name	Shortcut	Address
Start value for texture U coordinate	GPU2D0TXA_U_OFFSET_0	<GPU2Dn_base> + 300_H
Start value for texture V coordinate	GPU2D0TXA_V_OFFSET_0	<GPU2Dn_base> + 304_H
Increment of texture U coordinate for a pixel step in X direction	GPU2D0TXA_DUX_0	<GPU2Dn_base> + 308_H
Increment of texture U coordinate for a pixel step in Y direction	GPU2D0TXA_DUY_0	<GPU2Dn_base> + $30C_H$
Increment of texture V coordinate for a pixel step in X direction	GPU2D0TXA_DVX_0	<GPU2Dn_base> + 310_H
Increment of texture V coordinate for a pixel step in Y direction	GPU2D0TXA_DVY_0	<GPU2Dn_base> + 314_H
Start value for texture U coordinate.	GPU2D0TXA_U_OFFSET_1	<GPU2Dn_base> + 318_H
Start value for texture V coordinate.	GPU2D0TXA_V_OFFSET_1	<GPU2Dn_base> + $31C_H$
Increment of texture U coordinate for a pixel step in X direction	GPU2D0TXA_DUX_1	<GPU2Dn_base> + 320_H
Increment of texture U coordinate for a pixel step in Y direction	GPU2D0TXA_DUY_1	<GPU2Dn_base> + 324_H
Increment of texture V coordinate for a pixel step in X direction	GPU2D0TXA_DVX_1	<GPU2Dn_base> + 328_H
Increment of texture V coordinate for a pixel step in Y direction	GPU2D0TXA_DVY_1	<GPU2Dn_base> + $32C_H$

Table 42.9 ZSS, Texture RHW and Z Interpolation registers

Register name	Shortcut	Address
ZSA cache scheduler pitch	GPU2D0ZSS_PITCH	<GPU2Dn_base> + 400_H
ZSA cache scheduler span config	GPU2D0ZSS_SPAN_CONFIG	<GPU2Dn_base> + 404_H
Start value for Z coordinate	GPU2D0TXA_Z_OFFSET	<GPU2Dn_base> + 408_H
Increment of Z coordinate for a pixel step in X direction	GPU2D0TXA_DZX	<GPU2Dn_base> + $40C_H$
Increment of Z coordinate for a pixel step in Y direction	GPU2D0TXA_DZY	<GPU2Dn_base> + 410_H
Start value for RHW attribute (1/w) of Z	GPU2D0TXA_Z_RHW_OFFSET	<GPU2Dn_base> + 414_H
Increment of RHW attribute of Z for a pixel step in X direction	GPU2D0TXA_Z_DRHWX	<GPU2Dn_base> + 418_H
Increment of RHW attribute of Z for a pixel step in Y direction	GPU2D0TXA_Z_DRHWY	<GPU2Dn_base> + $41C_H$
Start value for RHW attribute (1/w)	GPU2D0TXA_RHW_OFFSET_0	<GPU2Dn_base> + 420_H
Increment of RHW attribute for a pixel step in X direction	GPU2D0TXA_DRHWX_0	<GPU2Dn_base> + 424_H
Increment of RHW attribute for a pixel step in Y direction	GPU2D0TXA_DRHWY_0	<GPU2Dn_base> + 428_H
Start value for RHW attribute (1/w)	GPU2D0TXA_RHW_OFFSET_1	<GPU2Dn_base> + $42C_H$
Increment of RHW attribute for a pixel step in X direction	GPU2D0TXA_DRHWX_1	<GPU2Dn_base> + 430_H
Increment of RHW attribute for a pixel step in Y direction	GPU2D0TXA_DRHWY_1	<GPU2Dn_base> + 434_H

Table 42.10 ZSA Pipeline + TXA registers

Register name	Shortcut	Address
ZSA control register	GPU2D0ZSA_CTRL	<GPU2Dn_base> + 600_H
ZSA buffer start address	GPU2D0ZSC_START_ADDRESS	<GPU2Dn_base> + 604_H
ZSA Unit Control register 1	GPU2D0ZSU_CTRL1	<GPU2Dn_base> + 608_H
ZSA Unit Control register 2	GPU2D0ZSU_CTRL2	<GPU2Dn_base> + $60C_H$
Texture size	GPU2D0TXA_SIZE_0	<GPU2Dn_base> + 610_H
Filter shape, size and texture pitch	GPU2D0TXA_ACCESS_0	<GPU2Dn_base> + 614_H
Texture size	GPU2D0TXA_SIZE_1	<GPU2Dn_base> + 618_H
Filter shape, size and texture pitch	GPU2D0TXA_ACCESS_1	<GPU2Dn_base> + $61C_H$

Table 42.11 TEX/COL (Global), FBD/FBS and RLD registers

Register name	Shortcut	Address
Global texture unit register	GPU2D0TEX_GLOBAL	<GPU2Dn_base> + 800_H
Global color unit register	GPU2D0COL_GLOBAL	<GPU2Dn_base> + 804_H
Framebuffer read decision register	GPU2D0FBD_CONFIG	<GPU2Dn_base> + 808_H
Framebuffer cache scheduler pitch and debug control	GPU2D0FBS_PITCH	<GPU2Dn_base> + $80C_H$
Framebuffer cache scheduler span config	GPU2D0FBS_SPAN_CONFIG	<GPU2Dn_base> + 810_H
Start address of RLE code of the texture in memory	GPU2D0RLD_START_ADDRESS	<GPU2Dn_base> + 814_H
Constant color register	GPU2D0COL_CONST_COLOR_0	<GPU2Dn_base> + 818_H
Constant color register	GPU2D0COL_CONST_COLOR_1	<GPU2Dn_base> + $81C_H$
Constant color register	GPU2D0COL_CONST_COLOR_2	<GPU2Dn_base> + 820_H
Constant color register	GPU2D0COL_CONST_COLOR_3	<GPU2Dn_base> + 824_H

Table 42.12 TEX+TXP registers

Register name	Shortcut	Address
Texture mode	GPU2D0TEX_MODE_0	<GPU2Dn_base> + $A00_H$
Start address of top-left texel of the texture in memory	GPU2D0TXC_START_ADDRESS_0	<GPU2Dn_base> + $A04_H$
TXP control	GPU2D0TXP_CTRL_0	<GPU2Dn_base> + $A08_H$
Offset for all CLUT lookups of the texture unit	GPU2D0TXP_CLUT_OFFSET_0	<GPU2Dn_base> + $A0C_H$
Color key comparison value	GPU2D0TXP_COLOR_KEY_0	<GPU2Dn_base> + $A10_H$
Texel fill color for wrap mode FILL	GPU2D0TXP_FILL_COLOR_0	<GPU2Dn_base> + $A14_H$
Scale and bias for filter in TXP	GPU2D0TXP_FILTER_SCALE_BIAS_0	<GPU2Dn_base> + $A18_H$
Texture mode	GPU2D0TEX_MODE_1	<GPU2Dn_base> + $A1C_H$
Start address of top-left texel of the texture in memory	GPU2D0TXC_START_ADDRESS_1	<GPU2Dn_base> + $A20_H$
TXP control	GPU2D0TXP_CTRL_1	<GPU2Dn_base> + $A24_H$
Offset for all CLUT lookups of the texture unit	GPU2D0TXP_CLUT_OFFSET_1	<GPU2Dn_base> + $A28_H$
Color key comparison value	GPU2D0TXP_COLOR_KEY_1	<GPU2Dn_base> + $A2C_H$
Texel fill color for wrap mode FILL	GPU2D0TXP_FILL_COLOR_1	<GPU2Dn_base> + $A30_H$
Scale and bias for filter in TXP	GPU2D0TXP_FILTER_SCALE_BIAS_1	<GPU2Dn_base> + $A34_H$

Table 42.13 COL Pass registers

Register name	Shortcut	Address
Control structure for Op1a (Alpha and RGB path)	GPU2D0COL_ARGB_OP1A_0	<GPU2Dn_base> + $B00_H$
Control structure for Op1b (Alpha and RGB path)	GPU2D0COL_ARGB_OP1B_0	<GPU2Dn_base> + $B04_H$
Control structure for Op2a (Alpha and RGB path)	GPU2D0COL_ARGB_OP2A_0	<GPU2Dn_base> + $B08_H$
Control structure for Op2b (Alpha and RGB path)	GPU2D0COL_ARGB_OP2B_0	<GPU2Dn_base> + $B0C_H$
Control structure for Op3 (Alpha and RGB path)	GPU2D0COL_ARGB_OP3_0	<GPU2Dn_base> + $B10_H$
Control structure for core output (Alpha and RGB path)	GPU2D0COL_ARGB_OUT_0	<GPU2Dn_base> + $B14_H$
Control structure for Op1a (Alpha and RGB path)	GPU2D0COL_ARGB_OP1A_1	<GPU2Dn_base> + $B18_H$
Control structure for Op1b (Alpha and RGB path)	GPU2D0COL_ARGB_OP1B_1	<GPU2Dn_base> + $B1C_H$
Control structure for Op2a (Alpha and RGB path)	GPU2D0COL_ARGB_OP2A_1	<GPU2Dn_base> + $B20_H$
Control structure for Op2b (Alpha and RGB path)	GPU2D0COL_ARGB_OP2B_1	<GPU2Dn_base> + $B24_H$
Control structure for Op3 (Alpha and RGB path)	GPU2D0COL_ARGB_OP3_1	<GPU2Dn_base> + $B28_H$
Control structure for core output (Alpha and RGB path)	GPU2D0COL_ARGB_OUT_1	<GPU2Dn_base> + $B2C_H$
Control structure for Op1a (Alpha and RGB path)	GPU2D0COL_ARGB_OP1A_2	<GPU2Dn_base> + $B30_H$
Control structure for Op1b (Alpha and RGB path)	GPU2D0COL_ARGB_OP1B_2	<GPU2Dn_base> + $B34_H$
Control structure for Op2a (Alpha and RGB path)	GPU2D0COL_ARGB_OP2A_2	<GPU2Dn_base> + $B38_H$
Control structure for Op2b (Alpha and RGB path)	GPU2D0COL_ARGB_OP2B_2	<GPU2Dn_base> + $B3C_H$
Control structure for Op3 (Alpha and RGB path)	GPU2D0COL_ARGB_OP3_2	<GPU2Dn_base> + $B40_H$
Control structure for core output (Alpha and RGB path)	GPU2D0COL_ARGB_OUT_2	<GPU2Dn_base> + $B44_H$

Table 42.14 Blending Pipeline registers

Register name	Shortcut	Address
Framebuffer pixel organization/format	GPU2D0FB_PIXEL_ORG	<GPU2Dn_base> + $C00_H$
Framebuffer start address	GPU2D0FBC_START_ADDRESS	<GPU2Dn_base> + $C04_H$
Blend unit factor and mode	GPU2D0BLU_BLEND	<GPU2Dn_base> + $C08_H$
Blend unit dither control	GPU2D0BLU_DITHER	<GPU2Dn_base> + $C0C_H$
Blend unit write control	GPU2D0BLU_WRITE	<GPU2Dn_base> + $C10_H$

Table 42.15 PFC registers

Register name	Shortcut	Address
Performance counter enable	GPU2D0PFC_ENABLE	<GPU2Dn_base> + $E00_H$
Performance counter clear	GPU2D0PFC_CLEAR	<GPU2Dn_base> + $E04_H$
Performance counter event selection	GPU2D0PFC_EVENT_SELECT_0	<GPU2Dn_base> + $E08_H$
Performance counter event selection	GPU2D0PFC_EVENT_SELECT_1	<GPU2Dn_base> + $E0C_H$
Performance counter event selection	GPU2D0PFC_EVENT_SELECT_2	<GPU2Dn_base> + $E10_H$
Performance counter event selection	GPU2D0PFC_EVENT_SELECT_3	<GPU2Dn_base> + $E14_H$

Table 42.16 CLR registers

Register name	Shortcut	Address
Clear unit color value	GPU2D0CLR_VALUE	<GPU2Dn_base> + $F00_H$
Clear unit line config	GPU2D0CLR_LINE_CONFIG	<GPU2Dn_base> + $F04_H$
Clear unit number of lines	GPU2D0CLR_CTRL	<GPU2Dn_base> + $F08_H$
Framebuffer start address for clear unit	GPU2D0CLR_START_ADDRESS	<GPU2Dn_base> + $F0C_H$

<GPU2Dn_base>

The base addresses <GPU2Dn_base> of the GPU2Dn is defined in the first section of this chapter under the key word “Register addresses”.

42.3.2 Register list by address

Table 42.17 Register list by address (1/5)

Register name	Shortcut	R/W	Address	Access Size
STC version	GPU2D0STC_VERSION	R	FFFD0000	32
STC configuration 1	GPU2D0STC_CONFIG_1	R	FFFD0004	32
STC configuration 2.	GPU2D0STC_CONFIG_2	R	FFFD0008	32
STC configuration 3.	GPU2D0STC_CONFIG_3	R	FFFD000C	32
Modules busy bits	GPU2D0STC_BUSY_STATUS	R	FFFD0010	32
STC control	GPU2D0STC_CONTROL	R/W	FFFD0014	32
STC interrupt status/clear	GPU2D0STC_INTERRUPT_STATUS_CLEAR	R/W	FFFD0018	32
Stream start address	GPU2D0STC_STREAM_ADDRESS	R/W	FFFD001C	32
Ring pause address	GPU2D0STC_RING_PAUSE_ADDRESS	R/W	FFFD0020	32
Current ring address	GPU2D0STC_CURRENT_RING_ADDRESS	R	FFFD0024	32
Current stream address	GPU2D0STC_CURRENT_STREAM_ADDRESS	R	FFFD0028	32
Checksum	GPU2D0STC_CHECKSUM	R/W	FFFD002C	32
Synchronization ID	GPU2D0STC_SYNCHRONIZATION_ID_0	R/W	FFFD0030	32
Synchronization ID	GPU2D0STC_SYNCHRONIZATION_ID_1	R/W	FFFD0034	32
Synchronization ID	GPU2D0STC_SYNCHRONIZATION_ID_2	R/W	FFFD0038	32
Call Stack Pointer	GPU2D0STC_CALL_STACK_POINTER	R/W	FFFD003C	32
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_0	R/W	FFFD0040	32
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_1	R/W	FFFD0044	32
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_2	R/W	FFFD0048	32
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_3	R/W	FFFD004C	32
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_4	R/W	FFFD0050	32
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_5	R/W	FFFD0054	32
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_6	R/W	FFFD0058	32
Call Stack Entry	GPU2D0STC_CALL_STACK_ENTRY_7	R/W	FFFD005C	32
Performance Counter Values	GPU2D0STC_PERFORMANCE_COUNTER_VALUES_0	R	FFFD0080	32
Performance Counter Values	GPU2D0STC_PERFORMANCE_COUNTER_VALUES_1	R	FFFD0084	32
Performance Counter Values	GPU2D0STC_PERFORMANCE_COUNTER_VALUES_2	R	FFFD0088	32
Performance Counter Values	GPU2D0STC_PERFORMANCE_COUNTER_VALUES_3	R	FFFD008C	32
Bounding box upper left corner	GPU2D0PSU_LIM_BBOX_MIN	R/W	FFFD0200	32
Bounding box lower right corner	GPU2D0PSU_LIM_BBOX_MAX	R/W	FFFD0204	32
Start position for enumeration	GPU2D0PSU_LIM_START	R/W	FFFD0208	32
Pixel selection unit Limiter control registers	GPU2D0PSU_LIM_CTRL	R/W	FFFD020C	32
Enumeration stripes width and offset	GPU2D0PSU_LIM_STRIPE	R/W	FFFD0210	32
Bézier control	GPU2D0PSU_BEZ_CTRL	R/W	FFFD0214	32
Offset to Limiter value for Bézier function (14.18 signed fixed point)	GPU2D0PSU_BEZ_VOFF_0	R/W	FFFD0218	32
Offset to Limiter value for Bézier function (14.18 signed fixed point)	GPU2D0PSU_BEZ_VOFF_1	R/W	FFFD021C	32

Table 42.17 Register list by address (2/5)

Register name	Shortcut	R/W	Address	Access Size
Bézier antialiasing and width control	GPU2D0PSU_BEZ_AA_CTRL	R/W	FFFD0220	32
Initial Limiter value of enumeration start position (14.18 signed fixed point)	GPU2D0PSU_LIM_VSTART_0	R/W	FFFD0224	32
Fractional part of Limiter x step (2.18 signed fixed point)	GPU2D0PSU_LIM_DX_0	R/W	FFFD0228	32
Fractional part of Limiter y step (2.18 signed fixed point)	GPU2D0PSU_LIM_DY_0	R/W	FFFD022C	32
Initial Limiter value of enumeration start position (14.18 signed fixed point)	GPU2D0PSU_LIM_VSTART_1	R/W	FFFD0230	32
Fractional part of Limiter x step (2.18 signed fixed point)	GPU2D0PSU_LIM_DX_1	R/W	FFFD0234	32
Fractional part of Limiter y step (2.18 signed fixed point)	GPU2D0PSU_LIM_DY_1	R/W	FFFD0238	32
Initial Limiter value of enumeration start position (14.18 signed fixed point)	GPU2D0PSU_LIM_VSTART_2	R/W	FFFD023C	32
Fractional part of Limiter x step (2.18 signed fixed point)	GPU2D0PSU_LIM_DX_2	R/W	FFFD0240	32
Fractional part of Limiter y step (2.18 signed fixed point)	GPU2D0PSU_LIM_DY_2	R/W	FFFD0244	32
Initial Limiter value of enumeration start position (14.18 signed fixed point)	GPU2D0PSU_LIM_VSTART_3	R/W	FFFD0248	32
Fractional part of Limiter x step (2.18 signed fixed point)	GPU2D0PSU_LIM_DX_3	R/W	FFFD024C	32
Fractional part of Limiter y step (2.18 signed fixed point)	GPU2D0PSU_LIM_DY_3	R/W	FFFD0250	32
Initial Limiter value of enumeration start position (14.18 signed fixed point)	GPU2D0PSU_LIM_VSTART_4	R/W	FFFD0254	32
Fractional part of Limiter x step (2.18 signed fixed point)	GPU2D0PSU_LIM_DX_4	R/W	FFFD0258	32
Fractional part of Limiter y step (2.18 signed fixed point)	GPU2D0PSU_LIM_DY_4	R/W	FFFD025C	32
Initial Limiter value of enumeration start position (14.18 signed fixed point)	GPU2D0PSU_LIM_VSTART_5	R/W	FFFD0260	32
Fractional part of Limiter x step (2.18 signed fixed point)	GPU2D0PSU_LIM_DX_5	R/W	FFFD0264	32
Fractional part of Limiter y step (2.18 signed fixed point)	GPU2D0PSU_LIM_DY_5	R/W	FFFD0268	32
Debug control	GPU2D0PSU_DEBUG_CTRL	R/W	FFFD026C	32
Maximum coordinates generated by PSU, clipping limits for bounding box	GPU2D0PSU_LIM_MAX_COORD	R/W	FFFD0270	32
Start value for texture U coordinate.	GPU2D0TXA_U_OFFSET_0	R/W	FFFD0300	32
Start value for texture V coordinate.	GPU2D0TXA_V_OFFSET_0	R/W	FFFD0304	32
Increment of texture U coordinate for a pixel step in X direction	GPU2D0TXA_DUX_0	R/W	FFFD0308	32
Increment of texture U coordinate for a pixel step in Y direction	GPU2D0TXA_DUY_0	R/W	FFFD030C	32

Table 42.17 Register list by address (3/5)

Register name	Shortcut	R/W	Address	Access Size
Increment of texture V coordinate for a pixel step in X direction	GPU2D0TXA_DVX_0	R/W	FFFD0310	32
Increment of texture V coordinate for a pixel step in Y direction	GPU2D0TXA_DVY_0	R/W	FFFD0314	32
Start value for texture U coordinate.	GPU2D0TXA_U_OFFSET_1	R/W	FFFD0318	32
Start value for texture V coordinate.	GPU2D0TXA_V_OFFSET_1	R/W	FFFD031C	32
Increment of texture U coordinate for a pixel step in X direction	GPU2D0TXA_DUX_1	R/W	FFFD0320	32
Increment of texture U coordinate for a pixel step in Y direction	GPU2D0TXA_DUY_1	R/W	FFFD0324	32
Increment of texture V coordinate for a pixel step in X direction	GPU2D0TXA_DVX_1	R/W	FFFD0328	32
Increment of texture V coordinate for a pixel step in Y direction	GPU2D0TXA_DVY_1	R/W	FFFD032C	32
ZSA cache scheduler pitch	GPU2D0ZSS_PITCH	R/W	FFFD0400	32
ZSA cache scheduler span config	GPU2D0ZSS_SPAN_CONFIG	R/W	FFFD0404	32
Start value for Z coordinate	GPU2D0TXA_Z_OFFSET	R/W	FFFD0408	32
Increment of Z coordinate for a pixel step in X direction	GPU2D0TXA_DZX	R/W	FFFD040C	32
Increment of Z coordinate for a pixel step in Y direction	GPU2D0TXA_DZY	R/W	FFFD0410	32
Start value for RHW attribute (1/w) of Z	GPU2D0TXA_Z_RHW_OFFSET	R/W	FFFD0414	32
Increment of RHW attribute of Z for a pixel step in X direction	GPU2D0TXA_Z_DRHWX	R/W	FFFD0418	32
Increment of RHW attribute of Z for a pixel step in Y direction	GPU2D0TXA_Z_DRHWY	R/W	FFFD041C	32
Start value for RHW attribute (1/w)	GPU2D0TXA_RHW_OFFSET_0	R/W	FFFD0420	32
Increment of RHW attribute for a pixel step in X direction	GPU2D0TXA_DRHWX_0	R/W	FFFD0424	32
Increment of RHW attribute for a pixel step in Y direction	GPU2D0TXA_DRHWY_0	R/W	FFFD0428	32
Start value for RHW attribute (1/w)	GPU2D0TXA_RHW_OFFSET_1	R/W	FFFD042C	32
Increment of RHW attribute for a pixel step in X direction	GPU2D0TXA_DRHWX_1	R/W	FFFD0430	32
Increment of RHW attribute for a pixel step in Y direction	GPU2D0TXA_DRHWY_1	R/W	FFFD0434	32
ZSA control registers	GPU2D0ZSA_CTRL	R/W	FFFD0600	32
ZSA buffer start address	GPU2D0ZSC_START_ADDRESS	R/W	FFFD0604	32
ZSA Unit Control register 1	GPU2D0ZSU_CTRL1	R/W	FFFD0608	32
ZSA Unit Control register 2	GPU2D0ZSU_CTRL2	R/W	FFFD060C	32
Texture size	GPU2D0TXA_SIZE_0	R/W	FFFD0610	32
Filter shape, size and texture pitch	GPU2D0TXA_ACCESS_0	R/W	FFFD0614	32
Texture size	GPU2D0TXA_SIZE_1	R/W	FFFD0618	32
Filter shape, size and texture pitch	GPU2D0TXA_ACCESS_1	R/W	FFFD061C	32
Global texture unit register	GPU2D0TEX_GLOBAL	R/W	FFFD0800	32
Global color unit registers	GPU2D0COL_GLOBAL	R/W	FFFD0804	32
Framebuffer read decision registers	GPU2D0FBD_CONFIG	R/W	FFFD0808	32

Table 42.17 Register list by address (4/5)

Register name	Shortcut	R/W	Address	Access Size
Framebuffer cache scheduler pitch and debug control	GPU2D0FBS_PITCH	R/W	FFFD080C	32
Framebuffer cache scheduler span config	GPU2D0FBS_SPAN_CONFIG	R/W	FFFD0810	32
Start address of RLE code of the texture in memory	GPU2D0RLD_START_ADDRESS	R/W	FFFD0814	32
Constant color registers	GPU2D0COL_CONST_COLOR_0	R/W	FFFD0818	32
Constant color registers	GPU2D0COL_CONST_COLOR_1	R/W	FFFD081C	32
Constant color registers	GPU2D0COL_CONST_COLOR_2	R/W	FFFD0820	32
Constant color registers	GPU2D0COL_CONST_COLOR_3	R/W	FFFD0824	32
Texture mode	GPU2D0TEX_MODE_0	R/W	FFFD0A00	32
Start address of top-left texel of the texture in memory	GPU2D0TXC_START_ADDRESS_0	R/W	FFFD0A04	32
TXP control	GPU2D0TXP_CTRL_0	R/W	FFFD0A08	32
Offset for all CLUT lookups of the texture unit	GPU2D0TXP_CLUT_OFFSET_0	R/W	FFFD0A0C	32
Color key comparison value	GPU2D0TXP_COLOR_KEY_0	R/W	FFFD0A10	32
Texel fill color for wrap mode FILL	GPU2D0TXP_FILL_COLOR_0	R/W	FFFD0A14	32
Scale and bias for filter in TXP	GPU2D0TXP_FILTER_SCALE_BIAS_0	R/W	FFFD0A18	32
Texture mode	GPU2D0TEX_MODE_1	R/W	FFFD0A1C	32
Start address of top-left texel of the texture in memory	GPU2D0TXC_START_ADDRESS_1	R/W	FFFD0A20	32
TXP control	GPU2D0TXP_CTRL_1	R/W	FFFD0A24	32
Offset for all CLUT lookups of the texture unit	GPU2D0TXP_CLUT_OFFSET_1	R/W	FFFD0A28	32
Color key comparison value	GPU2D0TXP_COLOR_KEY_1	R/W	FFFD0A2C	32
Texel fill color for wrap mode FILL	GPU2D0TXP_FILL_COLOR_1	R/W	FFFD0A30	32
Scale and bias for filter in TXP	GPU2D0TXP_FILTER_SCALE_BIAS_1	R/W	FFFD0A34	32
Control structure for Op1a (Alpha and RGB path)	GPU2D0COL_ARGB_OP1A_0	R/W	FFFD0B00	32
Control structure for Op1b (Alpha and RGB path)	GPU2D0COL_ARGB_OP1B_0	R/W	FFFD0B04	32
Control structure for Op2a (Alpha and RGB path)	GPU2D0COL_ARGB_OP2A_0	R/W	FFFD0B08	32
Control structure for Op2b (Alpha and RGB path)	GPU2D0COL_ARGB_OP2B_0	R/W	FFFD0B0C	32
Control structure for Op3 (Alpha and RGB path)	GPU2D0COL_ARGB_OP3_0	R/W	FFFD0B10	32
Control structure for core output (Alpha and RGB path)	GPU2D0COL_ARGB_OUT_0	R/W	FFFD0B14	32
Control structure for Op1a (Alpha and RGB path)	GPU2D0COL_ARGB_OP1A_1	R/W	FFFD0B18	32
Control structure for Op1b (Alpha and RGB path)	GPU2D0COL_ARGB_OP1B_1	R/W	FFFD0B1C	32
Control structure for Op2a (Alpha and RGB path)	GPU2D0COL_ARGB_OP2A_1	R/W	FFFD0B20	32
Control structure for Op2b (Alpha and RGB path)	GPU2D0COL_ARGB_OP2B_1	R/W	FFFD0B24	32
Control structure for Op3 (Alpha and RGB path)	GPU2D0COL_ARGB_OP3_1	R/W	FFFD0B28	32

Table 42.17 Register list by address (5/5)

Register name	Shortcut	R/W	Address	Access Size
Control structure for core output (Alpha and RGB path)	GPU2D0COL_ARGB_OUT_1	R/W	FFFD0B2C	32
Control structure for Op1a (Alpha and RGB path)	GPU2D0COL_ARGB_OP1A_2	R/W	FFFD0B30	32
Control structure for Op1b (Alpha and RGB path)	GPU2D0COL_ARGB_OP1B_2	R/W	FFFD0B34	32
Control structure for Op2a (Alpha and RGB path)	GPU2D0COL_ARGB_OP2A_2	R/W	FFFD0B38	32
Control structure for Op2b (Alpha and RGB path)	GPU2D0COL_ARGB_OP2B_2	R/W	FFFD0B3C	32
Control structure for Op3 (Alpha and RGB path)	GPU2D0COL_ARGB_OP3_2	R/W	FFFD0B40	32
Control structure for core output (Alpha and RGB path)	GPU2D0COL_ARGB_OUT_2	R/W	FFFD0B44	32
Framebuffer pixel organization/format	GPU2D0FB_PIXEL_ORG	R/W	FFFD0C00	32
Framebuffer start address	GPU2D0FBC_START_ADDRESS	R/W	FFFD0C04	32
Blend unit factor and mode	GPU2D0BLU_BLEND	R/W	FFFD0C08	32
Blend unit dither control	GPU2D0BLU_DITHER	R/W	FFFD0C0C	32
Blend unit write control	GPU2D0BLU_WRITE	R/W	FFFD0C10	32
Performance counter enable	GPU2D0PFC_ENABLE	R/W	FFFD0E00	32
Performance counter clear	GPU2D0PFC_CLEAR	R/W	FFFD0E04	32
Performance counter event selection	GPU2D0PFC_EVENT_SELECT_0	R/W	FFFD0E08	32
Performance counter event selection	GPU2D0PFC_EVENT_SELECT_1	R/W	FFFD0E0C	32
Performance counter event selection	GPU2D0PFC_EVENT_SELECT_2	R/W	FFFD0E10	32
Performance counter event selection	GPU2D0PFC_EVENT_SELECT_3	R/W	FFFD0E14	32
Clear unit color value	GPU2D0CLR_VALUE	R/W	FFFD0F00	32
Clear unit line config	GPU2D0CLR_LINE_CONFIG	R/W	FFFD0F04	32
Clear unit number of lines	GPU2D0CLR_CTRL	R/W	FFFD0F08	32
Framebuffer start address for clear unit	GPU2D0CLR_START_ADDRESS	R/W	FFFD0F0C	32

42.4 Setup of the GPU2D

1. Set Stream commands list to memory.
   Regarding the usage of stream command, please refer D/AVE HD Technical Reference Manual.
2. Set stream address to GPU2D registers
   - GPU2D0STC_RING_PAUSE_ADDRESS[020H] = stream end address
   - GPU2D0STC_STREAM_ADDRESS[01CH]= stream start address

   After setting GPU2D0STC_STREAM_ADDRESS, GPU2D automatically starts to read stream commands and execute them.

Section 43 Sprite Engine (SPEA)

This section contains a generic description of the Sprite Engine.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

43.1 Overview of the Sprite and RLE Units

43.1.1 Units

This microcontroller has the following number of units of the Sprite Engine.

Table 43.1 Units

Sprite Engine (SPEA)	D1L1	D1L2(H)	D1M1(H) D1M1-V2	D1M1A	D1M2(H)
Units	0	1	1	1	1
Names	–	SPEA0	SPEA0	SPEA0	SPEA0

43.1.2 Sprite and RLE Units indices

In this section following indices are used:

Table 43.2 Indices

Index	Meaning
n	Sprite Engine (SPEA) unit index n Throughout this section, the individual units of the Sprite Engines are identified by the index “n” (n = 0), for example, SPEAnPHAi for the SPEAn physical address register of RLE Engine i.
i	RLE Engine number index i The two entities of an RLE Engine are identified by the index “i” (i = 0, 1), for example, SPEAnPHAi for the physical address register of RLE Engine i.
j	RLE Unit number index j The RLE Units are identified by the index “j”: • D1L2(H), D1M1(H), D1M2(H): j = 0 • D1M1-V2, D1M1A: j = 0 to 3, for example, SPEAnRjPHAi for the physical address register of RLE Unit j.
k	Sprite Unit number index k The Sprite Units are identified by the index “k”: • D1L2(H), D1M1(H), D1M2(H): k = 0 to 2 • D1M1-V2, D1M1A: k = 0 to 3, for example, SPEAnSkEN for the enable register of Sprite Unit k.
m	Sprite index m The 16 sprites of a Sprite Unit k are identified by the index “m” (m = 0 to 15), for example, SPEAnSkDAm for the destination address register of Sprite Unit k’s sprite m.

43.1.3 Register addresses

All Sprite Engines register addresses are given as address offsets from the individual base addresses <SPEAn_base>.
The <SPEAn_base> addresses of each SPEAn are listed in the following table:

Table 43.3 Register base addresses <SPEAn_base>

SPEAn unit	<SPEAn_base> address
SPEA0	FFFD 4000_H

43.1.4 Clock supply

All Sprite Engines provide one clock input.

Table 43.4 Clock supply

SPEAn unit	SPEAn clock	Connected to
SPEA0	PBUS clock	Clock Controller C_ISO_PCLK
	Cross-connect clock	Clock Controller C_ISO_XCCLK

43.1.5 Reset sources

The Sprite Engines and their registers are initialized by the following reset signal:

Table 43.5 Reset sources

SPEAn unit	Reset signal
SPEA0	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

43.1.6 Internal signal connections

The internal signal connections of the Sprite Engine are listed in the following table.

Table 43.6 SPEA internal signal connections

SPEAn signal	Function	Connected to
VUPDATE0	Sprite definition registers update triggers	Sprite Engine registers update control VUPDATE0
VUPDATE1		Sprite Engine registers update control VUPDATE1

Refer to Section 37.6, Sprite Engine registers update control for details the sprite definition registers update trigger selections.

43.1.7 Bus master IDs

The Sprite Engines bus master interfaces are connected to the XC0 cross-connect system. The master interfaces have the following master IDs:

- All devices except D1M1A, D1M1-V2
  - RLE unit: MSTID8
  - Sprite unit 0: MSTID9
  - Sprite unit 1: MSTID10
  - Sprite unit 2: MSTID11
- D1M1A, D1M1-V2
  - RLE unit 0/Sprite unit 3: MSTID8
  - RLE unit 1/Sprite unit 0: MSTID9
  - RLE unit 2/Sprite unit 1: MSTID10
  - RLE unit 3/Sprite unit 2: MSTID11

43.2 Functional Overview

The Image Synthesizers can select different ways to acquire their data from the memory for adding a layer to the image to be displayed:

- Direct access
  The color data to be processed by an Image Synthesizer is read directly in the correct format out of the memory.
  All video channel's Image Synthesizer's 0 to 3 can directly access the memory.
- RLE (run-length encoded) layer
  RLE compressed color data in the memory is expanded and delivered to the Image Synthesizer as separate pixel color data.
  - All devices except D1M1A, D1M1-V2
    The video channel's Image Synthesizer 0, which determines the background layer, can use an RLE layer.
  - D1M1A, D1M1-V2
    All Image Synthesizers can use an RLE layer.
- Sprite layer
  A layer with up to 16 rectangle areas, which can be placed anywhere on the layer area.
  Since only the color data for these rectangles - and not for the entire layer - need to be stored in memory, required memory occupation and bandwidth is minimized.
  - All devices except D1M1A, D1M1-V2
    The video channel's Image Synthesizer's 1 to 3, which add the layers 1 to 3, can use Sprite layers
  - D1M1A, D1M1-V2
    All Image Synthesizers can use Sprite layers.

Features summary

- RLE Units
  - separate RLE Engines for each video channel's background layer, which is generated by an Image Synthesizer (only by Image Synthesizer 0 for all devices except D1M1A, D1M1-V2)
  - RLE compressed data in Targa format
  - RLE compressed color data formats: 24 bpp, 18 bpp and 8 bpp
- Sprite Units
  - four Sprite Units for video channel's layer 0 to 3, which are generated by an Image Synthesizer (only by Image Synthesizer 1 to 3 for all devices except D1M1A, D1M1-V2)
  - each Sprite Unit is shared by two Image Synthesizers, one from each video channel
  - up to 16 separate sprites processed by each Sprite Unit
  - each sprite of each Sprite Unit can be freely assigned to both Image Synthesizers
  - all Image Synthesizer's color data formats are supported
- RLE/Sprite Units (D1M1A, D1M1-V2 only)
  - RLE/Sprite Units are selectable to be RLE unit or Sprite Unit by register setting.
  - four RLE/Sprite Units are connected to XC2 cross-connect

43.3 RLE Units Functional Description

The following figure shows the block diagram of the RLE Units.

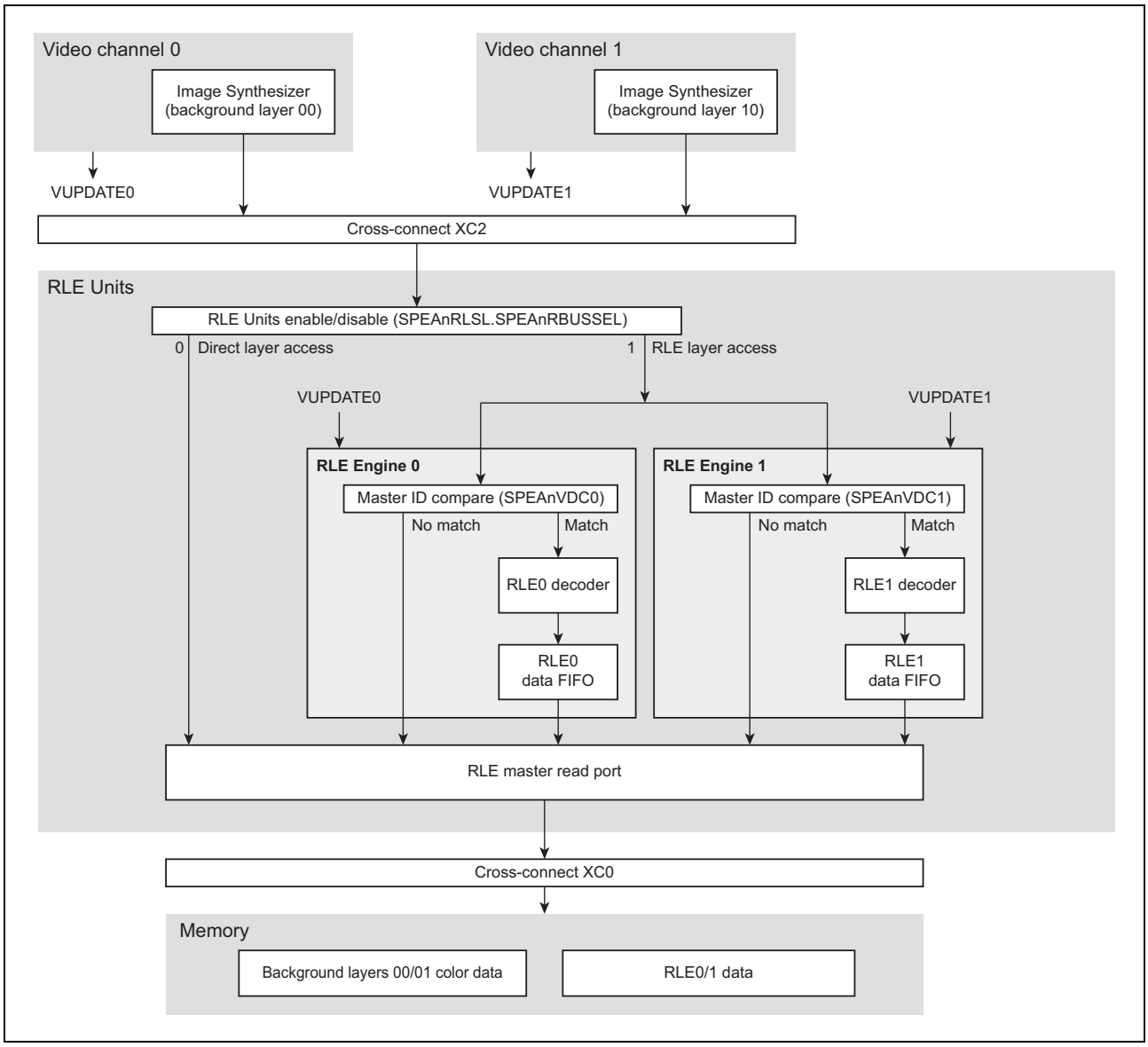


Figure 43.1 RLE Engines block diagram (all devices except D1M1A, D1M1-V2)

The Image Synthesizers 00 and 10 (background layers) can read RLE compressed color data from the memory via an RLE Engine.
If a background layer Image Synthesizer does not read RLE data, it accesses the physical memory for reading frame buffer data directly.

NOTE

Refer to Section 43.5, RLE/Sprite Units (D1M1A, D1M1-V2 only) for details about the D1M1A, D1M1-V2 RLE configuration.

RLE Engines enable

The RLE Engines can be enabled and disabled:

RLE or direct layer selection is done by a bypass switch:

- SPEAnRLSL.SPEAnRBUSSEL = 0: RLE Engines disabled
  The Image Synthesizer performs direct layer accesses to the memory.
- SPEAnRLSL.SPEAnRBUSSEL = 1: RLE Engines enabled
  The Image Synthesizer performs RLE layer accesses.

Direct or RLE layer access is commonly selected for the background layer of both video channels.

RLE Engines assignment and activation

Each Image Synthesizer background layer has its own RLE Engine:

- video channel 0 Image Synthesizer layer 00 uses RLE Engine 0
- video channel 1 Image Synthesizer layer 10 uses RLE Engine 1

Separate activation and de-activation of an RLE Engine for a background layer is done by compare of the Image Synthesizers cross-connect master ID with the predefined master ID in the respective RLE Engine.

The master ID to active the RLE Engine i is programmable in the SPEAnVDCi register.
If the ID of the master, reading data via an RLE Engine, does not match the SPEAnVDCi ID, the RLE decoder of RLE Engine i is deactivated.

RLE Units data FIFO

The data read from the memory by an RLE Engine is stored in a RLE data FIFO, that allows prefetching of memory data.
The prefetching of data can be influenced by the FIFO fill stage and the length of the read bursts, the RLE Engine issues read accesses towards the memory via the XC0 cross-connect.

The size of each RLE data FIFO is 1024 bit, organized as 16 x 64 bit.

Two values in the RLE prefetch configuration register SPEAnRCFG allow to configure the FIFO fill threshold when a new read burst is issued and the length of the burst:

- SPEAnRDTH[2:0] set the FIFO fill threshold for the next data prefetch
  The threshold is defined as SPEAnRDTH[2:0] x 64 bit.
- SPEAnRLEN[2:0] set the burst length for the next data prefetch
  The burst length is defined as SPEAnRLEN[2:0] x 64 bit.

NOTE

The burst size SPEAnRLEN[2:0] and the prefetch timing SPEAnRDTH[2:0] must be set to values, that prevent the RLE data FIFO from overflow.
The RLE data FIFO size is 1024 bit.

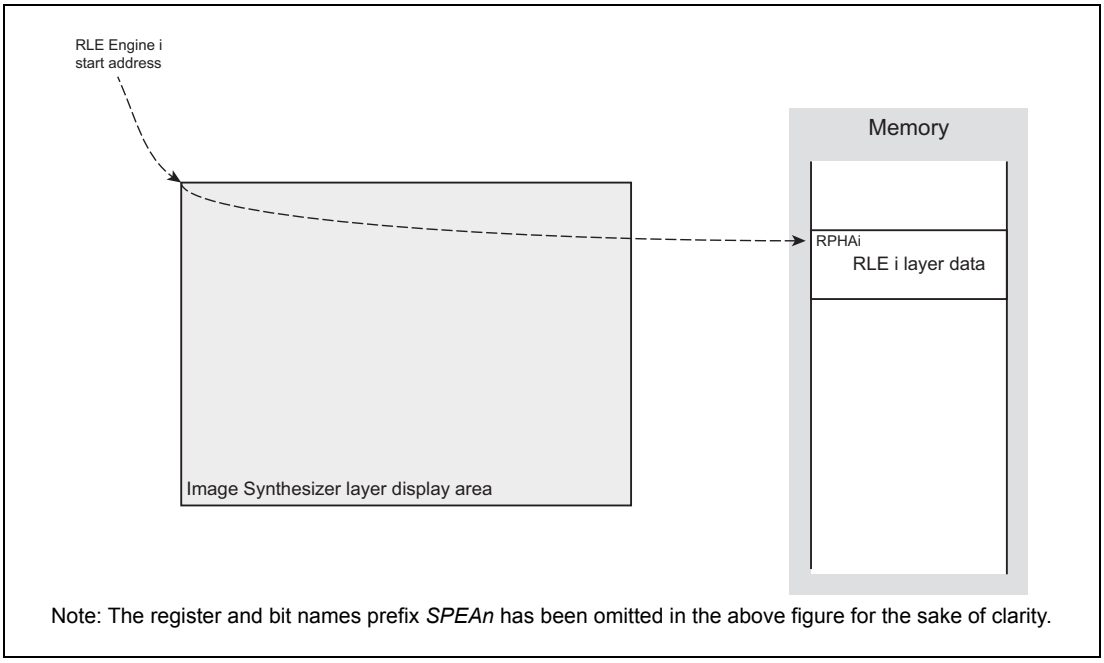
43.3.1 RLE layer definition

The RLE Unit j Engine i identifies a RLE layer read access of an Image Synthesizer by

- Image Synthesizer's bus master ID
  The valid bus master ID is defined in the read master ID register SPEAnVDCi
- the Image Synthesizer's read address of the first layer pixel
  The valid address is defined in the start address register SPEAnSTAi, that marks the virtual frame start address.

If an Image Synthesizer read access matches the valid master ID and address, the RLE Engine reads RLE data from the memory address, defined by the physical address register SPEAnPHAi.

Since the entire Image Synthesizer layer data is read via the RLE Engine, the RLE Engine reads data from the memory consecutively, until the Image Synthesizer issues the start address SPEAnSTAi again. This causes the RLE Engine to restart data read from address SPEAnPHAi.



Note: The register and bit names prefix *SPEAn* has been omitted in the above figure for the sake of clarity.

Figure 43.2 RLE definitions

CAUTION

As the RLE Engine is reading the memory consecutively, the following Image Synthesizer restrictions need to be considered:

1. **The Image Synthesizer always issues read bursts aligned to 128byte.**
   If an image width does not fit to this alignment, some dummy border (some pixels in each line) needs to be added.
2. **The Image Synthesizer performs an additional burst read of 128 bytes after every line.**
   In order to display its contents correctly any compressed image needs an additional "dummy border" of 32 pixels on its right side.

Example: for an 32 bpp image of 240 pixel width following dummy border needs to be added: In each line 16 pixels need to satisfy the first restriction and 32 pixels to satisfy

the second. The resulting total image width will be 288 pixel.

43.3.2 Color modes

The following table lists the possible combinations of the

- color data formats selectable in the Image Synthesizer,
- RLE color mode selection and the data format delivered to the Image Synthesizer
- color data format in the memory

when RLE data is used for the background layer.

Table 43.7 RLE Units supported color formats

Image Synthesizer color format selection		RLE Engine configuration		
		Color mode selection	Image Synthesizer data	Color data in the memory
32 bpp	αRGB8888	SPEAnRCMi = 2: 24 bpp	RGB888 with α = 1	RGB888
		SPEAnRCMi = 1: 18 bpp	RGB666 expanded to RGB888 with α = 1	RGB666
	RGBα8888	not supported	–	–
24 bpp	RGB888	SPEAnRCMi = 2: 24 bpp	RGB888	RGB888
		SPEAnRCMi = 1: 18 bpp	RGB666 expanded to RGB888	RGB666
16 bpp	RGB565 αRGB1555, RGBα5551 αRGB4444	not supported by RLE units	–	–
8 bpp	CLUT8	SPEAnRCMi = 0: 8 bpp	1 x 8 bpp in 1 byte	One 8 bpp color index in one byte
4 bpp	CLUT4	SPEAnRCMi = 0: 8 bpp	2 x 4 bpp in 1 byte	Two 4 bpp color indices in one byte
1 bpp	CLUT1	SPEAnRCMi = 0: 8 bpp	8 x 1 bpp in 1 byte	Eight 1 bpp color indices in one byte

Color expansion

Depending on the color format the Image Synthesizer is expecting, the color data read from the memory is expanded to fit a 32-bit word.

The following diagrams show how memory data in different formats are passed to the Image Synthesizer.

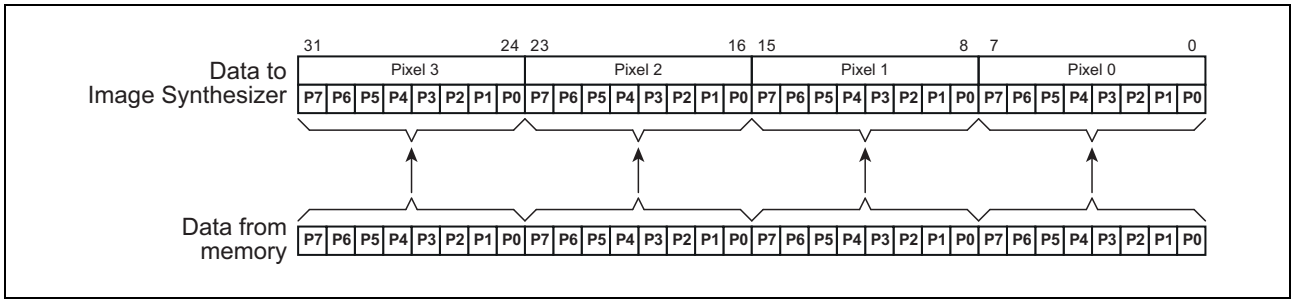


Figure 43.3 8 bpp color expansion to 32 bpp

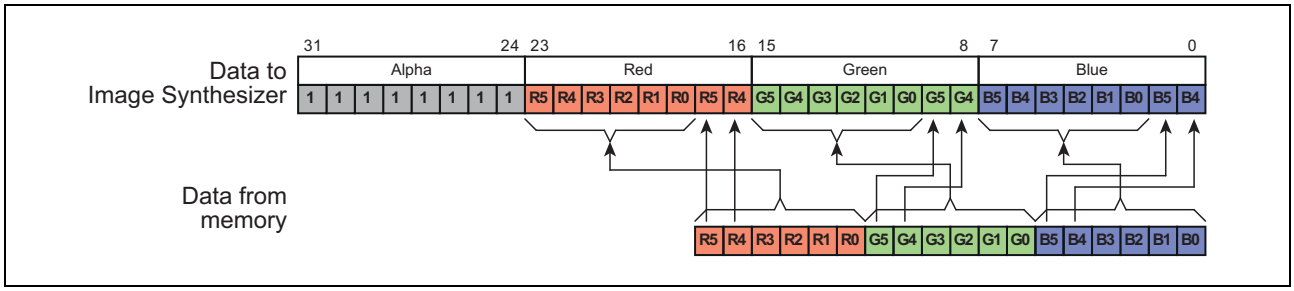


Figure 43.4 18 bpp color expansion to 32 bpp

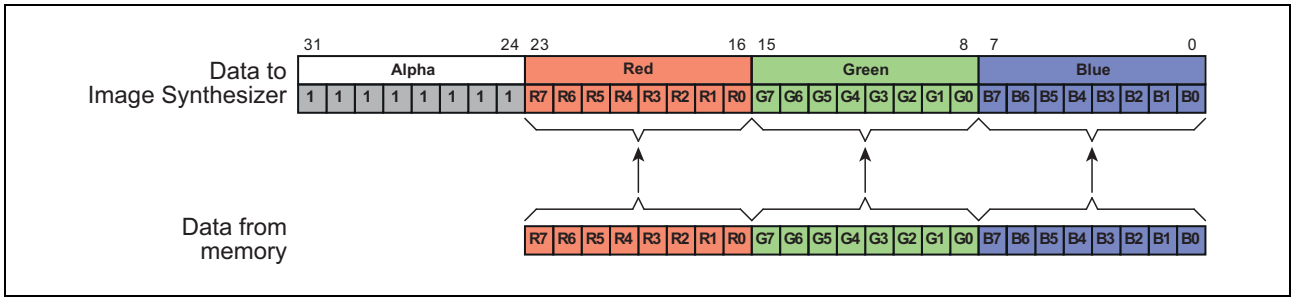


Figure 43.5 24 bpp color expansion to 32 bpp

CAUTION

If an RLE Unit reads data from the on-chip Video RAM, a direct 32-bit access area (with or without Error Correction Coding) must be used in order to avoid additional wrapping/unwrapping of the color data by the Video RAM Wrapper.

43.3.3 RLE data packets in the memory

The RLE Units support Targa RLE data packet formats.
The Targa RLE packets comprise two types of data elements:

- run-length packets
- raw packets

The packet consists of two fields:

- Control byte: defines the type of data and the number of pixels covered by the packet

- Data field: defines the pixels color data

Table 43.8 RLE Targa packets

Packet type	Control byte		Data field
Run-length	• Packet type = 1 for run-length packet • Size: 1 bit	• Pixel count = (number of pixel repetitions with color of data field) - 1 • Size: 7 bit	• Pixel color • Size: pixel bpp
Raw	• Packet type = 0 for raw packet • Size: 1 bit	• Pixel count = (number of pixels in following data field) - 1 • Size: 7 bit	• Pixels color • Size: pixel bpp x pixel count

43.3.4 Packet arrangement in the memory

Padding bits

A data packet must always start at a byte boundary. Thus it may be necessary to add padding bits to the previous packet.

The following diagrams show the arrangement of run-length and raw packets with different color format in the memory.

Figure 43.6 Run-length data packets in the memory

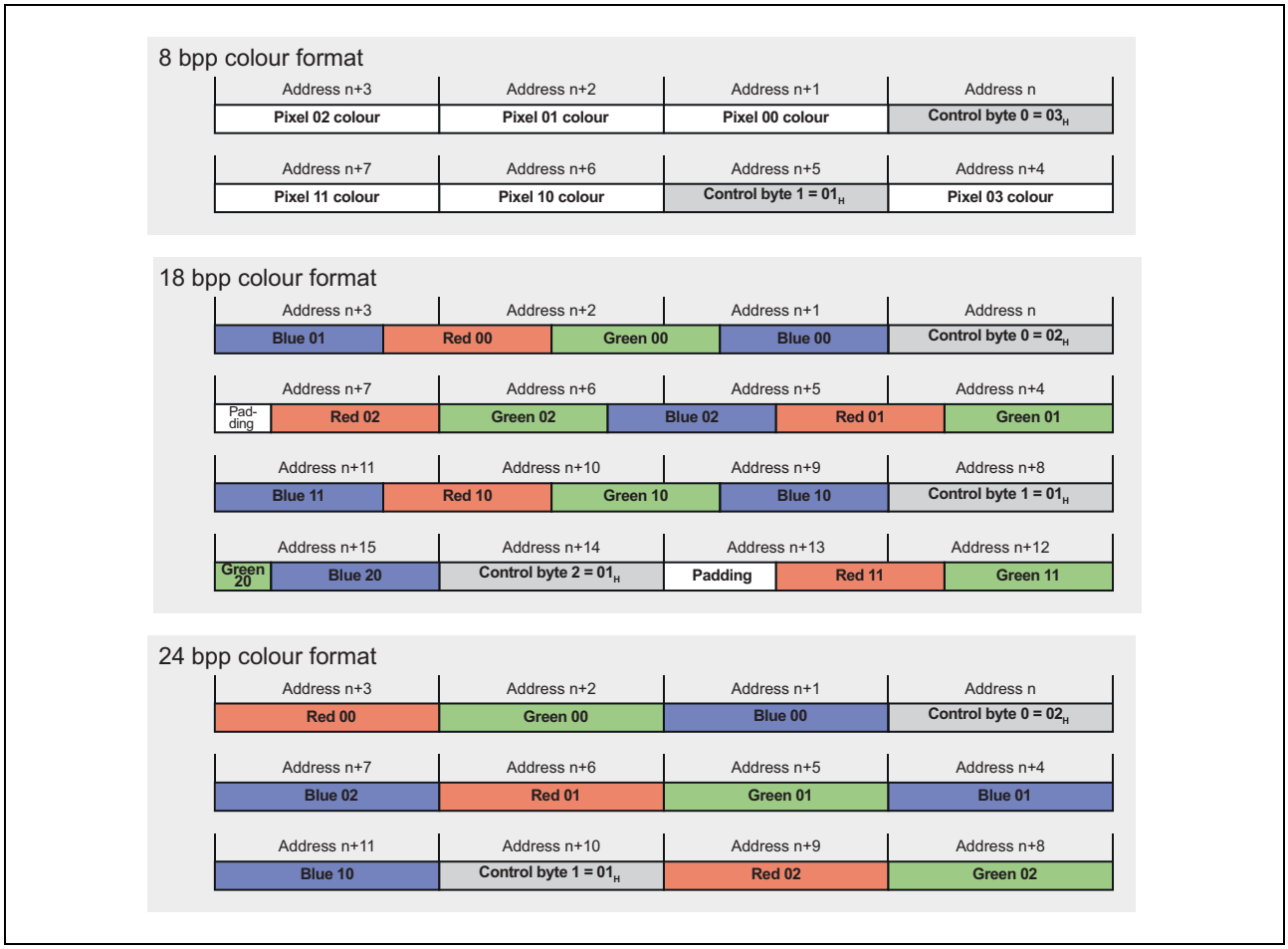


Figure 43.7 Raw data packets in the memory

The following diagram shows an example with a mixture of different packet types with 18 bpp data.

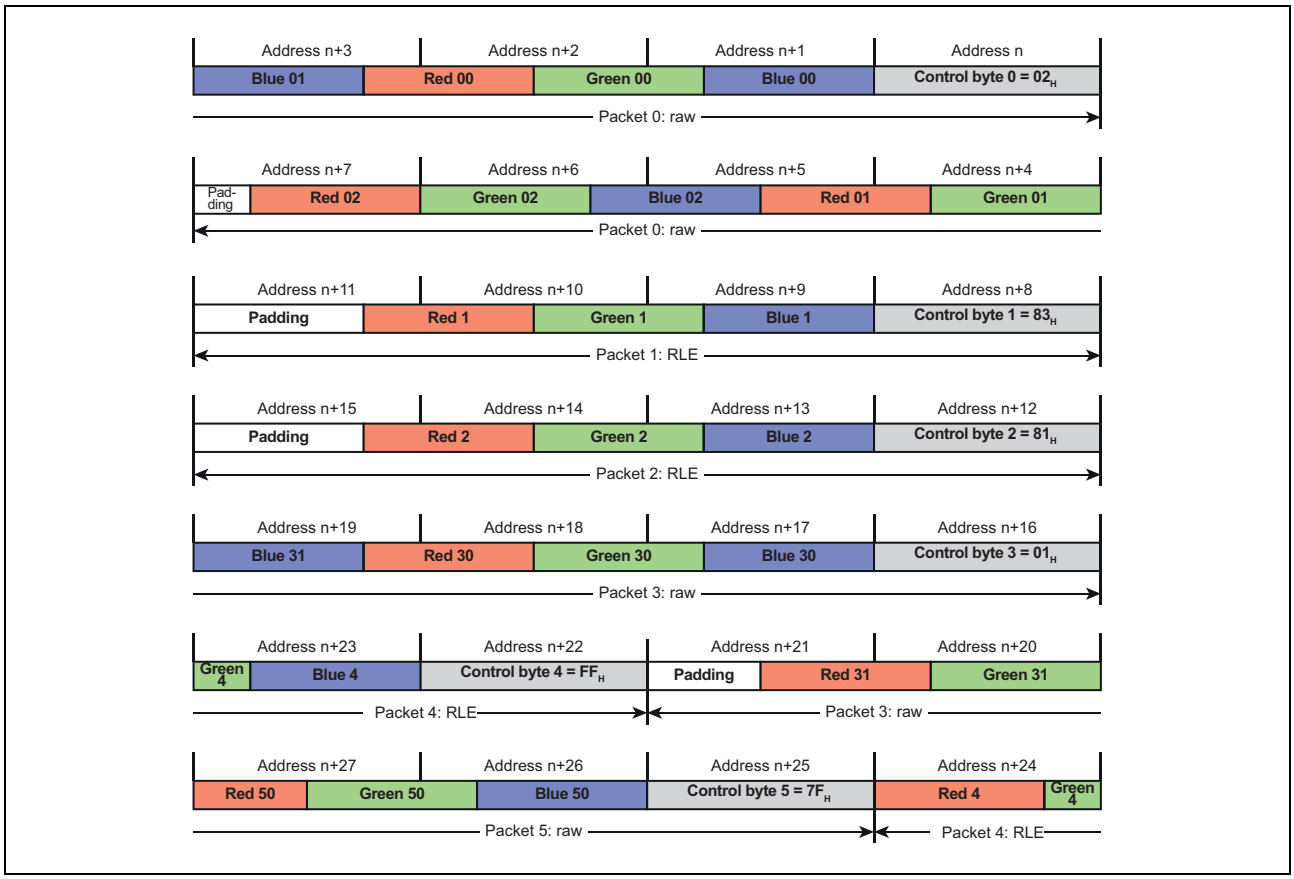


Figure 43.8 Mixture of RLE and raw data packets in the memory

43.3.5 RLE definition registers modification

The RLE definition registers are buffered.
While the RLE definition registers are in operation, the buffer registers can be modified without disturbing the video output.
In case the RLE definitions must be changed, the buffer registers are written with new values and all new sprite definitions become active with the start of a new video output frame, i.e. with a VUPDATEn (n = 0 or 1) signal.

The buffered RLE Engine i definition registers are:

- Start address register SPEAnSTAi
- physical address register SPEAnPHAi
- read master ID register SPEAnVDCi
- color mode register SPEAnRCMi

When setup of new RLE definitions is completed, the update of all definition registers of RLE Engine i must be requested via the SPEAnRUP register:

- D1L2(H), D1M1(H), D1M2(H) only
  - SPEAnRUP.SPEAnRUP0 = 1: initiates update of all RLE Engine 0 definition registers with the next VUPDATE0
  - SPEAnRUP.SPEAnRUP1 = 1: initiates update of all RLE Engine 1 definition registers with the next VUPDATE1
- D1M1A, D1M1-V2 only
  - SPEAnRjRUP.SPEAnRUP0 = 1: initiates update of all RLE Unit j, RLE Engine 0 definition registers with the next VUPDATE0
  - SPEAnRjRUP.SPEAnRUP1 = 1: initiates update of all RLE Unit j, RLE Engine 1 definition registers with the next VUPDATE1

CAUTION

All RLE definition buffer registers must not be modified while SPEAnRUPi = 1.

43.4 Sprite Units functional description

The Image Synthesizers for the layers 1 to 3 of the video channel's read accesses to the memory are performed via the XC2 cross-connect and the Sprite Units of the Sprite Engine.

The assignment of the video channel's Image Synthesizer layers to the three Sprite Units is determined by separate layers of the multi-layer cross-connect XC2, as shown in the figure below.

NOTE

Sharing of Sprite Units 0 and 2 by a video channel's layer 3 with the layer 1 of the other video channel achieves an optimum balance of the bandwidth to read from the memory, under the assumption that in most applications
– reading data for most upper layer 3 generates lowest memory bandwidth occupation
– reading data for lowest layer 1 generates highest memory bandwidth occupation.

An Image Synthesizer can use a layer, that can be populated with sprites, through access to a virtual frame buffer, the sprite virtual frame.
If an Image Synthesizer does not implement a sprite layer, it accesses the physical memory for reading frame buffer data directly.

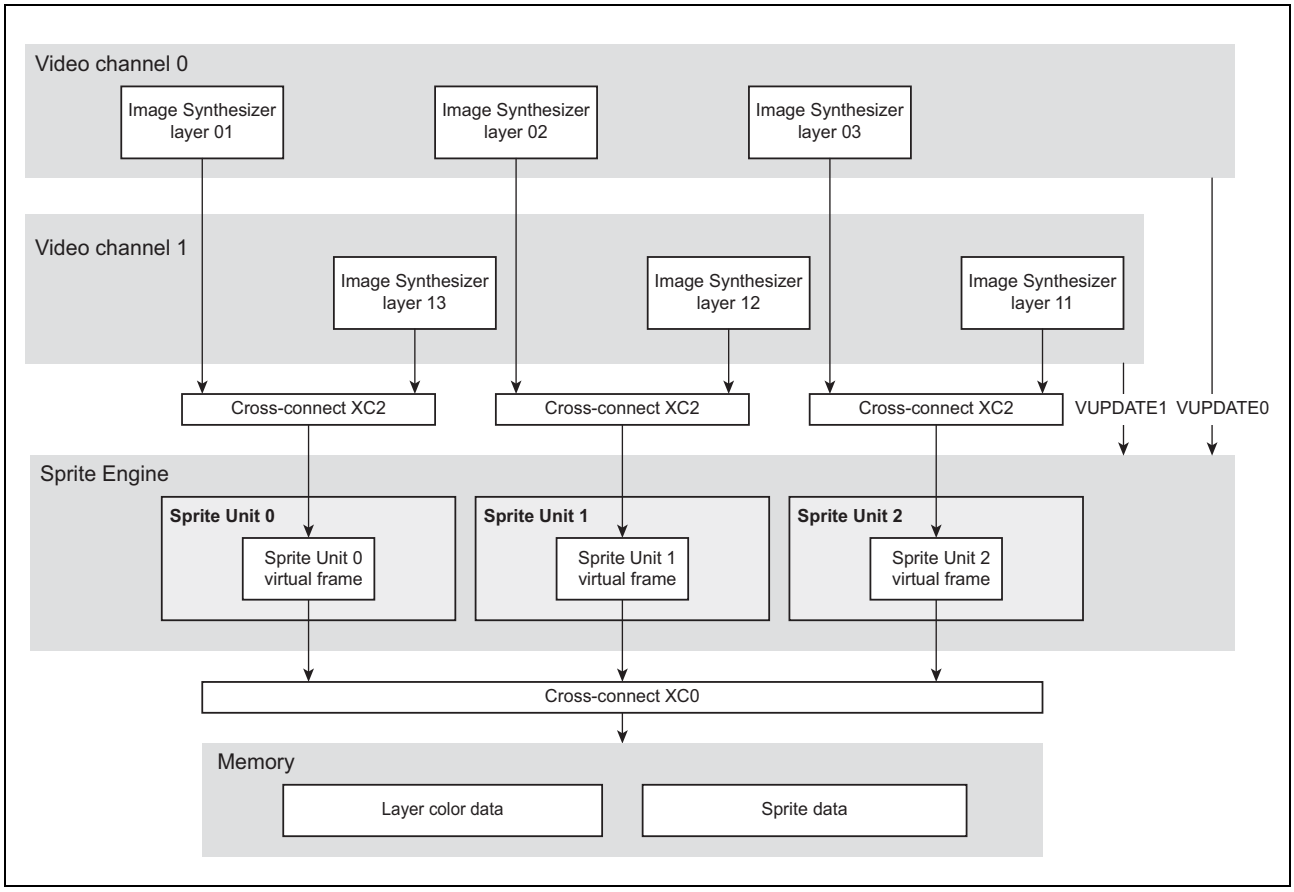


Figure 43.9 Sprite Units block diagram

NOTE

Refer to Section 43.5, RLE/Sprite Units (D1M1A, D1M1-V2 only) for details about the D1M1A, D1M1-V2 RLE configuration.

43.4.1 Sprite virtual frame

The size of the sprite virtual frame is fixed to 8192 x 8192 byte.
Its address is mapped outside of any physical address space, that is used to access the physical memory directly.

The address of the first pixel of the virtual frame is the virtual frame base address

Virtual frame base address = $3000\ 0000_H$

43.4.2 Sprite activation

Each sprite m of a Sprite Unit k can be enabled or disabled separately:

- Sprite enable/disable registers
  - SPEAnSkEN.SPEAnSkENm = 1 enables sprite k
  - SPEAnSkDS.SPEAnSkDSm = 1 disables sprite k

The enable/disable status of each sprite can be read via the SPEAnSkEN register.

43.4.3 Sprite definition

A sprite m of Sprite Unit k is defined by several register settings:

Sprite X/Y position

Sprite X/Y position register SPEAnSkPSm: X/Y position on the sprite virtual frame, with respect to its virtual frame base address.

(a) SPEAnSkPSm.SPEAnSkPSXm: X position

The X position is defined as multiple of 64 bit and thus depends on the sprite color format.
The corresponding pixel position depends on the color format and the amount of pixels fitting into 64 bits.

- 8 bpp color formats: 64 bit / 8 = 8 pixel units
  - possible X positions at a multiple of 8 = 0, 8, 16, 32, 64, ...
  - SPEAnSkPSX[9:0] = X position in pixel / 8
- 16 bpp color formats: 64 bit / 16 = 4 pixel units
  - possible X positions at a multiple of 4 = 0, 4, 8, 12, 16, ...
  - SPEAnSkPSX[9:0] = X position in pixel / 4
- 18/24/32 bpp color formats: 64 bit / 32 = 2 pixel units
  - possible X positions at a multiple of 2 = 0, 2, 4, 6, 8, ...
  - SPEAnSkPSX[9:0] = X position in pixel / 2

(b) SPEAnSkPSm.SPEAnSkPSYm: Y position

The Y position is defined in pixels.

Sprite width and height

Sprite size register SPEAnSkLYm: width SPEAnSkLYWm and height SPEAnSkLYHm

(a) SPEAnSkLYm.SPEAnSkLYWm: width

The width is defined as multiple of 64 bit and thus depends on the sprite color format.

- 8 bpp color formats: 64 bit / 8 = 8 pixel units
  - possible widths: multiple of 8 = 0, 8, 16, 32, 64, ...
  - SPEAnSkLYW[9:0] = width in pixel / 8
- 16 bpp color formats: 64 bit / 16 = 4 pixel units
  - possible widths: multiple of 4 = 0, 4, 8, 12, 16, ...
  - SPEAnSkLYW[9:0] = width in pixel / 4
- 18/24/32 bpp color formats: 64 bit / 32 = 2 pixel units
  - possible X widths: multiple of 2 = 0, 2, 4, 6, 8, ...
  - SPEAnSkLYW[9:0] = width in pixel / 2

(b) SPEAnSkLYm.SPEAnSkLYHm: height

The height is defined in pixels.

Destination address

Destination address register SPEAnSkDAm: address of 1st pixel color data of sprite m in the memory
The destination address must be aligned to 64 bit.

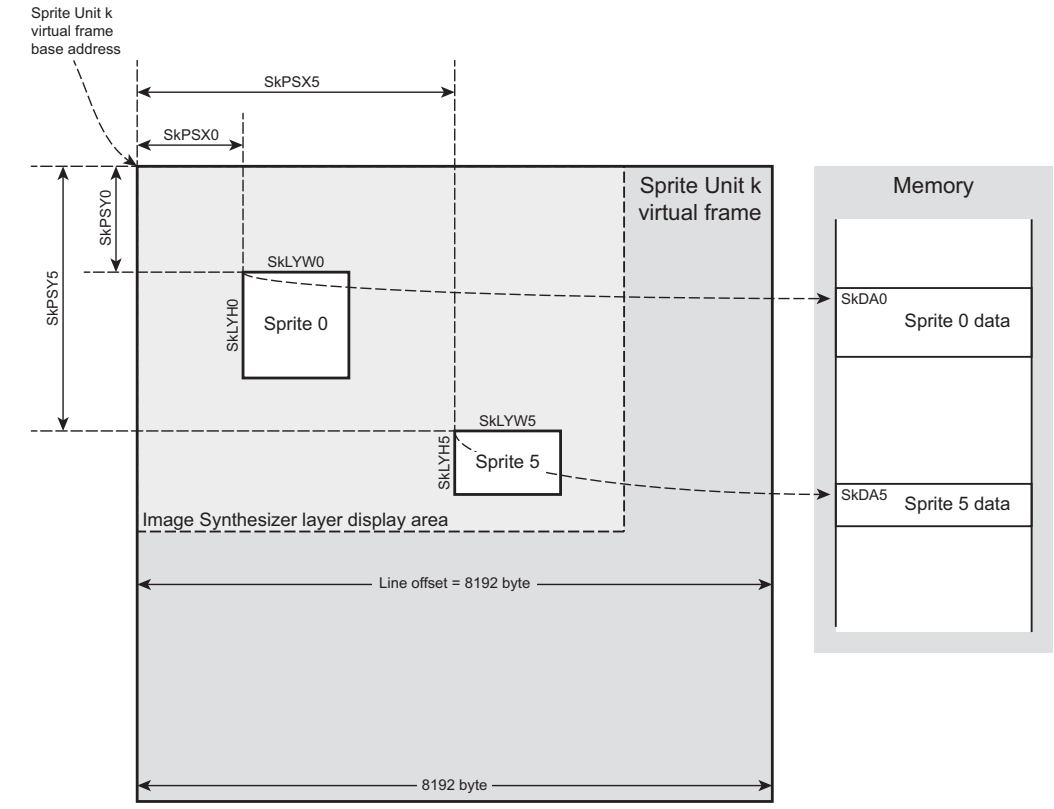
Image Synthesizer line offset

Since the Sprite Unit calculates the address of each new line with respect to the fixed virtual frame width of 8192 Byte, the Image Synthesizer must define the same line offset, when reading layer data via a Sprite Unit.

Note that the maximum number of pixels per line in the virtual frame depends on the color format:

- 18/24/32 bpp: max. 2048 pixels
- 16 bpp: max. 4096 pixels
- 8 bpp: max. 8192 pixels

The figure below shows an example with two enabled sprites.



Note: The register and bit names prefix *SPEAn* has been omitted in the above figure for the sake of clarity.

Figure 43.10 Sprite definitions

43.4.4 Sharing of sprites by two Image Synthesizers

A Sprite Unit can not be exclusively assigned to a certain Image Synthesizer.
It's activation is only determined by the read address from an Image Synthesizer, provided the Sprite Unit is activated by SPEAnSkEN.SPEAnSkENm = 1.

Thus basically both Image Synthesizers will receive the same pixel color data, when they issue the same read address.

If a sprite is used by two Image Synthesizers and the sprite definitions shall be modified while the Sprite Unit is in operation, attention must be paid to the sprite definition update process (see Section 43.4.7, Sprite definition registers modification for details).

43.4.5 Overlapping sprites

Sprites can be also placed overlapping.
In such case a fixed priority order rules which sprite data is forwarded to the Image Synthesizer:

- Sprite m = 0: highest priority
- ...
- Sprite m = 15: lowest priority

The color data of the sprite with the highest priority is forwarded to the Image Synthesizer.

43.4.6 Sprite layer areas without active sprites

Areas of the sprite layer without active sprites return color data with value 0. These data are processed as fully transparent color to the Image Synthesizer.

43.4.7 Sprite definition registers modification

The sprite definition registers are buffered.
While the sprite definition registers are in operation, the buffer registers can be modified without disturbing the video output.
In case the sprite definitions must be change, the buffer registers are written with new values and all new sprite definitions become active with the start of a new video output frame, i.e. with a VUPDATEn (n=0 or 1) signal, provided that update is enabled by SPEAnSkUP.SPEAnSkUPn = 1 (see below for details).

The buffered Sprite Unit k sprite m definition registers are:

- Sprite Unit k enable registers SPEAnSkEN
- Sprite Unit k disable registers SPEAnSkDS
- Destination address registers SPEAnSkDAm
- Width/height registers SPEAnSkLYm
- X/Y position registers SPEAnSkPSm

Each sprite m is assigned to the VUPDATE0 or VUPDATE1 update group via the SPEAnSkVDm register:

- SPEAnSkVDm = 0: Sprite Unit k sprite m definition registers updated with VUPDATE0
- SPEAnSkVDm = 1: Sprite Unit k sprite m definition registers updated with VUPDATE1

All sprite definitions of the VUPDATE0 or VUPDATE1 update group are updated simultaneously with the respective VUPDATE0 or VUPDATE1 signal.

When setup of new sprite definitions is completed, the update of all sprite definition registers of Sprite Unit k must be requested via the SPEAnSkUP register:

- SPEAnSkUP.SPEAnSkUP0 = 1: initiates update of all Sprite Unit k sprite definition registers of update group VUPDATE0 with the next VUPDATE0
- SPEAnSkUP.SPEAnSkUP1 = 1: initiates update of all Sprite Unit k sprite definition registers of update group VUPDATE1 with the next VUPDATE1

CAUTION

All sprite definition buffer registers must not be modified while SPEAnSkUP0 = 1 or SPEAnSkUP1 = 1.

43.5 RLE/Sprite Units (D1M1A, D1M1-V2 only)

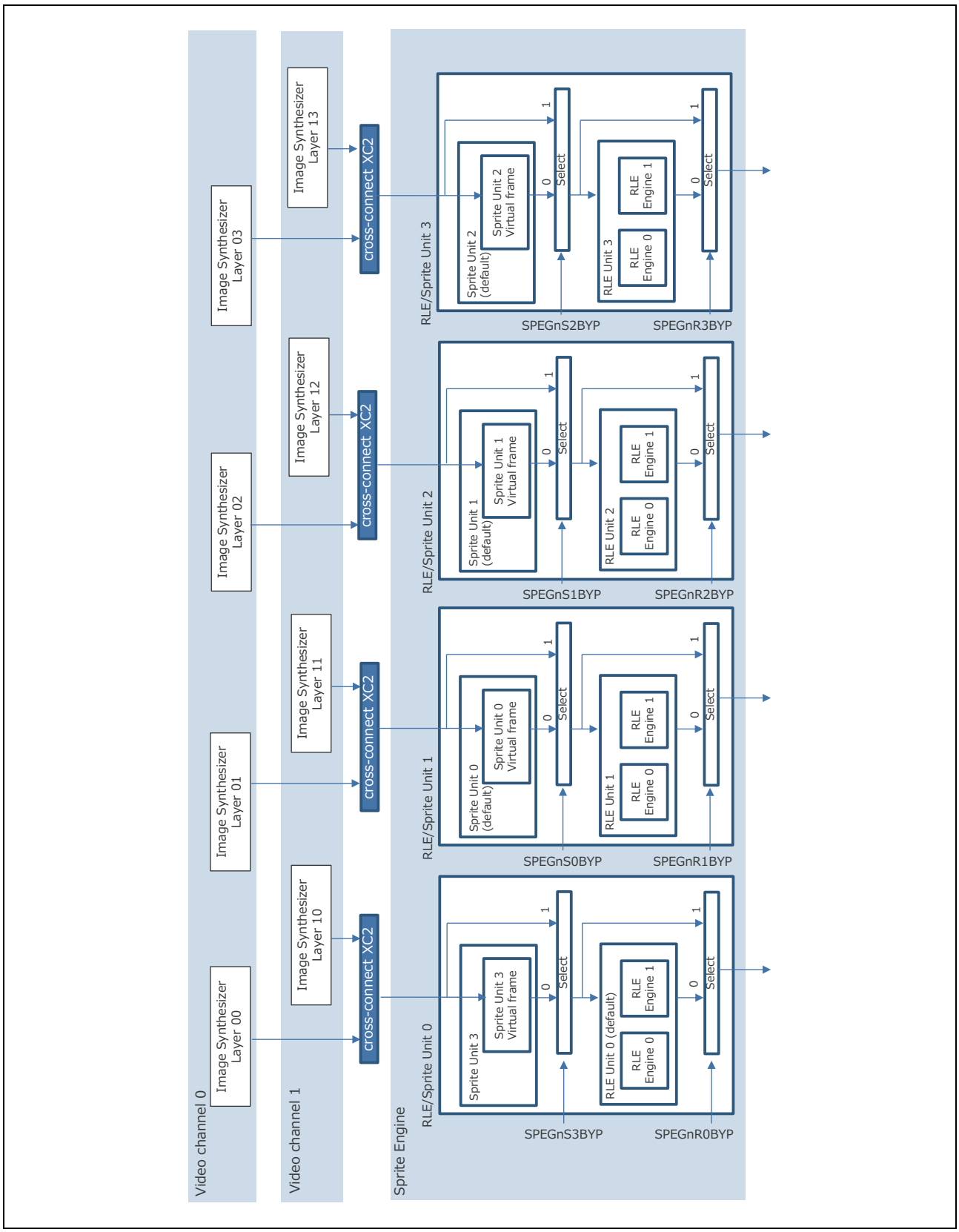


Figure 43.11 D1M1A, D1M1-V2 RLE/Sprite units block diagram

A RLE/Sprite Unit comprises both: an RLE unit and a Sprite Unit. The Usage as RLE or Sprite Unit is selectable. However, it is prohibit to select both units at the same .

SPEGnSmBYP, SPEGnRmBYP select to bypass them. In the initial setting, these register are set as shown in the table below.

Table 43.9 RLE/Sprite Unit initial setting

Unit	Initial register setting		Selected unit
RLE/Sprite Unit 0 (RLE Unit 0/Sprite Unit 3)	SPEGnS3BYP = 1	SPEGnR0BYP = 0	RLE Unit 0
RLE/Sprite Unit 1 (RLE Unit 1/Sprite Unit 0	SPEGnS0BYP = 0	SPEGnR1BYP = 1	Sprite Unit 0
RLE/Sprite Unit 2 (RLE Unit 2/Sprite Unit 1)	SPEGnS1BYP = 0	SPEGnR2BYP = 1	Sprite Unit 1
RLE/Sprite Unit 3 (RLE Unit 3/Sprite Unit 2)	SPEGnS2BYP = 0	SPEGnR3BYP = 1	Sprite Unit 2

NOTES

1. The setting {SPEGnS3BYP, SPEGnR0BYP} = {0,0} is prohibit.
   The setting {SPEGnS0BYP, SPEGnR1BYP} = {0,0} is prohibit.
   The setting {SPEGnS1BYP, SPEGnR2BYP} = {0,0} is prohibit.
   The setting {SPEGnS2BYP, SPEGnR3BYP} = {0,0} is prohibit.
2. When RLE unit is used, SPEAnRLSL.RBUSSEL must be set.

43.6 Sprite and RLE Units Registers

This section contains a description of all registers of the Sprite and RLE Units.

The Sprite and RLE Units are controlled and operated by the following registers:

Table 43.10 Sprite and RLE Units registers overview (all devices except D1M1A, D1M1-V2)

Register name	Shortcut	Address
RLE Units registers		
RLE enable control register	SPEAnRLSL	<SPEAn_base> + 10_H
RLE Engine i start address register	SPEAnSTAi	<SPEAn_base> + 20_H + i x 10_H
RLE Engine i physical address register	SPEAnPHAi	<SPEAn_base> + 24_H + i x 10_H
RLE Engine i read master ID register	SPEAnVDCi	<SPEAn_base> + 28_H + i x 10_H
RLE Engine i color mode register	SPEAnRCMi	<SPEAn_base> + $2C_H$ + i x 10_H
RLE register update request register	SPEAnRUP	<SPEAn_base> + 40_H
RLE configuration register	SPEAnRCFG	<SPEAn_base> + 48_H
Sprite Unit k registers		
Sprite Unit k enable register	SPEAnSkEN	<SPEAn_base> + 100_H + k x 10_H
Sprite Unit k disable register	SPEAnSkDS	<SPEAn_base> + 104_H + k x 10_H
Sprite Unit k update request register	SPEAnSkUP	<SPEAn_base> + 108_H + k x 10_H
Sprite Unit k sprite m destination address register	SPEAnSkDAm	<SPEAn_base> + 400_H + k x 400_H + m x 20_H
Sprite Unit k sprite m VUPDATEn selection register	SPEAnSkVDm	<SPEAn_base> + 404_H + k x 400_H + m x 20_H
Sprite Unit k sprite m width/height register	SPEAnSkLYm	<SPEAn_base> + 408_H + k x 400_H + m x 20_H
Sprite Unit k sprite m X/Y position register	SPEAnSkPSm	<SPEAn_base> + $40C_H$ + k x 400_H + m x 20_H

Table 43.11 RLE/Sprite Units registers overview (D1M1A, D1M1-V2 only)

RLE units j	Register name	Shortcut	Address
j = 0	RLE enable control register	SPEAnRjRLSL	<SPEAn_base> + 10_H
	RLE Engine i start address register	SPEAnRjSTAi	<SPEAn_base> + 20_H + i x 10_H
	RLE Engine i physical address register	SPEAnRjPHAi	<SPEAn_base> + 24_H + i x 10_H
	RLE Engine i read master ID register	SPEAnRjVDCi	<SPEAn_base> + 28_H + i x 10_H
	RLE Engine i color mode register	SPEAnRjRCMi	<SPEAn_base> + $2C_H$ + i x 10_H
	RLE register update request register	SPEAnRjRUP	<SPEAn_base> + 40_H
	RLE units bypass mode register	SPEAnRjRBYP	<SPEAn_base> + 44_H
	RLE configuration register	SPEAnRjRCFG	<SPEAn_base> + 48_H
j = 1 to 3	RLE enable control register	SPEAnRjRLSL	<SPEAn_base> + 1010_H + ((j-1) x 100_H)
	RLE Engine i start address register	SPEAnRjSTAi	<SPEAn_base> + 1020_H + ((j-1) x 100_H) + i x 10_H
	RLE Engine i physical address register	SPEAnRjPHAi	<SPEAn_base> + 1024_H + ((j-1) x 100_H) + i x 10_H
	RLE Engine i read master ID register	SPEAnRjVDCi	<SPEAn_base> + 1028_H + ((j-1) x 100_H) + i x 10_H
	RLE Engine i color mode register	SPEAnRjRCMi	<SPEAn_base> + $102C_H$ + ((j-1) x 100_H) + i x 10_H
	RLE register update request register	SPEAnRjRUP	<SPEAn_base> + 1040_H + ((j-1) x 100_H)
	RLE Units bypass mode register	SPEAnRjRBYP	<SPEAn_base> + 1044_H + ((j-1) x 100_H)
	RLE configuration register	SPEAnRjRCFG	<SPEAn_base> + 1048_H + ((j-1) x 100_H)
Sprite units k	**Register name**	**Shortcut**	**Address**
k = 0 to 3	Sprite Unit k enable register	SPEAnSkEN	<SPEAn_base> + 100_H + k x 10_H
	Sprite Unit k disable register	SPEAnSkDS	<SPEAn_base> + 104_H + k x 10_H
	Sprite Unit k update request register	SPEAnSkUP	<SPEAn_base> + 108_H + k x 10_H
	Sprite Unit k bypass mode register	SPEAnSkBYP	<SPEAn_base> + $10C_H$ + k x 10_H
k = 0 to 2	Sprite Unit k sprite m destination address register	SPEAnSkDAm	<SPEAn_base> + 400_H + k x 400_H + m x 20_H
	Sprite Unit k sprite m VUPDATEn selection register	SPEAnSkVDm	<SPEAn_base> + 404_H + k x 400_H + m x 20_H
	Sprite Unit k sprite m width/height register	SPEAnSkLYm	<SPEAn_base> + 408_H + k x 400_H + m x 20_H
	Sprite Unit k sprite m X/Y position register	SPEAnSkPSm	<SPEAn_base> + $40C_H$ + k x 400_H + m x 20_H
k = 3	Sprite Unit k sprite m destination address register	SPEAnSkDAm	<SPEAn_base> + 1400_H + m x 20_H
	Sprite Unit k sprite m VUPDATEn selection register	SPEAnSkVDm	<SPEAn_base> + 1404_H + m x 20_H
	Sprite Unit k sprite m width/height register	SPEAnSkLYm	<SPEAn_base> + 1408_H + m x 20_H
	Sprite Unit k sprite m X/Y position register	SPEAnSkPSm	<SPEAn_base> + $140C_H$ + m x 20_H

<SPEAn_base>

The base addresses <SPEAn_base> of the SPEAn is defined in the first section of this chapter under the key word “Register addresses”.

43.6.1 RLE Units registers

43.6.1.1 SPEAnRLSL/SPEAnRjRLSL - RLE Units enable control register

The RLE Units can be enabled and disabled via this register.

Access: This register can be accessed in 32-bit units.

Address: SPEAnRLSL: <SPEAn_base> + 10_H
SPEAnRjRLSL (j = 0): <SPEAn_base> + 10_H
SPEAnRjRLSL (j = 1 to 3): <SPEAn_base> + 1010_H + ((j-1) x 100_H) + i x 10_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SPEAn RBUSSEL
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 43.12 SPEAnRLSL/SPEAnRjRLSL register contents

Bit position	Bit name	Function
31 to 1	Reserved	When read, the value after reset is returned. When written, write the value after reset.
0	SPEAn RBUSSEL	RLE Units enable control 0: RLE Units are disabled (direct layer access) 1: RLE Units are enabled (RLE layer access)

43.6.1.2 SPEAnSTAi/SPEAnRjSTAi - RLE Engine i start address register

This register defines the Image Synthesizer's read address of the 1st pixel to read from the RLE Engine i.

Access: This register can be accessed in 32-bit units.

Address: SPEAnSTAi: <SPEAn_base> + 20_H + i x 10_H
SPEAnRjSTAi (j = 0): <SPEAn_base> + 20_H + i x 10_H
SPEAnRjSTAi (j = 1 to 3): <SPEAn_base> + 1020_H + ((j-1) x 100_H) + i x 10_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPEAnRSTAi[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPEAnRSTAi[15:9]							0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R

Table 43.13 SPEAnSTAi/SPEAnRjSTAi register contents

Bit position	Bit name	Function
31 to 9	SPEAn RSTAi[31:9]	RLE Engine i start address The address must be 128-byte aligned, so the lower 9 bits SPEAnSTAi[8:0] of the address are always 0 0000 0000_B.
8 to 0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

43.6.1.3 SPEAnPHAi/SPEAnRjPHAi - RLE Engine i physical address register

This register defines the address of the RLE Engine i first data in the memory.

Access: This register can be accessed in 32-bit units.

Address: SPEAnPHAi: <SPEAn_base> + 24_H + i x 10_H
SPEAnRjPHAi (j = 0): <SPEAn_base> + 24_H + i x 10_H
SPEAnRjPHAi (j = 1 to 3): <SPEAn_base> + 1024_H + ((j-1) x 100_H) + i x 10_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPEAnRPHAi[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPEAnRPHAi[15:3]													0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 43.14 SPEAnPHAi/SPEAnRjPHAi register contents

Bit position	Bit name	Function
31 to 3	SPEAn RPHAi[31:3]	RLE Engine i data memory address The address must be 64-bit aligned, so the lower 3 bits SPEAnPHAi[2:0] of the address are always 000_B.
2 to 0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

43.6.1.4 SPEAnVDCi/SPEAnRjVDCi - RLE Engine i read master ID register

This register selects the video channel to use RLE Engine i.

Access: This register can be accessed in 32-bit units.

Address: SPEAnVDCi: <SPEAn_base> + 28_H + i x 10_H
SPEAnRjVDCi (j = 0): <SPEAn_base> + 28_H + i x 10_H
SPEAnRjVDCi (j = 1 to 3): <SPEAn_base> + 1028_H + ((j-1) x 100_H) + i x 10_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	SPEAnRVDCi[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 43.15 SPEAnVDCi/SPEAnRjVDCi register contents

Bit position	Bit name	Function
31 to 4	Reserved	When read, the value after reset is returned. When written, write the value after reset.
3 to 0	SPEAn RVDCi[3:0]	This register defines the valid bus master ID for RLE Engine i. Since both RLE Engines are fixed assigned to the background layers of the video channels, correct setting of SPEAnRVDCi[3:0] is different for both RLE Engines. In order to deactivate a certain RLE Engine SPEAnRVDCi[3:0] must be set to an invalid bus master ID. For details about bus master ID assignment refer to Table 43.16, Master ID assignment (except D1M1A, D1M1-V2) and Section Table 43.17, Master ID assignment (D1M1A, D1M1-V2 only).

CAUTION

SPEAnRVDC0[3:0] and SPEAnRVDC1[3:0] must not define the same master ID, when RLE is enabled.

Table 43.16 Master ID assignment (except D1M1A, D1M1-V2)

SPEAnRVDCi[3:0]	0	1	2	Others
RLE Engine 0	Image Synthesizer 00	Image Synthesizer 10	Invalid master ID*1	Invalid master ID*1
RLE Engine 1				

Note 1. If invalid bus master ID is set to SPEAnRVDCi[3:0], RLE Engine i becomes inactive.

Table 43.17 Master ID assignment (D1M1A, D1M1-V2 only)

SPEAnRVDCi[3:0]		0	1	2	Others
RLE unit 0	RLE Engine 0	Image Synthesizer 00	Image Synthesizer 10	Invalid master ID*1	Invalid master ID*1
	RLE Engine 1				
RLE unit 1	RLE Engine 0	Image Synthesizer 01	Image Synthesizer 13	Output Image Generator	Invalid master ID*1
	RLE Engine 1				
RLE unit 2	RLE Engine 0	Image Synthesizer 02	Image Synthesizer 12	Setting prohibited	Invalid master ID*1
	RLE Engine 1				
RLE unit 3	RLE Engine 0	Image Synthesizer 03	Image Synthesizer 11	Invalid master ID*1	Invalid master ID*1
	RLE Engine 1				

Note 1. If invalid bus master ID is set to SPEAnRVDCi[3:0], RLE Engine i becomes inactive.

43.6.1.5 SPEAnRCMi/SPEAnRjRCMi - RLE Engine i color mode selection register

This register selects the color format of data in memory for RLE Engine i.

Access: This register can be accessed in 32-bit units.

Address: SPEAnRCMi: <SPEAn_base> + $2C_H$ + i x 10_H
SPEAnRjRCMi (j = 0): <SPEAn_base> + $2C_H$ + i x 10_H
SPEAnRjRCMi (j = 1 to 3): <SPEAn_base> + $102C_H$ + ((j-1) x 100_H) + i x 10_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SPEAnRCMi[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 43.18 SPEAnRCMi/SPEAnRjRCMi register contents

Bit position	Bit name	Function
31 to 2	Reserved	When read, the value after reset is returned. When written, write the value after reset.
1 to 0	SPEAn RCMi[1:0]	RLE Engine i color mode selection 00_B: 8 bpp 01_B: 18 bpp 10_B: 24 bpp 11_B: setting prohibited

43.6.1.6 SPEAnRUP/SPEAnRjRUP - RLE registers update request register

This register controls the update of the RLE Engine's registers.

Access: This register can be accessed in 32-bit units.

Address: SPEAnRUP: <SPEAn_base> + 40_H
SPEAnRjRUPi (j = 0): <SPEAn_base> + 40_H
SPEAnRjRUPi (j = 1 to 3): <SPEAn_base> + 1040_H + ((j-1) x 100_H)

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SPEAn RUP1	SPEAn RUP0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 43.19 SPEAnRUP/SPEAnRjRUP register contents

Bit position	Bit name	Function
31 to 2	Reserved	When read, the value after reset is returned. When written, write the value after reset.
1	SPEAn RUP1	RLE Engine 1 register update request 0: no function 1: update RLE Engine 1 register with the next VUPDATE1
0	SPEAn RUP0	RLE Engine 0 register update request 0: no function 1: update RLE Engine 0 register with the next VUPDATE0

NOTES

1. The update request bit SPEAnRUPi remains set, after it has been set to 1, until the update is completed. Afterwards it returns to 0 automatically.
2. All RLE definition buffer registers must not be modified while SPEAnRUPi = 1. Refer to Section 43.3.5, RLE definition registers modification for details.

43.6.1.7 SPEAnRjRBYP - RLE unit bypass mode register (D1M1A, D1M1-V2 only)

This register controls the bypass of the RLE unit j.

Access: This register can be accessed in 32-bit units.

Address: SPEAnRjRBYP (j = 0): <SPEAn_base> + 44_H
SPEAnRjRBYP (j = 1 to 3): <SPEAn_base> + 1044_H + ((j-1) x 100_H)

Initial value: for j = 0: $0000\ 0000_H$
for j = 1 to 3: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPEAnRjRBPROT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SPEAn RjRBYP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Note 1. 0 for j = 0, 1 for j = 1 to 3

Table 43.20 SPEAnRjRBYP register contents

Bit position	Bit name	Function
31 to 16	SPEAnRjRB PROT[15:0]	Write protection 5963_H: Write to SPEAnRjRBYP bit is allowed. Others: Write to SPEAnRjRBYP bit is ignored. During write to the SPEAnRjRBYP bit, 5963_H must be written these bits. Reading these bits returns is always 0000_H.
15 to 1	Reserved	When read, the value after reset is returned. When written, write the value after reset.
0	SPEAn RjRBYP	Bypass mode selection 0: RLE unit j is available 1: RLE unit j is bypassed

NOTE

This register must not be changed while VDCE and VOWE accesses are ongoing.

43.6.1.8 SPEAnRCFG/SPEAnRjRCFG - RLE prefetch configuration register

This register controls the size and timing for the RLE data stream prefetching.

Access: This register can be accessed in 32-bit units.

Address: SPEAnRCFG: <SPEAn_base> + 48_H
SPEAnRjRCFG (j = 0): <SPEAn_base> + 48_H
SPEAnRjRCFG (j = 1 to 3): <SPEAn_base> + 1048_H + ((j-1) x 100_H)

Initial value: 0000 0044_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	SPEAnRLEN[2:0]			0	SPEAnRDTH[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 43.21 SPEAnRCFG register contents

Bit position	Bit name	Function
31 to 7	Reserved	When read, the value after reset is returned. When written, write the value after reset.
6 to 4	SPEAn RLEN[2:0]	Burst size of the RLE Engine next data prefetching. RLE Engine prefetches next data every burst length x 64 bit. 0: Burst length = 1 1: Burst length = 2 2: Burst length = 4 3: Burst length = 6 4: Burst length = 8 (default) 5: Burst length = 10 6: Burst length = 12 7: Burst length = 14
3	Reserved	When read, the value after reset is returned. When written, write the value after reset.
2 to 0	SPEAn RDTH[2:0]	Prefetch timing. RLE Engine prefetches next data when remaining amount of the data in the RLE FIFO is less than threshold x 64 bit. 0: setting prohibited 1: threshold = 2 2: threshold = 4 3: threshold = 6 4: threshold = 8 (default) 5: threshold = 10 6: threshold = 12 7: threshold = 14

NOTE

The burst size SPEAnRLEN[2:0] and the prefetch timing SPEAnRDTH[2:0] must be set to values, that prevent the RLE data FIFO from overflow.
The RLE data FIFO size is 1024 bit.

43.6.2 Sprite Units registers

43.6.2.1 SPEAnSkEN - Sprite Unit k enable register

Each sprite of Sprite Unit k can be enabled by this register.
The enabled/disabled status of the sprites on Sprite Unit k can be checked by reading this register.

Disabling a sprite is done via the Sprite Unit k disable register SPEAnSkDS.

Access: This register can be accessed in 32-bit units.

Address: <SPEAn_base> + 100_H + k x 10_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPEAn SkEN15	SPEAn SkEN14	SPEAn SkEN13	SPEAn SkEN12	SPEAn SkEN11	SPEAn SkEN10	SPEAn SkEN9	SPEAn SkEN8	SPEAn SkEN7	SPEAn SkEN6	SPEAn SkEN5	SPEAn SkEN4	SPEAn SkEN3	SPEAn SkEN2	SPEAn SkEN1	SPEAn SkEN0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.22 SPEAnSkEN register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, write the value after reset.
15 to 0	SPEAn SkENm	Register write: Sprite m enable 0: at write: no function 1: at write: enable sprite m Register read: Sprite m status 0: at read: sprite m is disabled 1: at read: sprite m is enabled

43.6.2.2 SPEAnSkDS - Sprite Unit k disable register

Each sprite of Sprite Unit k can be disabled by this register.

Enabling a sprite is done via the Sprite Unit k enable register SPEAnSkEN.

Access: This register can be accessed in 32-bit units.

Address: <SPEAn_base> + 104_H + k x 10_H

Initial value: Reading this register returns an undefined value.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPEAn SkDS15	SPEAn SkDS14	SPEAn SkDS13	SPEAn SkDS12	SPEAn SkDS11	SPEAn SkDS10	SPEAn SkDS9	SPEAn SkDS8	SPEAn SkDS7	SPEAn SkDS6	SPEAn SkDS5	SPEAn SkDS4	SPEAn SkDS3	SPEAn SkDS2	SPEAn SkDS1	SPEAn SkDS0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 43.23 SPEAnSkDS register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, write the value after reset.
15 to 0	SPEAn SkDSm	Sprite m disable 0: no function 1: disable sprite m Note that reading of this register returns an undefined value. The enable/disable status of each sprite can be evaluated by reading the Sprite Unit k enable register SPEAnSkEN.

43.6.2.3 SPEAnSkUP - Sprite Unit k update request register

This register controls the update of the Sprite Unit k registers.

Access: This register can be accessed in 32-bit units.

Address: <SPEAn_base> + 108_H + k x 10_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SPEAn SkUP1	SPEAn SkUP0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 43.24 SPEAnSkUP register contents

Bit position	Bit name	Function
31 to 2	Reserved	When read, the value after reset is returned. When written, write the value after reset.
1	SPEAn SkUP1	Update request of all sprite definition registers of Sprite Unit k, which are assigned to VUPDATE1 0: no function 1: update VUPDATE1 assigned registers with next VUPDATE1
0	SPEAn SkUP0	Update request of all sprite definition registers of Sprite Unit k, which are assigned to VUPDATE0 0: no function 1: update VUPDATE0 assigned registers with next VUPDATE0

NOTES

1. The update request bit SPEAnSkUP0/SPEAnSkUP1 remains set, after it has been set to 1, until the update is completed. Afterwards it returns to 0 automatically.
2. All sprite definition buffer registers must not be modified while SPEAnSkUP0 = 1 or SPEAnSkUP1 = 1.
   Refer to Section 43.4.7, Sprite definition registers modification for details.
3. The sprite definition registers are assigned to VUPDATE0 or VUPDATE1 by the SPEAnSkVDm register.

43.6.2.4 SPEAnSkBYP - Sprite unit bypass mode register (D1M1A, D1M1-V2 only)

This register controls the bypass of the Sprite unit k.

Access: This register can be accessed in 32-bit units.

Address: <SPEAn_base> + $10C_H$ + k x 10_H

Initial value: for k = 0 to 2: $0000\ 0000_H$
for k = 3: $0000\ 0001_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPEAnSkBPROT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SPEAnSkBYP
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1*1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Note 1. 0 for k = 0 to 2; 1 for k = 3

Table 43.25 SPEAnSkBYP register contents

Bit position	Bit name	Function
31 to 16	SPEAnSkB PROT[15:0]	Write protection 5963_H: Write to SPEAnRjBYP bit is allowed. Others: Write to SPEAnRjBYP bit is ignored. During write to the SPEAnRjBYP bit, 5963_H must be written these bits. Reading these bits returns is always 0000_H.
15 to 1	Reserved	When read, the value after reset is returned. When written, write the value after reset.
0	SPEAn SkBYP	Bypass mode selection 0: Sprite unit k is available 1: Sprite unit k is bypassed

NOTE

This register must not be changed while VDCE and VOWE accesses are ongoing.

43.6.2.5 SPEAnSkDAm - Sprite Unit k sprite m destination address register

This register defines the address of the 1st pixel's color data in the memory of the sprite m of the Sprite Unit k.

Access: This register can be accessed in 32-bit units.

Address: SPEAnSkDAm (k = 0 to 2): <SPEAn_base> + 400_H + k x 400_H + m x 20_H
SPEAnSkDAm (k = 3): <SPEAn_base> + 1400_H + m x 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SPEAnSkDAm[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SPEAnSkDAm[15:3]													0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 43.26 SPEAnSkDAm register contents

Bit position	Bit name	Function
31 to 3	SPEAn SkDAm[31:3]	Sprite Unit k sprite m color data address The address must be 64-bit aligned, so the lower 3 bits SPEAnSkDAm[2:0] of the address are always 000_B.
2 to 0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

43.6.2.6 SPEAnSkVDm - Sprite Unit k sprite m VUPDATEn selection register

This register selects the VUPDATE signal for updating the Sprite Unit k sprite m definition registers.

Access: This register can be accessed in 32-bit units.

Address: SPEAnSkVDm (k = 0 to 2): <SPEAn_base> + 404_H + k x 400_H + m x 20_H
SPEAnSkVDm (k = 3): <SPEAn_base> + 1404_H + m x 20_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SPEAn SkVDm
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 43.27 SPEAnSkVDm register contents

Bit position	Bit name	Function
31 to 1	Reserved	When read, the value after reset is returned. When written, write the value after reset.
0	SPEAn SkVDm	VUPDATEn selection for Sprite Unit k sprite m definition registers update 0: VUPDATE0 updates registers 1: VUPDATE1 updates registers

43.6.2.7 SPEAnSkLYm - Sprite Unit k sprite m width/height register

This register defines the width and height of the Sprite Unit k sprite m.

Access: This register can be accessed in 32-bit units.

Address: SPEAnSkLYm (k = 0 to 2): <SPEAn_base> + 408_H + k x 400_H + m x 20_H
SPEAnSkLYm (k = 3): <SPEAn_base> + 1408_H + m x 20_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	SPEAn SkLYWm[9:0]										0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	SPEAn SkLYHm[10:0]										
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.28 SPEAnSkLYm register contents

Bit position	Bit name	Function
31 to 27	Reserved	When read, the value after reset is returned. When written, write the value after reset.
26 to 17	SPEAn SkLYWm[9:0]	Sprite Unit k sprite m width The width is defined as multiple of 64 bit and thus depends on the sprite pixel color format. • 8 bpp color formats: 64 bit / 8 = 8 pixel units – possible widths: multiple of 8 = 0, 8, 16, 32, 64, ... – SPEAnSkLYW[9:0] = width in pixel / 8 • 18/24/32 bpp color formats: 64 bit / 32 = 2 pixel units – possible X widths: multiple of 2 = 0, 2, 4, 6, 8, ... – SPEAnSkLYW[9:0] = width in pixel / 2
16	Bit16	This bit must always be written with 0.
15 to 11	Reserved	When read, the value after reset is returned. When written, write the value after reset.
10 to 0	SPEAn SkLYHm[10:0]	Sprite Unit k sprite m layer height in pixels

43.6.2.8 SPEAnSkPSm - Sprite Unit k sprite m X/Y position register

This register defines the position of the Sprite Unit k sprite m.

Access: This register can be accessed in 32-bit units.

Address: SPEAnSkPSm (k = 0 to 2): <SPEAn_base> + $40C_H$ + k x 400_H + m x 20_H
SPEAnSkPSm (k = 3): <SPEAn_base> + $140C_H$ + m x 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	SPEAn SkPSXm[9:0]										0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	SPEAn SkPSYm[12:0]												
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 43.29 SPEAnSkPSm register contents

Bit position	Bit name	Function
31 to 27	Reserved	When read, the value after reset is returned. When written, write the value after reset.
26 to 17	SPEAn SkPSXm[9:0]	Sprite Unit k sprite m X position The X position is defined as multiple of 64 bit and thus depends on the sprite color format. • 8 bpp color formats: 64 bit / 8 = 8 pixel units – possible X positions at a multiple of 8 = 0, 8, 16, 32, 64 – SPEAnSkPSX[9:0] = X position in pixel / 8 • 18/24/32 bpp color formats: 64 bit / 32 = 2 pixel units – possible X positions at a multiple of 2 = 0, 2, 4, 6, 8 – SPEAnSkPSX[9:0] = X position in pixel / 2
16	Bit16	This bit must always be written with 0.
15 to 13	Reserved	When read, the value after reset is returned. When written, write the value after reset.
12 to 0	SPEAn SkPSYm[12:0]	Sprite Unit k sprite m Y position in pixels

Section 44 JPEG Codec Unit A (JCUA)

This section contains a generic description of the JPEG Codec Unit A.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

44.1 Overview of the RH850/D1L/D1M JPEG Codec Unit A

44.1.1 Units

This microcontroller has the following number of units of the JPEG Codec Units A.

Table 44.1 Units

JPEG Codec Unit A	D1L1	D1L2(H)	D1M1(H) D1M1-V2 D1M1A	D1M2(H)
Units	0	0	1	1
Names	–	–	JCUA0	JCUA0

Unit index n

Throughout this section, the individual units of the JPEG Codec Units A are identified by the index “n” (n = 0), for example JCUAnJCMOD for the JCUAn mode register.

44.1.2 Register addresses

All JPEG Codec Units A register addresses are given as address offsets from the individual base addresses <JCUAn_base>.
The <JCUAn_base> addresses of each JCUAn are listed in the following table:

Table 44.2 Register base addresses <JCUAn_base>

JCUAn unit	<JCUAn_base> address
JCUA0	F200 $C000_H$

44.1.3 Clock supply

All JPEG Codec Units A provide one clock input.

Table 44.3 Clock supply

JCUAn unit	JCUAn clock	Connected to
JCUA0	Cross-connect clock XCCLK	Clock Controller C_ISO_XCCLK
	Register access clock PCLK	Clock Controller C_ISO_XCCLK/2

44.1.4 Interrupts

The JPEG Codec Units A can generate the following interrupt requests:

Table 44.4 JCUAn interrupt requests

JCUA0 signals	Function	Connected to
JEDI	Compression/decompression process interrupt request	Interrupt Controller INTJCUA0EDI
JDTI	Data transfer interrupt request	Interrupt Controller INTJCUA0DTI

44.1.5 Reset sources

The JPEG Codec Unit As and their registers are initialized by the following reset signal:

Table 44.5 Reset sources

JCUAn unit	Reset signal
JCUA0	• Reset Controller SYSRES • Reset Controller JCUA0RES • reset upon wake-up from DEEPSTOP mode

44.1.6 Bus master ID

The JPEG Codec Unit bus master interface is connected to the cross-connect system. The master interface has the following master ID:

JCUA: MSTID5

44.2 Features

The JPEG codec unit (JCUA) incorporates a JPEG codec conforming to the JPEG baseline compression and decompression standard to provide high-speed compression of image data and high-speed decoding of JPEG data.

The JPEG codec unit has the following features:

- Conforms to the JPEG baseline standard within the range described in this document.
  This module does not support the following basic features:
  Scanning with two elements
  Non-interleave scanning with multiple elements
- Operational precision: Conforming to JPEG Part 2, ISO-IEC10918-2
- Image input/output system: Block interleave method
- Pixel format:
  - Compression: YCbCr422 (H = 2:1:1, V = 1:1:1)
  - Decompression: YCbCr444 (H = 1:1:1, V = 1:1:1), YCbCr422 (H = 2:1:1, V = 1:1:1), YCbCr411 (H = 4:1:1, V = 1:1:1), YCbCr420 (H = 2:1:1, V = 2:1:1)
  - Output pixel format to the buffer: YCbCr422, ARGB8888, RGB565
- Four quantization tables provided
- Four Huffman tables provided (two tables for AC coefficients and two tables for DC coefficients)
- Markers supported: SOI (start of image), SOF0 (start of frame type 0), SOS (start of scan), DQT (define quantization tables), DHT (define Huffman tables), DRI (define restart interval), RSTm (restart marks), and EOI (end of image)
- The buffer size can be reduced by using the mode in which data transfer is temporarily stopped each time the specified number of lines or the specified amount of data is transferred during image data or coded data input/output.
- Processing unit: 8-byte address boundary units can be set
- Image sizes that can be processed: Sizes divisible by the minimum coded unit (MCU): 8 lines by 8 pixels in YCbCr444; 8 lines by 16 pixels in YCbCr422; 8 lines by 32 pixels in YCbCr411; 16 lines by 16 pixels in YCbCr420

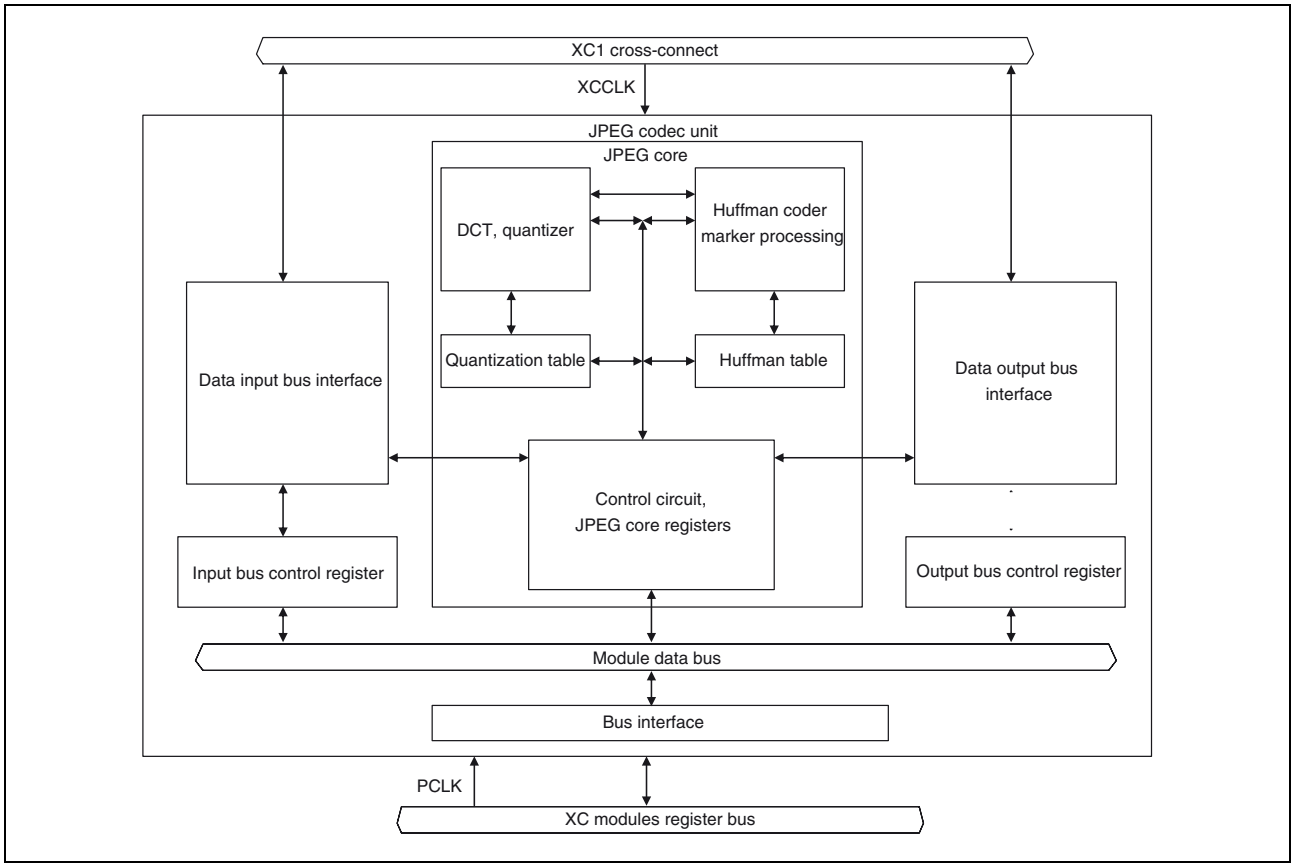


Figure 44.1 Block Diagram

44.3 Register Descriptions

The following table shows the JCUA registers.

Table 44.6 Register Configuration (1/2)

Module Name	Register Name	Symbol	R/W	Address	Access Size
JCUAn	JPEG code mode register	JCMOD	R/W	<JCUAn_base> + 000_H	8
JCUAn	JPEG code command register	JCCMD	R/W	<JCUAn_base> + 001_H	8
JCUAn	JPEG code quantization table number register	JCQTN	R/W	<JCUAn_base> + 003_H	8
JCUAn	JPEG code Huffman table number register	JCHTN	R/W	<JCUAn_base> + 004_H	8
JCUAn	JPEG code DRI upper register	JCDRIU	R/W	<JCUAn_base> + 005_H	8
JCUAn	JPEG code DRI lower register	JCDRID	R/W	<JCUAn_base> + 006_H	8
JCUAn	JPEG code vertical size upper register	JCVSZU	R/W	<JCUAn_base> + 007_H	8
JCUAn	JPEG code vertical size lower register	JCVSZD	R/W	<JCUAn_base> + 008_H	8
JCUAn	JPEG code horizontal size upper register	JCHSZU	R/W	<JCUAn_base> + 009_H	8
JCUAn	JPEG code horizontal size lower register	JCHSZD	R/W	<JCUAn_base> + $00A_H$	8
JCUAn	JPEG code data count upper register	JCDTCU	R	<JCUAn_base> + $00B_H$	8
JCUAn	JPEG code data count middle register	JCDTCM	R	<JCUAn_base> + $00C_H$	8
JCUAn	JPEG code data count lower register	JCDTCD	R	<JCUAn_base> + $00D_H$	8
JCUAn	JPEG interrupt enable register 0	JINTE0	R/W	<JCUAn_base> + $00E_H$	8
JCUAn	JPEG interrupt status register 0	JINTS0	R/W	<JCUAn_base> + $00F_H$	8
JCUAn	JPEG code decode error register	JCDERR	R/W	<JCUAn_base> + 010_H	8
JCUAn	JPEG code reset register	JCRST	R	<JCUAn_base> + 011_H	8
JCUAn	JPEG interface compression control register	JIFECNT	R/W	<JCUAn_base> + 040_H	32
JCUAn	JPEG interface compression source address register	JIFESA	R/W	<JCUAn_base> + 044_H	32
JCUAn	JPEG interface compression line offset register	JIFESOFST	R/W	<JCUAn_base> + 048_H	32
JCUAn	JPEG interface compression destination address register	JIFEDA	R/W	<JCUAn_base> + $04C_H$	32
JCUAn	JPEG interface compression source line count register	JIFESLC	R/W	<JCUAn_base> + 050_H	32
JCUAn	JPEG interface compression destination count register	JIFEDDC	R/W	<JCUAn_base> + 054_H	32
JCUAn	JPEG interface decompression control register	JIFDCNT	R/W	<JCUAn_base> + 058_H	32
JCUAn	JPEG interface decompression source address register	JIFDSA	R/W	<JCUAn_base> + $05C_H$	32
JCUAn	JPEG interface decompression line offset register	JIFDDOFST	R/W	<JCUAn_base> + 060_H	32
JCUAn	JPEG interface decompression destination address register	JIFDDA	R/W	<JCUAn_base> + 064_H	32
JCUAn	JPEG interface decompression source data count register	JIFDSDC	R/W	<JCUAn_base> + 068_H	32
JCUAn	JPEG interface decompression destination line count register	JIFDDLC	R/W	<JCUAn_base> + $06C_H$	32
JCUAn	JPEG interface decompression α set register	JIFDADT	R/W	<JCUAn_base> + 070_H	32
JCUAn	JPEG interrupt enable register 1	JINTE1	R/W	<JCUAn_base> + $08C_H$	32
JCUAn	JPEG interrupt status register 1	JINTS1	R/W	<JCUAn_base> + 090_H	32

Table 44.6 Register Configuration (2/2)

Module Name	Register Name	Symbol	R/W	Address	Access Size
JCUAn	JPEG input image data CbCr range setting register	JIFESVSZ	R/W	<JCUAn_base> + 094_H	32
JCUAn	JPEG output image data CbCr range setting register	JIFESHSZ	R/W	<JCUAn_base> + 098_H	32
JCUAn	JPEG code quantization table 0 register	JCQTBL0	R/W	<JCUAn_base> + 100_H to <JCUAn_base> + $13F_H$	8
JCUAn	JPEG code quantization table 1 register	JCQTBL1	R/W	<JCUAn_base> + 140_H to <JCUAn_base> + $17F_H$	8
JCUAn	JPEG code quantization table 2 register	JCQTBL2	R/W	<JCUAn_base> + 180_H to <JCUAn_base> + $1BF_H$	8
JCUAn	JPEG code quantization table 3 register	JCQTBL3	R/W	<JCUAn_base> + $1C0_H$ to <JCUAn_base> + $1FF_H$	8
JCUAn	JPEG code Huffman table DC0 register	JCHTBD0	W	<JCUAn_base> + 200_H to <JCUAn_base> + $21B_H$	8
JCUAn	JPEG code Huffman table AC0 register	JCHTBA0	W	<JCUAn_base> + 220_H to <JCUAn_base> + $2D1_H$	8
JCUAn	JPEG code Huffman table DC1 register	JCHTBD1	W	<JCUAn_base> + 300_H to <JCUAn_base> + $31B_H$	8
JCUAn	JPEG code Huffman table AC1 register	JCHTBA1	W	<JCUAn_base> + 320_H to <JCUAn_base> + $3D1_H$	8
SELB	JPEG Codec Unit A Software reset register	JCSWRST	R/W	FFC0 6010_H	32

NOTES

1. Register access sizes other than defined in the table above are not supported.
2. For the settings of the JPEG code quantization table and JPEG code Huffman table, see Section 44.4.1.4, Table Setting.

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

44.3.1 JPEG Code Mode Register (JCMOD)

JCMOD sets the operating mode before the JCUA starts operation.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	–	–	–	–	DSP	REDU[2:0]		
Compress	R	R	R	R	R/W	R/W	R/W	R/W
Decompress	R	R	R	R	R/W	R	R	R

Table 44.7 JCMOD register contents

Bit position	Bit name	Function
7 to 4	Reserved	When read, the value after reset is returned. When written, write the value after reset.
3	DSP	Compression/Decompression Set 0: Compression process 1: Decompression process Note: When changing between processing for compression and for decompression, be sure to reset this module in advance by setting the JCUA0RES bit in the JCSWRST.
2 to 0	REDU[2:0]	Pixel Format [Compression] 001: YCbCr422 Other than above: Setting prohibited. [Decompression] 000: YCbCr444 001: YCbCr422 010: YCbCr420 110: YCbCr411 Other than above: Error (JCUA cannot process normally.)

44.3.2 JPEG Code Command Register (JCCMD)

JCCMD sets commands. Bits of this register need not be cleared to 0 after setting a command. Multiple commands must not be set simultaneously.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	–	–	–	–	–	JEND	JRST	JSRT
Compress	R	R	R	R	R	R*1/W	undefined	R*1/W
Decompress	R	R	R	R	R	R*1/W	R*1/W	R*1/W

Note 1. Values read from these bits are undefined.

Table 44.8 JCCMD register contents

Bit position	Bit name	Function
7 to 3	Reserved	When read, the value after reset is returned. When written, write the value after reset.
2	JEND	Interrupt Request Clear Command This bit is valid only for the interrupt sources corresponding to bits INS6, INS5, and INS3 in JINTS0. To clear an interrupt request, set this bit to 1.
1	JRST	JPEG Core Process Stop Clear Command To clear the process-stopped state caused by requests to read the image size and pixel format (enabled by the INT3 bit in JINTE0), set this bit to 1.
0	JSRT	JPEG Core Process Start Command To start JPEG core processing, set this bit to 1. Do not write this bit to 1 again while the JCUA is in operation.

44.3.3 JPEG Code Quantization Table Number Register (JCQTN)

JCQTN sets the quantization table number before compression process is started.

- To use quantization table No. 0 (JCQTBL0) as the first color component, set QT1[1:0] to B'00
- To use quantization table No. 1 (JCQTBL1) as the first color component, set QT1[1:0] to B'01
- To use quantization table No. 2 (JCQTBL2) as the first color component, set QT1[1:0] to B'10
- To use quantization table No. 3 (JCQTBL3) as the first color component, set QT1[1:0] to B'11

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	–	–	QT3[1:0]		QT2[1:0]		QT1[1:0]	
Compress	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	R	R	R	R	R	R	R	R

Table 44.9 JCQTN register contents

Bit position	Bit name	Function
7 to 6	Reserved	When read, the value after reset is returned.
5 to 4	QT3[1:0]	Quantization table number for the third color component
3 to 2	QT2[1:0]	Quantization table number for the second color component
1 to 0	QT1[1:0]	Quantization table number for the first color component

44.3.4 JPEG Code Huffman Table Number Register (JCHTN)

JCHTN sets the Huffman table number (AC/DC) before compression process is started.

- To use DC/AC Huffman table No. 0 (JCHTBD0 and JCHTBA0) as the first color component, set bits HTA1 and HTD1 to B'0
- To use DC/AC Huffman table No. 1 (JCHTBD1 and JCHTBA1) as the first color component, set bits HTA1 and HTD1 to B'1

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	–	–	HTA3	HTD3	HTA2	HTD2	HTA1	HTD1
Compress	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	R	R	R	R	R	R	R	R

Table 44.10 JCHTN register contents

Bit position	Bit name	Function
7 to 6	Reserved	When read, the value after reset is returned. When written, write the value after reset.
5	HTA3	Huffman table number (AC) for the third color component
4	HTD3	Huffman table number (DC) for the third color component
3	HTA2	Huffman table number (AC) for the second color component
2	HTD2	Huffman table number (DC) for the second color component
1	HTA1	Huffman table number (AC) for the first color component
0	HTD1	Huffman table number (DC) for the first color component

44.3.5 JPEG Code DRI Upper Register (JCDRIU)

JCDRIU sets the upper bytes of the minimum coded units (MCUs) preceding an RST marker.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	DRIU[7:0]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Table 44.11 JCDRIU register contents

Bit position	Bit name	Function
7 to 0	DRIU[7:0]	Upper Bytes of MCUs Preceding RST Marker When both upper and lower bytes are set to H'00, neither a DRI nor an RST marker is placed.

44.3.6 JPEG Code DRI Lower Register (JCDRID)

JCDRID sets the lower bytes of MCUs preceding an RST marker.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	DRID[7:0]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Table 44.12 JCDRID register contents

Bit position	Bit name	Function
7 to 0	DRID[7:0]	Lower Bytes of MCUs Preceding RST Marker When both upper and lower bytes are set to H'00, neither a DRI nor an RST marker is placed.

44.3.7 JPEG Code Vertical Size Upper Register (JCVSZU)

JCVSZU sets the upper bytes of the vertical image size.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	VSZU[7:0]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	R	R	R	R	R	R	R	R

Table 44.13 JCVSZU register contents

Bit position	Bit name	Function
7 to 0	VSZU[7:0]	Upper Bytes of Vertical Image Size In decompression process, a downloaded value from the JPEG coded data is set.

44.3.8 JPEG Code Vertical Size Lower Register (JCVSZD)

JCVSZD sets the lower bytes of the vertical image size.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	VSZD[7:0]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	R	R	R	R	R	R	R	R

Table 44.14 JCVSZD register contents

Bit position	Bit name	Function
7 to 0	VSZD[7:0]	Lower Bytes of Vertical Image Size In decompression process, a downloaded value from the JPEG coded data is set.

44.3.9 JPEG Code Horizontal Size Upper Register (JCHSZU)

JCHSZU sets the upper bytes of the horizontal image size.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	HSZU[7:0]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	R	R	R	R	R	R	R	R

Table 44.15 JCHSZU register contents

Bit position	Bit name	Function
7 to 0	HSZU[7:0]	Upper Bytes of Horizontal Image Size In decompression process, a downloaded value from the JPEG coded data is set.

44.3.10 JPEG Coded Horizontal Size Lower Register (JCHSZD)

JCHSZD sets the lower bytes of the horizontal image size.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	HSZD[7:0]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	R	R	R	R	R	R	R	R

Table 44.16 JCHSZD register contents

Bit position	Bit name	Function
7 to 0	HSZD[7:0]	Lower Bytes of Horizontal Image Size In decompression process, a downloaded value from the JPEG coded data is set.

44.3.11 JPEG Code Data Count Upper Register (JCDTCU)

The upper bytes for the counted amount of data to be compressed are set to JCDTCU. The values of this register are reset before compression starts.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	DCU[7:0]							
Compress	R	R	R	R	R	R	R	R
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Table 44.17 JCDTCU register contents

Bit position	Bit name	Function
7 to 0	DCU[7:0]	Upper bytes of the counted amount of data to be compressed

44.3.12 JPEG Code Data Count Middle Register (JCDTCM)

The middle bytes for the counted amount of data to be compressed are set to JCDTCM. The values of this register are reset before compression starts.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	DCM[7:0]							
Compress	R	R	R	R	R	R	R	R
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Table 44.18 JCDTCM register contents

Bit position	Bit name	Function
7 to 0	DCM[7:0]	Middle bytes of the counted amount of data to be compressed

44.3.13 JPEG Code Data Count Lower Register (JCDTCD)

The lower bytes for the counted amount of data to be compressed are set to JCDTCD. The values of this register are reset before compression starts.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	DCD[7:0]							
Compress	R	R	R	R	R	R	R	R
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Table 44.19 JCDTCD register contents

Bit position	Bit name	Function
7 to 0	DCD[7:0]	Lower bytes of the counted amount of data to be compressed

44.3.14 JPEG Interrupt Enable Register 0 (JINTE0)

JINTE0 enables interrupts.

When any of bits INT7 to INT5 is set to B'1, the INS5 bit in JINTS0 indicates B'1 as the error status upon occurrence of the compression data error, and the ERR[3:0] bit in JCDERR indicates the particular error code.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	INT7	INT6	INT5	–	INT3	–	–	–
Compress	undefined	undefined	undefined	R	undefined	R	R	R
Decompress	R/W	R/W	R/W	R	R/W	R	R	R

Table 44.20 JINTE0 register contents

Bit position	Bit name	Function
7	INT7	This bit enables an interrupt to be generated when the number of data in the restart interval of the Huffman-coding segment is not correct in decompression. When this bit is not set to enable interrupt generation, an error code is not returned. Block data number error is enable by this bit. For the error code detail, please refer Table 44.44, Segment Error Codes.
6	INT6	This bit enables an interrupt to be generated when the total number of data in the Huffman-coding segment is not correct in decompression. When this bit is not set to enable interrupt generation, an error code is not returned. Block data number error is enable by this bit. For the error code detail, please refer Table 44.44, Segment Error Codes.
5	INT5	This bit enables an interrupt to be generated when the final number of MCU data in the Huffman-coding segment is not correct in decompression. When this bit is not set to enable interrupt generation, an error code is not returned. Block data number error is enable by this bit. For the error code detail, please refer Table 44.44, Segment Error Codes.
4	Reserved	When read, the value after reset is returned. When written, write the value after reset.
3	INT3	This bit enables an interrupt to be generated when it has been determined that the image size and the subsampling setting of the compressed data can be read through analyzing the data.
2 to 0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

44.3.15 JPEG Interrupt Status Register 0 (JINTS0)

JINTS0 identifies the interrupt sources.

The interrupt sources of this register should be cleared by clearing the corresponding interrupt status bits to 0 and setting the relevant bit in JCCMD appropriately.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	–	INS6	INS5	–	INS3	–	–	–
Compress	R	R/W*1	undefined	R	undefined	R	R	R
Decompress	R	R/W*1	R/W*1	R	R/W*1	R	R	R

Note 1. Clear this bit by writing 0 to it. Do not write 1 to this bit.

Table 44.21 JINTS0 register contents

Bit position	Bit name	Function
7	Reserved	When read, the value after reset is returned. When written, write the value after reset.
6	INS6	This bit is set to 1 when the JCUA completes compression process normally.
5	INS5	This bit is set to 1 when a compressed data error occurs.
4	Reserved	When read, the value after reset is returned. When written, write the value after reset.
3	INS3	This bit is set to 1 when the image size and pixel format can be read. When an interrupt occurs, this module stops processing and the state is indicated by the JCRST register. To make the JCUA resume processing, set the JPEG core process stop clear command bit (JRST) in JCCMD.
2 to 0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

44.3.16 JPEG Code Decode Error Register (JCDERR)

JCDERR indicates the error code to identify the type of the error which has occurred in the compressed data analysis for decompression. The values of this register are reset before the JCUA starts decompression.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: $0A_H$

	7	6	5	4	3	2	1	0
	–	–	–	–	ERR[3:0]			
Compress	R	R	R	R	undefined	undefined	undefined	undefined
Decompress	R	R	R	R	R/W	R/W	R/W	R/W

Table 44.22 JINTE0 register contents

Bit position	Bit name	Function
7 to 4	Reserved	When read, the value after reset is returned. When written, write the value after reset.
3 to 0	ERR[3:0]	Error Code (See Table 44.43, Decompression Error Codes and Table 44.44, Segment Error Codes)

44.3.17 JPEG Code Reset Register (JCRST)

JCRST indicates a processing-stopped state caused by requests to read the image size and pixel format (enabled by the INT3 bit in JINTE0). To resume processing, set the JPEG core process stop clear command bit (JRST) in JCCMD.

Access: This register can be accessed in 8-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 00_H

	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	RST
Compress	R	R	R	R	R	R	R	undefined
Decompress	R	R	R	R	R	R	R	R

Table 44.23 JCRST register contents

Bit position	Bit name	Function
7 to 1	Reserved	When read, the value after reset is returned.
0	RST	Operating State 0: State other than below 1: Suspended state caused by interrupt sources of JINTE0

44.3.18 JPEG Interface Compression Control Register (JIFECNT)

JIFECNT controls the compression process.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 0000 0000_H

	31	30	29	28	27	26	25	24
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	–	JOUTRINI	JOUTRCMD	JOUTC	–	JOUTSWAP[2:0]		
Compress	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Decompress	R	undefined	undefined	undefined	R	undefined	undefined	undefined

	7	6	5	4	3	2	1	0
	–	DINRINI	DINRCMD	DINLC	–	DINSWAP[2:0]		
Compress	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Decompress	R	undefined	undefined	undefined	R	undefined	undefined	undefined

Table 44.24 JIFECNT register contents (1/2)

Bit position	Bit name	Function
31 to 15	Reserved	When read, the value after reset is returned. When written, write the value after reset.
14	JOUTRINI	Address Initialization when Output Coded Data is Resumed This bit is only valid when the count mode for stopping the output of coded data is on. Set this bit before writing 1 to the data resume command bit. 0: The transfer address is not initialized when the output of coded data is restarted. 1: The transfer address is initialized when the output of coded data is restarted.
13	JOUTRCMD	Output Coded Data Resume Command This bit is only valid when the count mode for stopping the output of coded data is on. Setting this bit to 1 resumes writing output coded data. This bit is always read as 0.
12	JOUTC	Count Mode Setting for Stopping Output Coded Data 0: Count mode for stopping the output of coded data is off. 1: Count mode for stopping the output of coded data is on.
11	Reserved	When read, the value after reset is returned. When written, write the value after reset.

Table 44.24 JIFECNT register contents (2/2)

Bit position	Bit name	Function
10 to 8	JOUTSWAP [2:0]	Byte/Word/Longword Swap Output coded data in compression is swapped. 000: (1) (2) (3) (4) (5) (6) (7) (8) 001: (2) (1) (4) (3) (6) (5) (8) (7) [Byte swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Word swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Word - byte swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Longword swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Longword - byte swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Longword - word swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Longword - word - byte swap]
7	Reserved	When read, the value after reset is returned. When written, write the value after reset.
6	DINRINI	Address Initialization when Resuming Input of Image Data Lines This bit is only valid when the count mode for stopping the input of image data lines is on. Set this bit before writing 1 to the dataline resume command bit. 0: The transfer address is not initialized when the input of image data lines is restarted. 1: The transfer address is initialized when the input of image data lines is restarted.
5	DINRCMD	Input Image Data Lines Resume Command This bit is valid only when the count mode for stopping the input of image data lines is on. Setting this bit to 1 resumes reading input image data. This bit is always read as 0.
4	DINLC	Count Mode Setting for Stopping Input Image Data Lines 0: Count mode for stopping the input of image data lines is off. 1: Count mode for stopping the input of image data lines is on.
3	Reserved	When read, the value after reset is returned. When written, write the value after reset.
2 to 0	DINSWAP[2:0]	Byte/Word/Longword Swap Input image data in compression is swapped. 000: (1) (2) (3) (4) (5) (6) (7) (8) 001: (2) (1) (4) (3) (6) (5) (8) (7) [Byte swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Word swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Word - byte swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Longword swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Longword - byte swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Longword - word swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Longword - word - byte swap]

44.3.19 JPEG Interface Compression Source Address Register (JIFESA)

JIFESA sets the source address of the input image data. This register should be set in 8-byte units.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 0000 0000_H

	31	30	29	28	27	26	25	24
	ESA[31:24]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

	23	22	21	20	19	18	17	16
	ESA[23:16]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

	15	14	13	12	11	10	9	8
	ESA[15:8]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

	7	6	5	4	3	2	1	0
	ESA[7:0]							
Compress	R/W	R/W	R/W	R/W	R/W	R	R	R
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Table 44.25 JIFESA register contents

Bit position	Bit name	Function
31 to 0	ESA[31:0]	Input Image Data Source Address (in 8-byte units) The lower three bits should be set to 0.

44.3.20 JPEG Interface Compression Line Offset Register (JIFESOFST)

JIFESOFST sets the line offset of the input image data (refer to Section 44.4.4, Storing Image Data). This register should be set in 8-byte units.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 0000 0000_H

	31	30	29	28	27	26	25	24
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	–	ESMW[14:8]						
Compress	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	R	undefined	undefined	undefined	undefined	undefined	undefined	undefined

	7	6	5	4	3	2	1	0
	ESMW[7:0]							
Compress	R/W	R/W	R/W	R/W	R/W	R	R	R
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Table 44.26 JIFESOFST register contents

Bit position	Bit name	Function
31 to 15	Reserved	When read, the value after reset is returned. When written, write the value after reset.
14 to 0	ESMW[14:0]	Input Image Data Lines Offset (in 8-byte units) The lower three bits should be set to 0.

44.3.21 JPEG Interface Compression Destination Address Register (JIFEDA)

JIFEDA sets the destination address of the output coded data. This register should be set in 8-byte units.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 0000 0000_H

	31	30	29	28	27	26	25	24
	EDA[31:24]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

	23	22	21	20	19	18	17	16
	EDA[23:16]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

	15	14	13	12	11	10	9	8
	EDA[15:8]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

	7	6	5	4	3	2	1	0
	EDA[7:0]							
Compress	R/W	R/W	R/W	R/W	R/W	R	R	R
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Table 44.27 JIFEDA register contents

Bit position	Bit name	Function
31 to 0	EDA[31:0]	Output Coded Data Destination Address (in 8-byte units) The lower three bits should be set to 0.

44.3.22 JPEG Interface Compression Source Line Count Register (JIFESLC)

JIFESLC sets the number of input image data lines when the count mode for stopping the input of image data lines is on (the DINLC bit in JIFECNT is set to 1). This register should be set in 8-line units.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: FFF8 FFF8$_H$

	31	30	29	28	27	26	25	24
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	LINES[15:8]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

	7	6	5	4	3	2	1	0
	LINES[7:0]							
Compress	R/W	R/W	R/W	R/W	R/W	R	R	R
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Table 44.28 JIFESLC register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, write the value after reset.
15 to 0	LINES[15:0]	Input Image Data Lines Offset (in 8-line units) The lower three bits should be set to 0.

44.3.23 JPEG Interface Compression Destination Count Register (JIFEDDC)

JIFEDDC sets the amount of output coded data when the count mode for stopping the output of coded data is on (the JOUTC bit in JIFECNT is set to 1). This register should be set in 8-byte units.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: FFF8 FFF8$_H$

	31	30	29	28	27	26	25	24
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	JDATAS[15:8]							
Compress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

	7	6	5	4	3	2	1	0
	JDATAS[7:0]							
Compress	R/W	R/W	R/W	R/W	R/W	R	R	R
Decompress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined

Table 44.29 JIFEDDC register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, write the value after reset.
15 to 0	JDATAS[15:0]	Amount of Output Coded Data to be Written (in 8-byte units) The lower three bits should be set to 0.

44.3.24 JPEG Interface Decompression Control Register (JIFDCNT)

JIFDCNT controls the decompression process.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 0000 0000_H

	31	30	29	28	27	26	25	24
	–	–	VINTER[1:0]		HINTER[1:0]		OPF[1:0]	
Compress	R	R	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R	R	R/W	R/W	R/W	R/W	R/W	R/W

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	–	JINRINI	JINRCMD	JINC	–	JINSWAP[2:0]		
Compress	R	undefined	undefined	undefined	R	undefined	undefined	undefined
Decompress	R	R/W	R/W	R/W	R	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
	–	DOUTRINI	DOUTRCMD	DOUTLC	–	DOUTSWAP[2:0]		
Compress	R	undefined	undefined	undefined	R	undefined	undefined	undefined
Decompress	R	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 44.30 JIFDCNT register contents (1/2)

Bit position	Bit name	Function
31 to 30	Reserved	When read, the value after reset is returned. When written, write the value after reset.
29 to 28	VINTER[1:0]	Vertical Subsampling Subsamples vertical output image data. 00: No subsampling 01: Subsamples output data into 1/2. 10: Subsamples output data into 1/4. 11: Subsamples output data into 1/8.
27 to 26	HINTER[1:0]	Horizontal Subsampling Subsamples horizontal output image data. 00: No subsampling 01: Subsamples output data into 1/2. 10: Subsamples output data into 1/4. 11: Subsamples output data into 1/8.
25 to 24	OPF[1:0]	Specifies output image data pixel format. 00: YCbCr422 01: ARGB8888 10: RGB565 11: Setting prohibited
23 to 15	Reserved	When read, the value after reset is returned. When written, write the value after reset.

Table 44.30 JIFDCNT register contents (2/2)

Bit position	Bit name	Function
14	JINRINI	Address Initialization when Input Coded Data is Resumed This bit is only valid when the count mode for stopping the input of coded data is on. Set this bit before writing 1 to the data resume command bit. 0: The transfer address is not initialized when the input of coded data is restarted. 1: The transfer address is initialized when the input of coded data is restarted.
13	JINRCMD	Input Coded Data Resume Command This bit is valid only when the count mode for stopping the input of coded data is on. Setting this bit to 1 resumes reading input coded data. This bit is always read as 0.
12	JINC	Count Mode Setting for Stopping Input Coded Data 0: Count mode for stopping the input of coded data is off. 1: Count mode for stopping the input of coded data is on.
11	Reserved	When read, the value after reset is returned. When written, write the value after reset.
10 to 8	JINSWAP[2:0]	Byte/Word/Longword Swap Input coded data in decompression is swapped. 000: (1) (2) (3) (4) (5) (6) (7) (8) 001: (2) (1) (4) (3) (6) (5) (8) (7) [Byte swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Word swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Word - byte swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Longword swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Longword - byte swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Longword - word swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Longword - word - byte swap]
7	Reserved	When read, the value after reset is returned. When written, write the value after reset.
6	DOUTRINI	Address Initialization when Resuming Output of Image Data Lines This bit is only valid when the count mode for stopping the output of image data lines is on. Set this bit before writing 1 to the dataline resume command bit. 0: The transfer address is not initialized when the output of lines of image data is restarted. 1: The transfer address is initialized when the output of lines of image data is restarted.
5	DOUTRCMD	Output Image Data Lines Resume Command This bit is valid only when the count mode for stopping the output of image data lines is on. Setting this bit to 1 resumes writing image data. This bit is always read as 0.
4	DOUTLC	Count Mode for Stopping Output Image Data Lines 0: Count mode for stopping the output of image data lines is off. 1: Count mode for stopping the output of image data lines is on.
3	Reserved	When read, the value after reset is returned. When written, write the value after reset.
2 to 0	DOUTSWAP[2:0]	Byte/Word Swap Output image data in decompression is swapped. 000: (1) (2) (3) (4) (5) (6) (7) (8) 001: (2) (1) (4) (3) (6) (5) (8) (7) [Byte swap] 010: (3) (4) (1) (2) (7) (8) (5) (6) [Word swap] 011: (4) (3) (2) (1) (8) (7) (6) (5) [Word - byte swap] 100: (5) (6) (7) (8) (1) (2) (3) (4) [Longword swap] 101: (6) (5) (8) (7) (2) (1) (4) (3) [Longword - byte swap] 110: (7) (8) (5) (6) (3) (4) (1) (2) [Longword - word swap] 111: (8) (7) (6) (5) (4) (3) (2) (1) [Longword - word - byte swap]

44.3.25 JPEG Interface Decompression Source Address Register (JIFDSA)

JIFDSA sets the source address of the input coded data. This register should be set in 8-byte units.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 0000 0000_H

	31	30	29	28	27	26	25	24
	DSA[31:24]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	23	22	21	20	19	18	17	16
	DSA[23:16]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	15	14	13	12	11	10	9	8
	DSA[15:8]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
	DSA[7:0]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 44.31 JIFDSA register contents

Bit position	Bit name	Function
31 to 0	DSA[31:0]	Input Coded Data Source Address (in 8- byte units) The lower three bits should be set to 0.

44.3.26 JPEG Interface Decompression Line Offset Register (JIFDDOFST)

JIFDDOFST sets the line offset of the output image data to be transferred to the external buffer (refer to Section 44.4.4, Storing Image Data). This register should be set in 8-byte units.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: $0000\ 0000_H$

	31	30	29	28	27	26	25	24
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	–	DDMW[14:8]						
Compress	R	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
	DDMW[7:0]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 44.32 JIFDDOFST register contents

Bit position	Bit name	Function
31 to 15	Reserved	When read, the value after reset is returned. When written, write the value after reset.
14 to 0	DDMW[14:0]	Output Image Data Lines Offset (in 8-byte units) The lower three bits should be set to 0.

44.3.27 JPEG Interface Decompression Destination Address Register (JIFDDA)

JIFDDA sets the destination address of the output image data. This register should be set in 8-byte units.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 0000 0000_H

	31	30	29	28	27	26	25	24
	DDA[31:24]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	23	22	21	20	19	18	17	16
	DDA[23:16]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	15	14	13	12	11	10	9	8
	DDA[15:8]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
	DDA[7:0]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 44.33 (JIFDDA register contents

Bit position	Bit name	Function
31 to 0	DDA[31:0]	Output Image Data Destination Address (in 8-byte units) The lower three bits should be set to 0.

44.3.28 JPEG Interface Decompression Source Data Count Register (JIFDSDC)

JIFDSDC sets the amount of input coded data when the count mode for stopping the input of coded data is on (the JINC bit in JIFDCNT is set to 1). This register should be set in 8-byte units.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: FFF8 FFF8$_H$

	31	30	29	28	27	26	25	24
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	JDATAS[15:8]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
	JDATAS[7:0]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 44.34 JIFDSDC register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, write the value after reset.
15 to 0	JDATAS[15:0]	Amount of Input Coded Data to be Read (in 8-byte units) The lower three bits should be set to 0.

44.3.29 JPEG Interface Decompression Destination Line Count Register (JIFDDLC)

JIFDDLC sets the number of output image data lines when the count mode for stopping the output of image data lines is on (the DOUTLC bit in JIFDCNT is set to 1). This register should be set in 8-line units.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: FFF8 $FFF8_H$

	31	30	29	28	27	26	25	24
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	LINES[15:8]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

	7	6	5	4	3	2	1	0
	LINES[7:0]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 44.35 JIFDDLC register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, write the value after reset.
15 to 0	LINES[15:0]	Set this bit to the number of lines of output image data to write (in 8-line units). The lower three bits should be set to 0.

44.3.30 JPEG Interface Decompression α Set Register (JIFDADT)

JIFDADT is used to set the α value when output is in ARGB8888 format.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 0000 0000_H

	31	30	29	28	27	26	25	24
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	7	6	5	4	3	2	1	0
	ALPHA[7:0]							
Compress	undefined	undefined	undefined	undefined	undefined	undefined	undefined	undefined
Decompress	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.36 JIFDADT register contents

Bit position	Bit name	Function
31 to 8	Reserved	When read, the value after reset is returned. When written, write the value after reset.
7 to 0	ALPHA[7:0]	Setting of the α value for output in ARGB8888 format.

44.3.31 JPEG Interrupt Enable Register 1 (JINTE1)

JINTE1 enables interrupts.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 0000 0000_H

	31	30	29	28	27	26	25	24
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	7	6	5	4	3	2	1	0
	–	CBTEN	DINLEN	JOUTEN	–	DBTEN	JINEN	DOUTLEN
Compress	R	R/W	R/W	R/W	R	undefined	undefined	undefined
Decompress	R	undefined	undefined	undefined	R	R/W	R/W	R/W

Table 44.37 JINTE1 register contents (1/2)

Bit position	Bit name	Function
31 to 7	Reserved	When read, the value after reset is returned. When written, write the value after reset.
6	CBTEN	Enables or disables a data transfer processing interrupt request (JDTI) when the CBTF bit in JINTS1 is set to 1. 0: Disables an interrupt request. 1: Enables an interrupt request.
5	DINLEN	Enables or disables a data transfer processing interrupt request (JDTI) when the DINLF bit in JINTS1 is set to 1. 0: Disables an interrupt request. 1: Enables an interrupt request.
4	JOUTEN	Enables or disables a data transfer processing interrupt request (JDTI) when the JOUTF bit in JINTS1 is set to 1. 0: Disables an interrupt request. 1: Enables an interrupt request.
3	Reserved	When read, the value after reset is returned. When written, write the value after reset.
2	DBTEN	Enables or disables a data transfer processing interrupt request (JDTI) when the DBTF bit in JINTS1 is set to 1. 0: Disables an interrupt request. 1: Enables an interrupt request.
1	JINEN	Enables or disables a data transfer processing interrupt request (JDTI) when the JINF bit in JINTS1 is set to 1. 0: Disables an interrupt request. 1: Enables an interrupt request.

Table 44.37 JINTE1 register contents (2/2)

Bit position	Bit name	Function
0	DOUTLEN	Enables or disables a data transfer processing interrupt request (JDTI) when the DOUTLF bit in JINTS1 is set to 1. 0: Disables an interrupt request. 1: Enables an interrupt request.

44.3.32 JPEG Interrupt Status Register 1 (JINTS1)

This register shows the resons of interrupt.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 0000 0000_H

	31	30	29	28	27	26	25	24
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	7	6	5	4	3	2	1	0
	–	CBTF	DINLF	JOUTF	–	DBTF	JINF	DOUTLF
Compress	R	R/W*1	R/W*1	R/W*1	R	undefined	undefined	undefined
Decompress	R	undefined	undefined	undefined	R	R/W*1	R/W*1	R/W*1

Note 1. When the bit is read as 1, write 0 to clear it. When the bit is read as 0, write 1 to it.

Table 44.38 JINTS1 register contents

Bit position	Bit name	Function
6	CBTF	This bit is set to 1 when the last output coded data is written in compression.
5	DINLF	This bit is set to 1 when the number of input image data lines indicated by JIFESLC is read in compression. This bit is valid only when the DINLC bit in JIFECNT is set to 1.
4	JOUTF	This bit is set to 1 when the amount of output coded data indicated by JIFEDDC is written in compression. This bit is valid only when the JOUTC bit in JIFECNT is set to 1.
2	DBTF	This bit is set to 1 when the last output image data is written in decompression.
1	JINF	This bit is set to 1 when the amount of input coded data indicated by JIFDSDC is read in decompression. This bit is valid only when the JINC bit in JIFDCNT is set to 1.
0	DOUTLF	In decompression, this bit is set to 1 when the number of lines of output image data indicated by JIFDDLC have been written. This bit is valid only when the DOUTLC bit in JIFDCNT is set to 1.

44.3.33 JPEG input image data CbCr range setting register (JIFESVSZ)

JIFESVSZ sets the CbCr range of input image data.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 0000 0000_H

	31	30	29	28	27	26	25	24
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	DINYCHG	–	–	–	–	–	–	–
Compress	R/W	R	R	R	R	R	R	R
Decompress	undefined	R	R	R	R	R	R	R

	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

Table 44.39 JIFESVSZ register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, write the value after reset.
15	DINYCHG	Input Image Data CbCr Range Setting 0: Range from -128 to 127 1: Range from 0 to 255
14 to 0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

44.3.34 JPEG output image data CbCr range setting register (JIFESHSZ)

JIFESHSZ sets the CbCr range of output image data.

Access: This register can be accessed in 32-bit units.

Address: Refer to **Table 44.6**.

Initial Value: 0000 0000_H

	31	30	29	28	27	26	25	24
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

	15	14	13	12	11	10	9	8
	DOUTYCHG	–	–	–	–	–	–	–
Compress	R/W	R	R	R	R	R	R	R
Decompress	undefined	R	R	R	R	R	R	R

	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	–
Compress	R	R	R	R	R	R	R	R
Decompress	R	R	R	R	R	R	R	R

Table 44.40 JIFESHSZ register contents

Bit position	Bit name	Function
31 to 16	Reserved	When read, the value after reset is returned. When written, write the value after reset.
15	DOUTYCHG	Output Image Data CbCr Range Setting 0: Range from -128 to 127 1: Range from 0 to 255
14 to 0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

44.3.35 JPEG Codec Unit A Software reset register (JCSWRST)

Access: This register can be accessed in 32-bit units.

Address: FFC0 6010_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	–	–	–	–	–	–	–	JCUA0RES	–	–	–	–	–	–	–	–
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 44.41 JCSWRST register contents

Bit position	Bit name	Function
31 to 9	Reserved	When read, the value after reset is returned. When written, write the value after reset.
8	JCUA0RES	Software reset of JPEC Codec Unit A JCUA0 • Write: 0: de-activate software reset of JCUA module 1: activate software reset of JCUA module • Read: 0: software reset inactive (default) 1: software reset active Refer also to Section 44.6, Software Reset Processing.
7 to 0	Reserved	When read, the value after reset is returned. When written, write the value after reset.

44.4 Operation

44.4.1 Compression

44.4.1.1 Overview of Processing

The compression process flows are described below.

(1) The JPEG core is activated.
A marker is output. (After a marker is output, image data can be input.) Approximately 30,000 cycles (necessary for making SOI to SOS markers)

(2) Image data is transferred in MCUs from the external buffer to the JCUA.
If the count mode for stopping the input of image data lines is on, reading is stopped each time the number of lines set in JIFESLC is read. Reading is resumed by setting the DINRCMD bit in JIFECNT to 1.
When the DINRINI bit in JIFECNT is zero, the addresses for reading on resumption are continued from the addresses in the previous round of transfer.
When the DINRINI bit is one, the address set in JIFESA is used on resumption. Reading is also stopped when one frame of image data is completely transferred.
If the count mode for stopping the input of image data lines is off, reading is continued until one frame of image data is completely transferred.

(3) Image data is input to the JPEG core.
The input data is processed in MCUs at any time in the JPEG core.

(4) Coded data is transferred from the JCUA to the external buffer.
When the count mode for stopping the output of coded data is on, writing is stopped each time the amount of coded data set in JIFEDDC is written. Writing is resumed by setting the JOUTRCMD bit in JIFECNT to 1.
When the JOUTRINI bit in JIFECNT is zero, the addresses for writing on resumption are continued from the addresses in the previous round of transfer.
When the JOUTRINI bit is one, the address set in JIFEDA is used on resumption. Writing is also stopped when one frame of coded data is completely transferred.
If the count mode for stopping the output of coded data is off, writing is continued until one frame of coded data is completely transferred.

(5) Compression is completed after one frame of data is processed completely.

44.4.1.2 Flowchart (Compression)

(a) Initial Settings

After completing the JPEG core settings and input/output buffer settings and transferring image data to the external buffer, activate this module by setting the JSRT bit in JCCMD to 1. After the JCUA has been activated, the JPEG markers (SOI to SOS) are generated and output. It takes approximately 30,000 cycles to generate the markers.

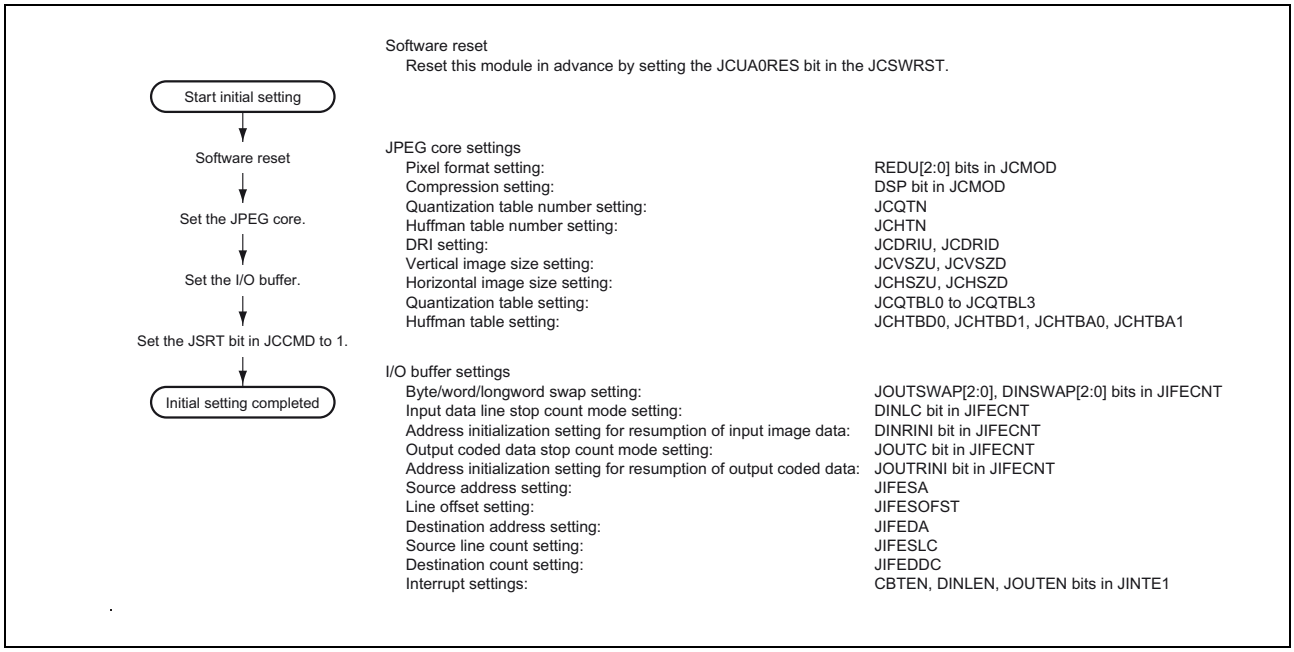


Figure 44.2 Compression Initial Setting Flow

(b) Compression Process

The compression process flows are described below.

- When JPEG compression process has been completed, the INS6 bit in JINTS0 is set to 1. However, the JCUA continues processing since the coded data remains to be transferred. The CBTF bit in JINTS1 is set to 1 when the last coded data is transferred. The interrupt source is cleared by writing 0 to the INS6 bit. However, the interrupt request asserted by this interrupt source cannot be cleared by writing 0 to the INS6 bit. Set an interrupt request clear command (by setting the JEND bit in JCCMD to 1) to clear the interrupt request.
- When the JCUA has completed compression and all coded data has been transferred, the CBTF flag in JINTS1 is set to 1. When the CBTEN bit in JINTE1 is 1 here, an interrupt is generated. The interrupt source is cleared by writing 0 to the CBTF flag.
- If the count mode for stopping image data lines is on, when the specified number of image data lines set in JIFESLC has been read, the DINLF flag in JINTS1 is set to 1, and reading is stopped. When the DINLEN bit in JINTE1 is 1 here, an interrupt is generated. An interrupt source is cleared by writing 0 to the DINLEN bit. Setting the DINRCMD bit in JIFECNT to 1 resumes reading.
  When the DINRINI bit in JIFECNT is zero, the addresses for reading on resumption are continued from the addresses in the previous round of transfer.
  When the DINRINI bit is one, the address set in JIFESA is used on resumption.
- If the count mode for stopping the output of coded data is on, when the specified amount of coded data set in JIFEDDC has been written, the JOUTF flag in JINTS1 is set to 1, and writing is stopped. When the JOUTEN bit in JINTE1 is 1 here, an interrupt is generated. An interrupt source is cleared by writing 0 to the JOUTF bit. Setting the JOUTRCMD bit in JIFECNT to 1 resumes writing.
  When the JOUTRINI bit in JIFECNT is zero, the addresses for writing on resumption are continued from the addresses in the previous round of transfer.
  When the JOUTRINI bit is one, the address set in JIFEDA is used on resumption.

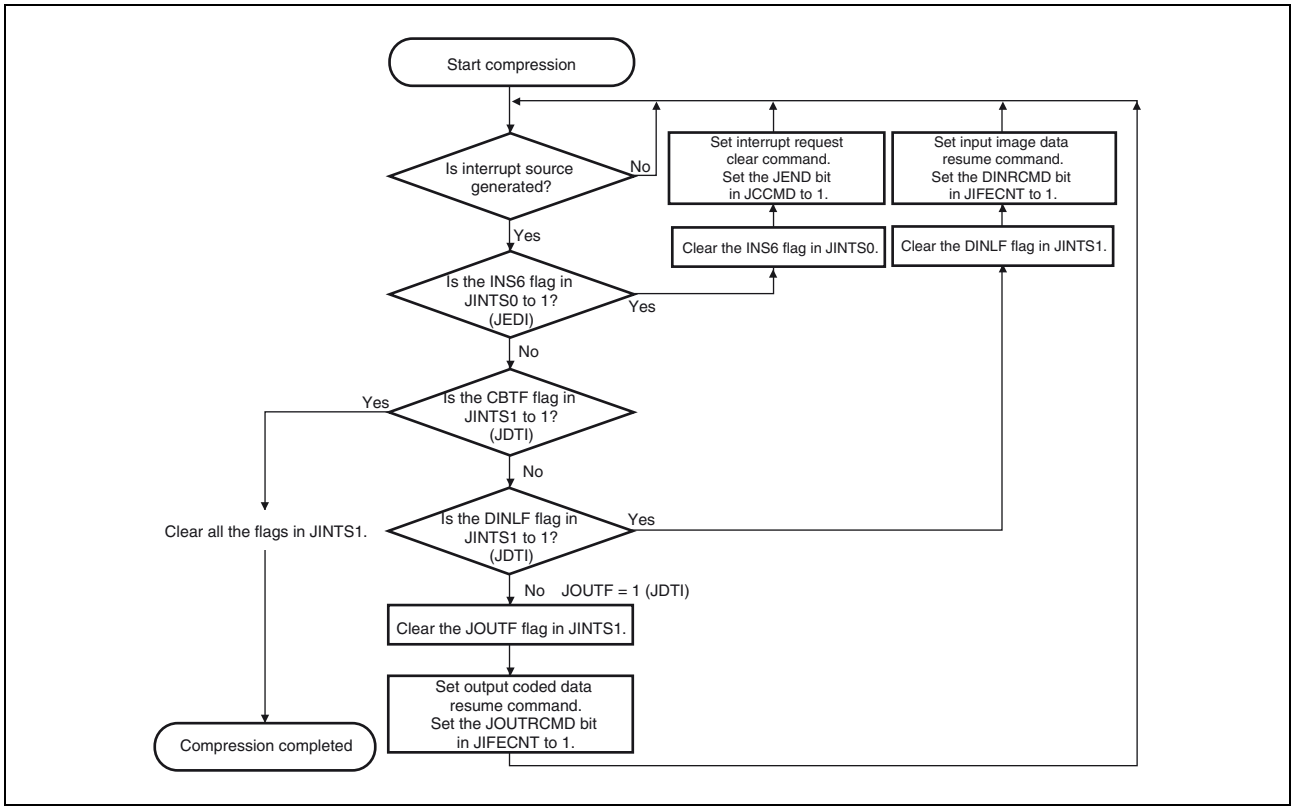


Figure 44.3 Compression Process Flow

44.4.1.3 JPEG Coded Data Format

The following figure shows the data output stream in compression. The amount of coded data from SOI to EOI is indicated by JCDTCU, JCDTCM, and JCDTCD. When both JCDRIU and JCDRID are set to H'00, the following markers are not output.

- DRI marker
- RST marker (in compressed image data)

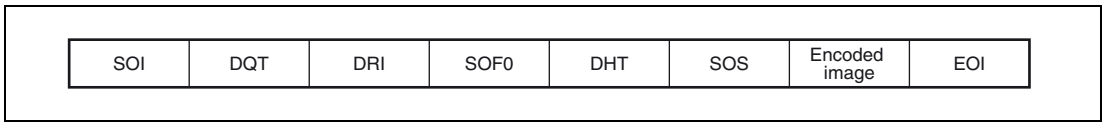


Figure 44.4 JPEG Coded Data Format

DQT: Not output for unused table.

DHT: Output in order DC0, AC0, DC1, and AC1. Not output for unused table.

SOF0: Component identifiers are C1 = first color component, C2 = second color component, and C3 = third color component.

SOS: Scan component selectors are CS1 = first color component, CS2 = second color component, and CS3 = third color component.

Header Volume (Reference):

SOI:	2 bytes (FFD8)
DQT:	134 bytes when two quantization tables are used, 199 bytes when three quantization tables are used (±65 bytes/table increase or decrease)
DRI:	6 bytes
SOF0:	19 bytes (4:2:2)
DHT:	420 bytes (two tables are used)
SOS:	14 bytes (4:2:2)
EOI:	2 bytes (FFD9)

44.4.1.4 Table Setting

(a) Quantization Table Specification

The order of addresses shown in 8 × 8 blocks corresponds to that of the register addresses. Do not access this table while the JCUA is in processing.

Table 44.42 Quantization Table

00	01	02	03	04	05	06	07
08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17
18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27
28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37
38	39	3A	3B	3C	3D	3E	3F

JCQTBL0 (<JCUAn_base> + H'100) = H'00

JCQTBL0 (<JCUAn_base> + H'101) = H'01

JCQTBL0 (<JCUAn_base> + H'102) = H'02

JCQTBL0 (<JCUAn_base> + H'103) = H'03

:

JCQTBL0 (<JCUAn_base> + H'13F) = H'3F

(b) Huffman Table Specification

Examples of the Huffman table specification given in the ITU-T T81 Annex K.3.3 recommended by JPEG are shown below. In compression, the following settings must be specified for all the codes so that Huffman codes can be generated for all the group numbers.

DC Huffman table:	The number of codes for each code length is 12. The group numbers in order of frequency of occurrence are 12.
AC Huffman table:	The number of codes for each code length is 162. The zero run length/the group numbers in order of frequency of occurrence are 162.

Do not access the following tables while the JCUA is in processing. In particular, read access is prohibited.

- Table K.3/T81
  JCHTBD0 (<JCUAn_base> + H'200) = H'00
  JCHTBD0 (<JCUAn_base> + H'201) = H'01
  JCHTBD0 (<JCUAn_base> + H'202) = H'05
  JCHTBD0 (<JCUAn_base> + H'203) = H'01
  :
  JCHTBD0 (<JCUAn_base> + H’21B) = H'0B
- Table K.4/T81
  JCHTBD1 (<JCUAn_base> + H'300) = H'00
  JCHTBD1 (<JCUAn_base> + H'301) = H'03
  JCHTBD1 (<JCUAn_base> + H'302) = H'01
  JCHTBD1 (<JCUAn_base> + H'303) = H'01
  :
  JCHTBD1 (<JCUAn_base> + H'31B) = H'0B
- Table K.5/T81
  JCHTBA0 (<JCUAn_base> + H'220) = H'00
  JCHTBA0 (<JCUAn_base> + H'221) = H'02
  JCHTBA0 (<JCUAn_base> + H'222) = H'01
  JCHTBA0 (<JCUAn_base> + H'223) = H'03
  :
  JCHTBA0 (<JCUAn_base> + H'2D1) = H'FA
- Table K.6/T81
  JCHTBA1 (<JCUAn_base> + H'320) = H'00
  JCHTBA1 (<JCUAn_base> + H'321) = H'02
  JCHTBA1 (<JCUAn_base> + H'322) = H'01
  JCHTBA1 (<JCUAn_base> + H'323) = H'02
  :
  JCHTBA1 (<JCUAn_base> + H'3D1) = H'FA

44.4.1.5 Input Pixel Format

Image data in the YCbCr422 format can be input to this module. Allocation of data in the YCbCr422 format can be changed by the DINSWAP[2:0] bits in JIFECNT as shown below.

- When the DINSWAP[2:0] bits = 000

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Y0 8 bits	Cb0 8 bits	Y1 8 bits	Cr0 8 bits	Y2 8 bits	Cb1 8 bits	Y3 8 bits	Cr1 8 bits

- When the DINSWAP[2:0] bits = 001

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Cb0 8 bits	Y0 8 bits	Cr0 8 bits	Y1 8 bits	Cb1 8 bits	Y2 8 bits	Cr1 8 bits	Y3 8 bits

- When the DINSWAP[2:0] bits = 010

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Y1 8 bits	Cr0 8 bits	Y0 8 bits	Cb0 8 bits	Y3 8 bits	Cr1 8 bits	Y2 8 bits	Cb1 8 bits

- When the DINSWAP[2:0] bits = 011

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Cr0 8 bits	Y1 8 bits	Cb0 8 bits	Y0 8 bits	Cr1 8 bits	Y3 8 bits	Cb1 8 bits	Y2 8 bits

- When the DINSWAP[2:0] bits = 100

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Y2 8 bits	Cb1 8 bits	Y3 8 bits	Cr1 8 bits	Y0 8 bits	Cb0 8 bits	Y1 8 bits	Cr0 8 bits

- When the DINSWAP[2:0] bits = 101

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Cb1 8 bits	Y2 8 bits	Cr1 8 bits	Y3 8 bits	Cb0 8 bits	Y0 8 bits	Cr0 8 bits	Y1 8 bits

- When the DINSWAP[2:0] bits = 110

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Y3 8 bits	Cr1 8 bits	Y2 8 bits	Cb1 8 bits	Y1 8 bits	Cr0 8 bits	Y0 8 bits	Cb0 8 bits

- When the DINSWAP[2:0] bits = 111

b63 b56	b55 b48	b47 b40	b39 b32	b31 b24	b23 b16	b15 b8	b7 b0
Cr1 8 bits	Y3 8 bits	Cb1 8 bits	Y2 8 bits	Cr0 8 bits	Y1 8 bits	Cb0 8 bits	Y0 8 bits

44.4.1.6 Output Coded Data

In the case of compression, coded data are output. This module handles the output of coded data in 16-bit units. For this reason, if the coded data have an odd code length (are fractional), the final code for output will be H'D9FF.

The JOUTSWAP[2:0] bits in JIFECNT can be used to alter the arrangement of coded data in the output.

44.4.2 Decompression

44.4.2.1 Overview of Processing

The decompression process flows are described below.

1. The JPEG core is activated.
2. Coded data is transferred from the external buffer to the JCUA.
   If the count mode for stopping the input of coded data is on, reading is stopped each time the amount of coded data set in JIFDSDC is read. Reading is resumed by setting the JINRCMD bit in JIFDCNT to 1. When the JINRINI bit in JIFDCNT is zero, the addresses for reading on resumption are continued from the addresses in the previous round of transfer.
   When the JINRINI bit is one, the address set in JIFDSA is used on resumption. Reading is stopped when the end of the coded data is detected.
   If the count mode for stopping the input of coded data is off, reading is continued until the end of code is detected.
   With this module, more coded data may be read than the coded data size since coded data reading is continued until the end of code is detected.
3. Coded data is input to the JPEG core.
   The input data is processed in MCUs at any time in the JPEG core.

4. Image data is transferred in MCUs from the JCUA to the external buffer.
   When the count mode for stopping the output of image data lines is on, writing is stopped each time the number of image data lines set in JIFDDLC is written. Writing is resumed by setting the DOUTRCMD bit in JIFDCNT to 1.
   When the DOUTRINI bit in JIFDCNT is zero, the addresses for writing on resumption are continued from the addresses in the previous round of transfer.
   When the DOUTRINI bit is one, the address set in JIFDDA is used on resumption. Writing is stopped when one frame of image data is completely transferred.
   If the count mode for stopping the output of image data lines is off, writing is continued until one frame of image data is completely transferred.
5. Decompression is completed after one frame of data is processed completely.

(a) Initial Settings

- When the INT3 bit in JINTE0 is set to 0:
  After completing the JPEG core settings and input/output buffer settings and transferring coded data to the external buffer, activate this module by setting the JSRT bit in JCCMD to 1.
- When the INT3 bit in JINTE0 is set to 1:
  After completing the JPEG core settings and input buffer settings and transferring coded data to the external buffer, activate this module by setting the JSRT bit in JCCMD to 1.
  When the image size and pixel format become readable after the coded data has been decompressed, the INS3 bit in JINTS0 is set. At this time, decompression is temporarily stopped.
  After the image size and pixel format have been read, set the output buffer.
  Setting the JRST bit in JCCMD to 1 after interrupt handling resumes decompression.

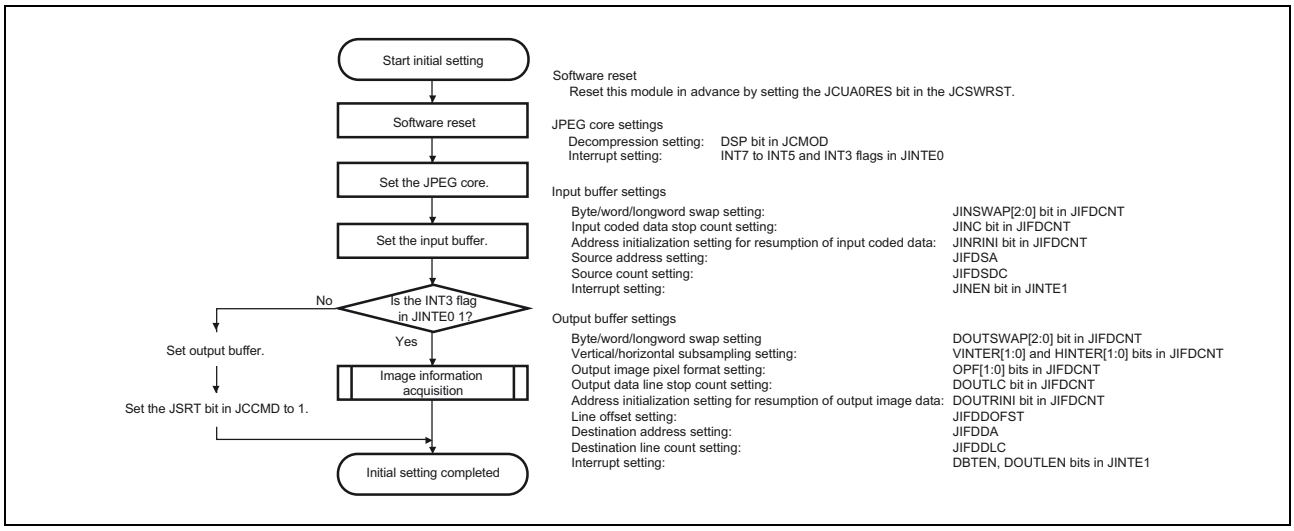


Figure 44.5 Decompression Initial Setting Flow

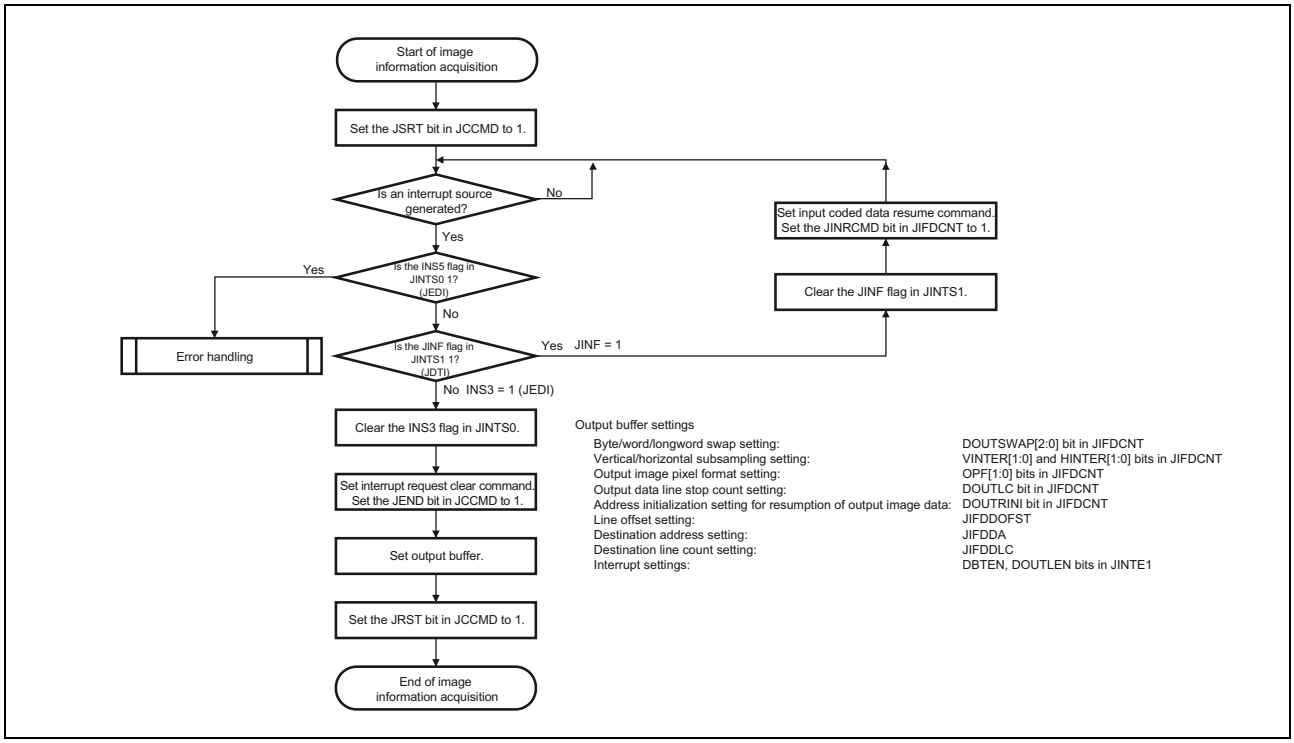


Figure 44.6 Image Information Acquisition Flow

(b) Decompression Process

The decompression process flows are described below.

- When JPEG decompression process has been completed, the INS6 bit in JINTS0 is set to 1. However, the JCUA continues processing since the image data remains to be transferred. The DBTF bit in JINTS1 is set to 1 when the last image data is transferred. The interrupt source is cleared by writing 0 to the INS6 bit. However, the interrupt request asserted by this interrupt source cannot be cleared by writing 0 to the INS6 bit. Set an interrupt request clear command (by setting the JEND bit in JCCMD to 1) to clear the interrupt request.
- When the JCUA has completed decompression process and all image data has been transferred, the DBTF flag in JINTS1 is set to 1. When the DBTEN bit in JINTE1 is 1 here, an interrupt is generated. The interrupt source is cleared by writing 0 to the DBTF flag.
- If the count mode for stopping input coded data is on, when the specified amount of coded data set in JIFDSDC have been read, the JINF flag in JINTS1 is set to 1, and reading is stopped.
  When the JINEN bit in JINTE1 is 1 here, an interrupt is generated. An interrupt source is cleared by writing 0 to the JINF bit. Setting the JINRCMD bit in JIFDCNT to 1 resumes reading.
  When the JINRINI bit in JIFDCNT is zero, the addresses for reading on resumption are continued from the addresses in the previous round of transfer.
  When the JINRINI bit is one, the address set in JIFDSA is used on resumption.
- If the count mode for stopping the output image data is on, when the specified number of image data lines set in JIFDDLC have been written, the DOUTLF flag in JINTS1 is set to 1, and writing is stopped. When the DOUTLEN bit in JINTE1 is 1 here, an interrupt is generated. An interrupt source is cleared by writing 0 to the DOUTLF bit. Setting the DOUTRCMD bit in JIFDCNT to 1 resumes writing.
  When the DOUTRINI bit in JIFDCNT is zero, the addresses for writing on resumption are continued from the addresses in the previous round of transfer.

- When the DOUTRINI bit is one, the address set in JIFDDA is used on resumption.

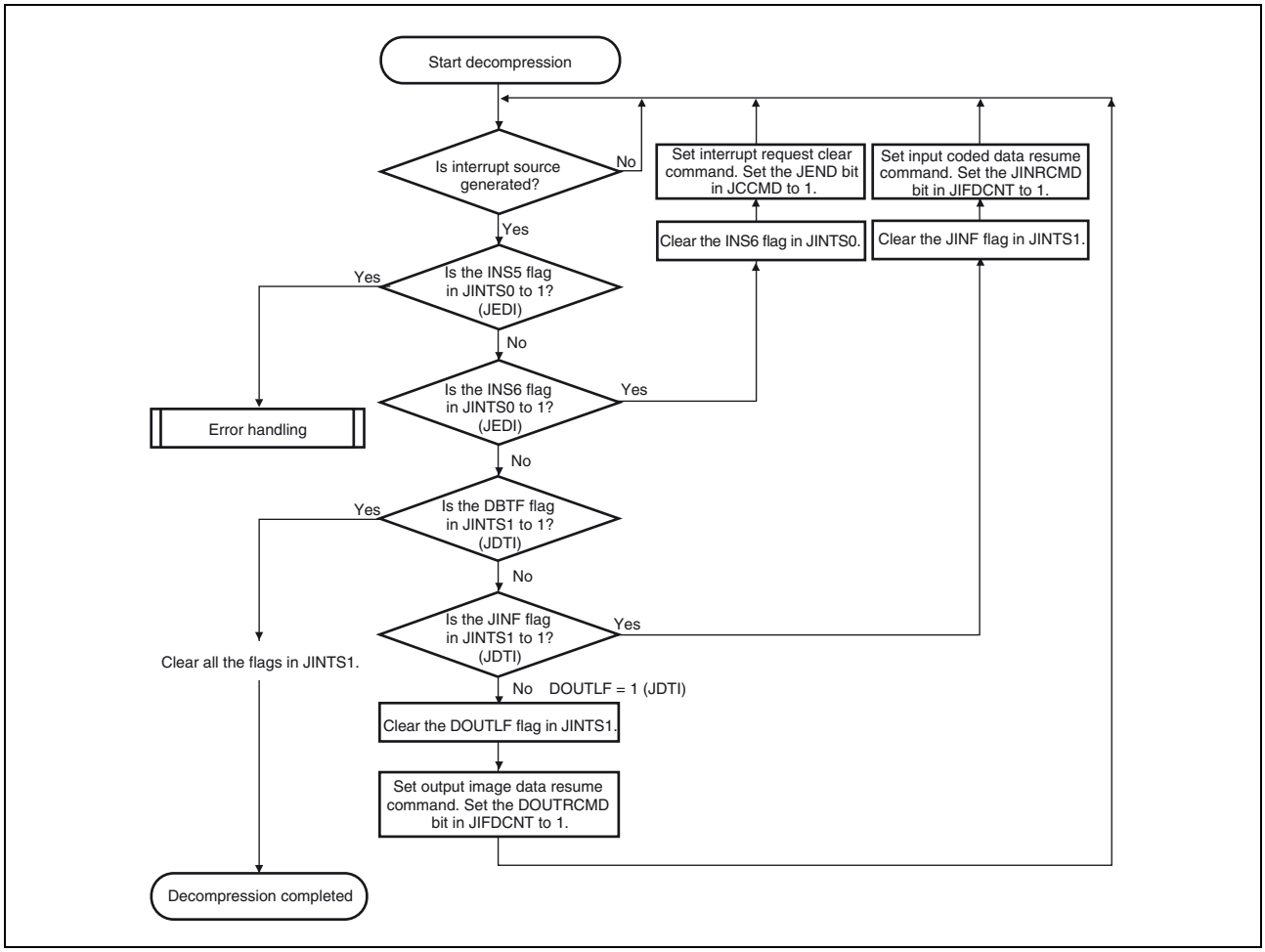


Figure 44.7 Decompression Process Flow

(c) Error Handling

If the INS5 bit in JINTS0 is 1, it indicates that there is an error in the input JPEG coded data and that the decompression process by this module has been ended. Read the ERR[3:0] bits in JCDERR to determine the cause of the error. The interrupt signal asserted due to the interrupt source indicated by the INS5 bit cannot be negated by clearing the interrupt status through 0-writing. To clear the interrupt request, set the interrupt request clear command (by setting the JEND bit in JCCMD to 1).

The JCUA module must be reset after detecting error. To ensure that JCUA register accesses are done before software reset, dummy read and SYNCM operation should be inserted. (This operation leads 30 µs waiting period and module reset can be done safely.)

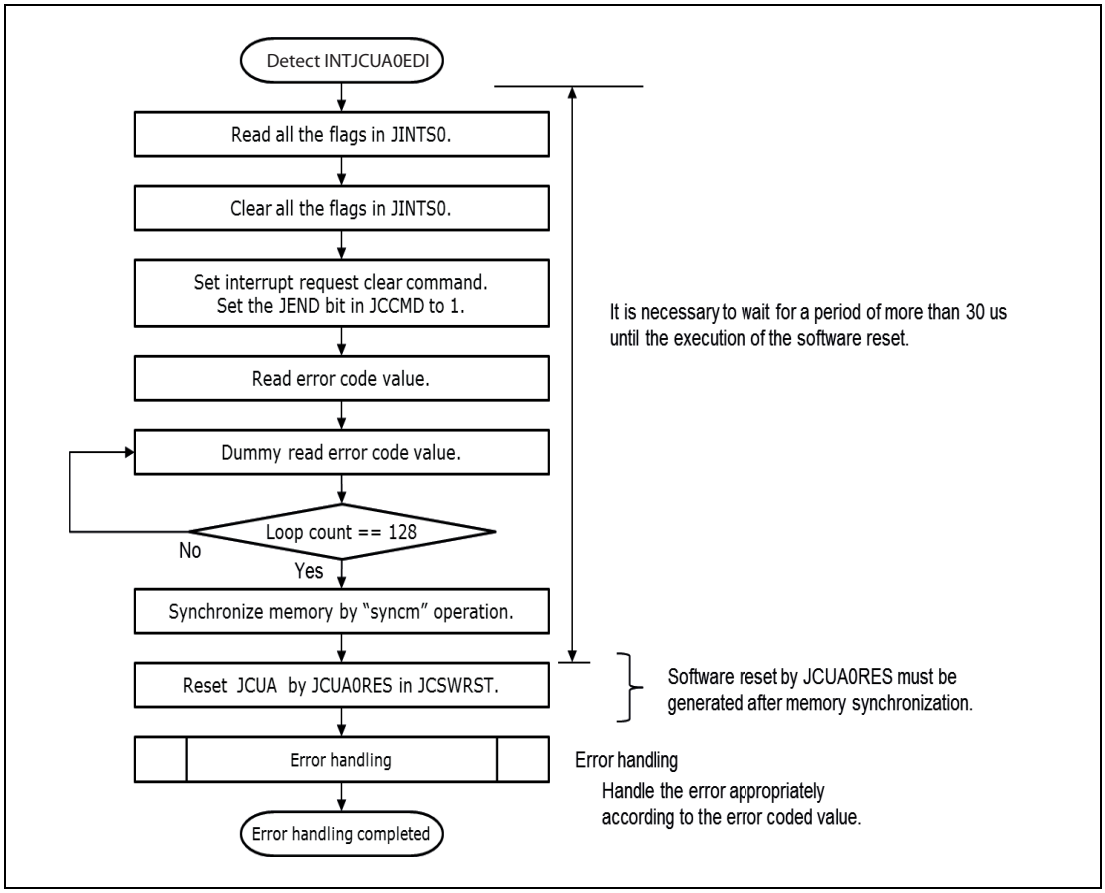


Figure 44.8 Error Handling Flow

44.4.2.2 Input JPEG Coded Data

Markers to be processed in decompression are SOI, SOF0, SOS, DQT, DHT, DRI, RSTm, and EOI. Other markers except for the error markers shown below are ignored even if they are read. The JINSWAP[2:0] bits in JIFDCNT can be used to alter the arrangement for the input of coded data.

NOTE

When JCUA ignored unknown markers, JCUA continue to read the next data until detecting valid marker.

Therefore, input data for decompressed must always include essential markers for JPEG like SOI and EOI.

44.4.2.3 JPEG Decompression Errors

(a) Error Marker

If a marker error is found while analyzing compressed data for decompression, the code to identify the error type (shown in **Table 44.43**) is set to ERR[3:0] bits in JCDERR. When an error is detected, the JCUA generates an interrupt signal and terminates decoding. The stored code value will be set to B'1010 (default value) at the start of processing of the next frame or after a bus reset.

Table 44.43 Decompression Error Codes

Code	Description
B'0000	Normal
B'0001	SOI not detected: SOI not detected until EOI detected
B'0010	SOF1 to SOFF detected
B'0011	Unprovided pixel format detected
B'0100	SOF accuracy error: Other than 8 detected
B'0101	DQT accuracy error: Other than 0 detected
B'0110	Component error 1: The number of SOF0 header components detected is other than 1, 3, or 4
B'0111	Component error 2: The number of components differs between SOF0 header and SOS
B'1000	SOF0, DQT, or DHT not detected when SOS detected
B'1001	SOS not detected: SOS not detected until EOI detected
B'1010	EOI not detected (default)
B'1011	Restart interval data number error detected
B'1100	Image size error detected
B'1101	Last MCU data number error detected
B'1110	Block data number error detected

(b) Huffman Coded Segment Error

During the compressed data analysis in decompression operation, if there is an increase or decrease in the decoded data count due to an error resulting from bit reversal or missing data in the Huffman-coded segment, determine the error type, and set the error code in the ERR[3:0] bit in JCDERR. **Table 44.44** lists the segment error codes. The error code is set, interrupt signal is issued, and the process is ended only if the bits INT7 to INT5 in JINTE0 corresponding to the detected error is set to 1. The set code value will turn to the default value (B'1010) at the start of processing of the next frame or after a reset.

However, in this error detection, if an error in the Huffman-coded segment does not result in an alteration in the decoded data count, the error will go undetected.

[Example]

The number of data in a Huffman coded segment with pixel format setting YCbCr422, DRI = 2, X = 80 pixels, and Y = 8 pixels

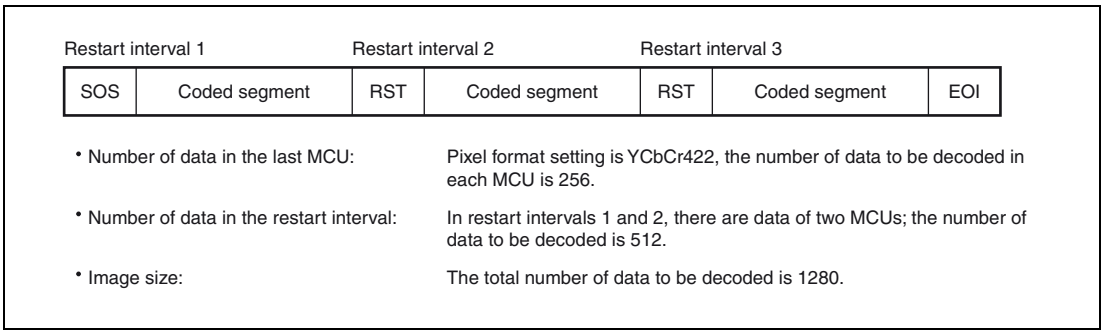


Figure 44.9 Huffman Coded Segment

Table 44.44 Segment Error Codes

Code	Description
B'0000	Normal
B'1011	Restart interval data number error: The number of data in each interval is compared with the number of data specified by the DRI marker. If an interval has more or less data that is specified by the DRI marker, the decompression error code (1011) is set. The last interval which is shorter than the restart interval is not compared. If the DRI marker segment is not placed or the specified number is 00, an error is not detected even if the RSTm marker is placed. Also an m which indicates the order of RSTm marker modulo 8 (m = 0 to 7) is exempt from the error detection analysis. When the INT7 bit in JINTE0 is set to 0, this error is not detected.
B'1100	Image size error: The data number of an image which is calculated from the number of lines specified by the frame parameter and the number of samples per line is compared with the total number of data from SOS to EOI (in pixel units). If the numbers of data do not match, the decompression error code (1100) is set. When the INT6 bit in JINTE0 is set to 0, this error is not detected. The data number of an image is shown in MCU units. Thus the number of lines and the number of samples per line for calculation need to be shown in MCU units.
B'1101	Last MCU data number error: Whether the number of data in the MCUs at the EOI detection is shown in MCU units is checked and fractions are detected. If error (1100) occurs simultaneously, error (1100) has priority. When the INT5 bit in JINTE0 is set to 0, this error is not detected.
B'1110	Block data number error: Whether a block is an 8 × 8 array is checked; the check is performed for fractions. When bits INT7 to INT5 in JINTE0 are all set to 0, this error is not detected.

44.4.3 Output Pixel Format in Decompression

This module is capable of decompressing JPEG encoded data created in the YCbCr444, YCbCr422, YCbCr411 and YCbCr420 formats. The pixel format of the output image will be YCbCr422, ARGB8888, or RGB565. The flow of conversion of decompressed data to the given output pixel format is shown below.

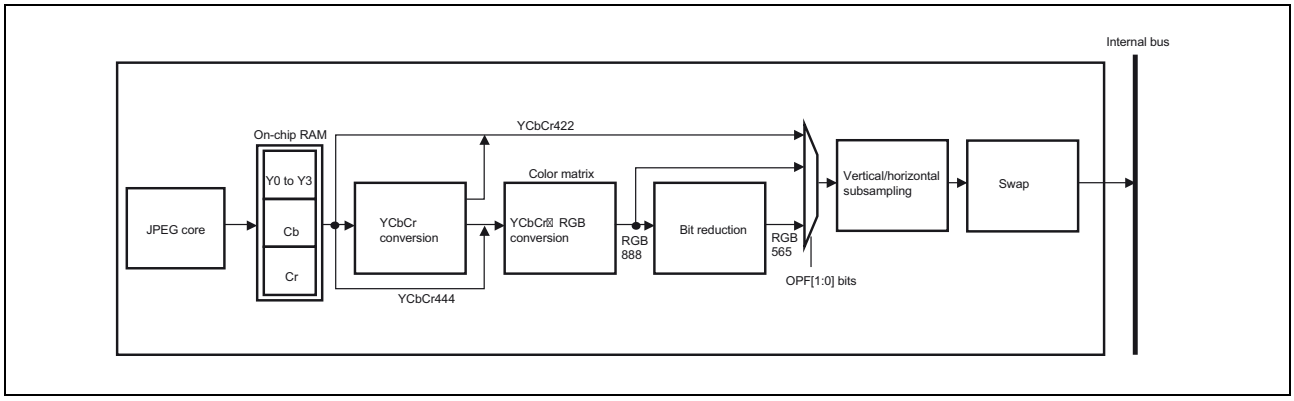


Figure 44.10 Block Diagram of Output Pixel Format Conversion in Decompression

44.4.3.1 On-chip RAM

Data decoded by the JPEG core are stored in MCUs on RAM in this module.

44.4.3.2 YCbCr Conversion

When data are to be output in the ARGB8888 or RGB565 format, data in the YCbCr422, YCbCr411or YCbCr420 format are first converted to the YCbCr444 format.

When data are to be output in the YCbCr422 format, data in the YCbCr444, YCbCr411 or YCbCr420 format are converted to the YCbCr422 format.

Conversion is performed using simple interpolation.

44.4.3.3 YCbCr → RGB Conversion

Data in the YCbCr444 format are converted to the RGB888 format. The following formulae are used.

R = 1.000Y + 1.402Cr

G = 1.000Y - 0.344Cb - 0.714Cr

B = 1.000Y + 1.772Cb

44.4.3.4 Bit Reduction

RGB888 data is reduced to RGB565 data. The lower three bits of red and blue, and lower two bits of green are removed.

44.4.3.5 Output Pixel Format Selection

The pixel format to be output is selected by the OPF[1:0] bit in JIFDCNT.

Allocation of data (while the DOUTSWAP[2:0] bits in JIFDCNT = 000) in the pixel format is shown below.

- YCbCr422 (32 bits/pixel)

b31 – b24	b23 – b16	b15 – b8	b7 – b0
Y0 8 bits	Cb 8 bits	Y1 8 bits	Cr 8 bits

- ARGB8888 (32 bits/pixel)

b31 – b24	b23 – b16	b15 – b8	b7 – b0
*1	Red 8 bits	Green 8 bits	Blue 8 bits

Note 1. This value is determined by the ALPHA[7:0] bits in JIFDADT.

- RGB565 (16 bits/pixel)

b15 – b11	b10 – b5	b4 – b0
Red 5 bits	Green 6 bits	Blue 5 bits

44.4.3.6 Vertical/Horizontal Subsampling

The output data can be horizontally and vertically subsampled according to the VINTER[1:0] and HINTER[1:0] bit setting in JIFDCNT.

Figure 44.11 to **Figure 44.13** show line subsampling modes.

For the output formats ARGB8888 and RGB565, one cell represents one pixel in the figures. For the output format YCbCr422, one cell represents one set of Y0Cb0Y1Cr0 in the figures.

As subsampling is carried out by minimum coded units (MCU), the numbers of the horizontal and vertical block units will vary according to the decompressed pixel format.

The following tables show the values of n and m in the figures.

Horizontal:

Table 44.45 Number of Horizontal Blocks

Compression Format	Output Format	n
YCbCr444	YCbCr422	1/2
YCbCr444	ARGB8888, RGB565	1
YCbCr422	YCbCr422	1
YCbCr422	ARGB8888, RGB565	2
YCbCr411	YCbCr422	2
YCbCr411	ARGB8888, RGB565	4
YCbCr420	YCbCr422	1
YCbCr420	ARGB8888, RGB565	2

Vertical:

Table 44.46 Number of Vertical Blocks

Compression Format	Output Format	m
YCbCr444	YCbCr422	1
YCbCr444	ARGB8888, RGB565	1
YCbCr422	YCbCr422	1
YCbCr422	ARGB8888, RGB565	1
YCbCr411	YCbCr422	1
YCbCr411	ARGB8888, RGB565	1
YCbCr420	YCbCr422	2
YCbCr420	ARGB8888, RGB565	2

- Subsampling into 1/2
  Even lines are skipped by subsampling.

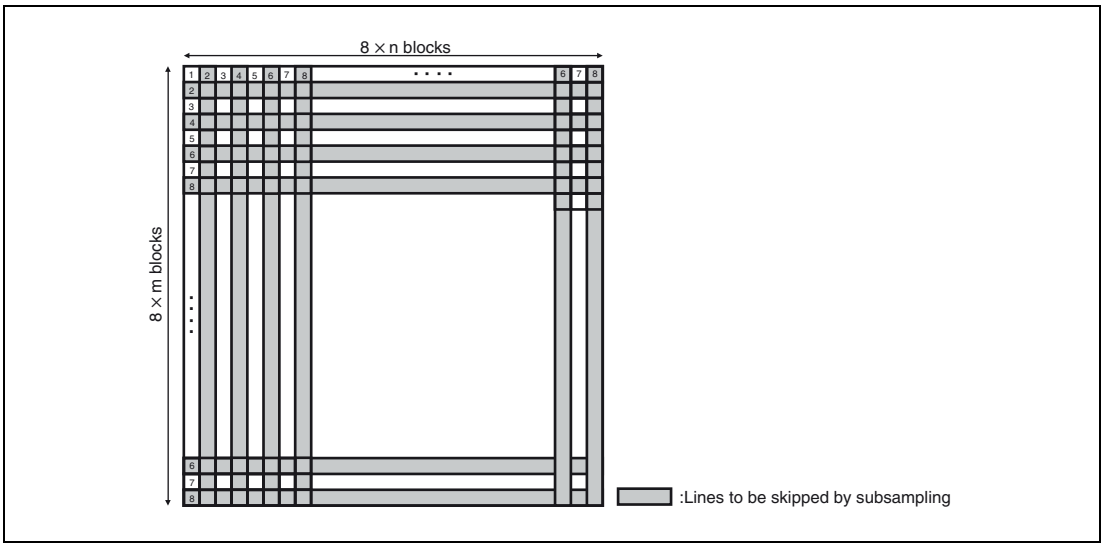


Figure 44.11 MCU when subsampling into 1/2 is selected

- Subsampling into 1/4
  The second, third, and fourth lines are skipped by subsampling.

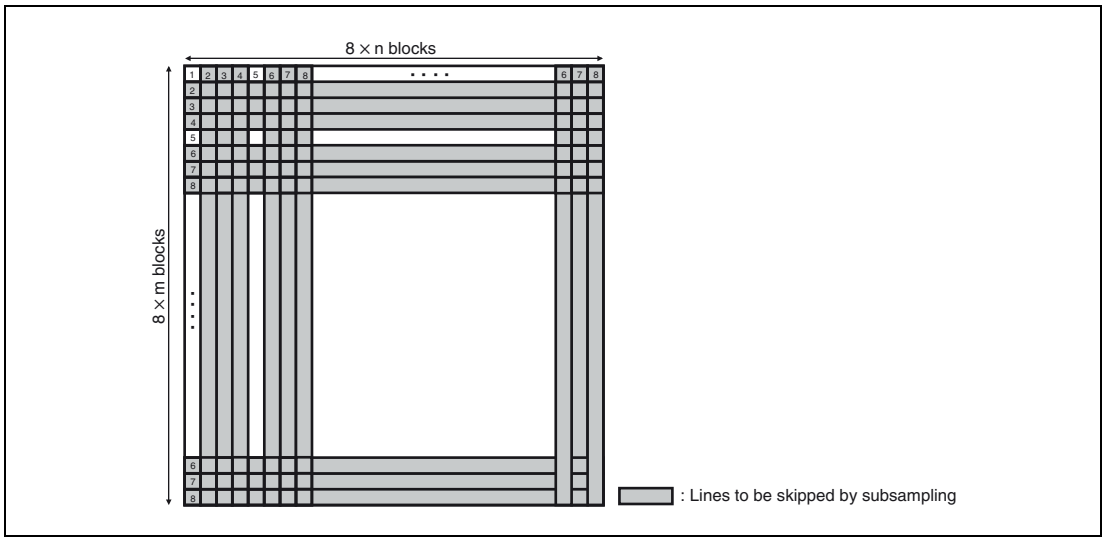


Figure 44.12 MCU when subsampling into 1/4 is selected

- Subsampling into 1/8
  The second, third, fourth, fifth, sixth, seventh, and eighth lines are skipped by subsampling.

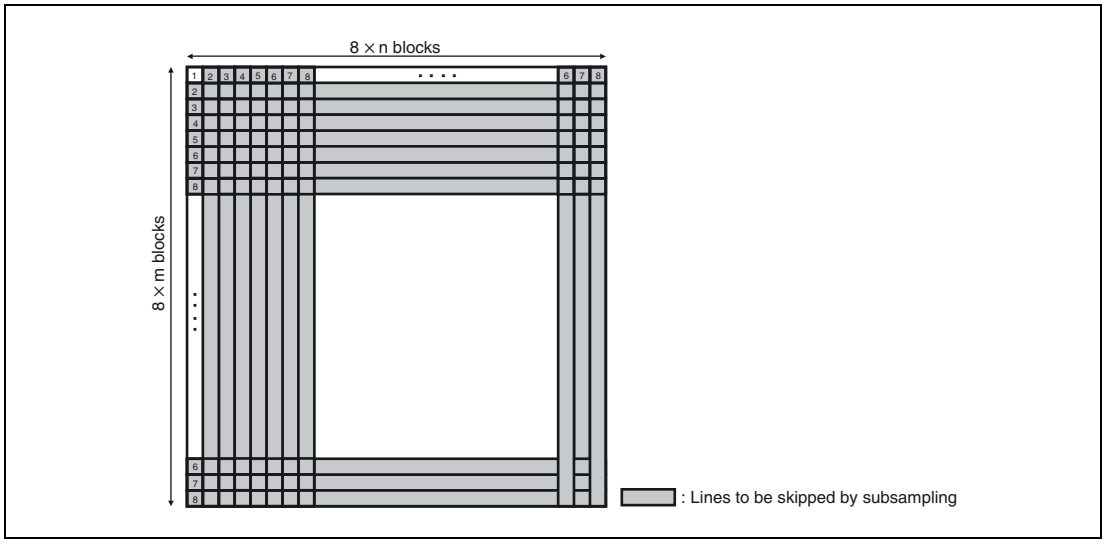


Figure 44.13 MCU when subsampling into 1/8 is selected

44.4.3.7 Swap

Allocation of data can be changed by the DOUTSWAP[2:0] bits in JIFDCNT.

44.4.4 Storing Image Data

The following figure shows the buffer area for storing the image data.

- Start address
  Compression: JIFESA
  Decompression: JIFDDA

- Horizontal size
  Compression, decompression: JCHSZU, JCHSZD
- Vertical size
  Compression, decompression: JCVSZU, JCVSZD
- Offset
  Compression: JIFESOFST
  Decompression: JIFDDOFST

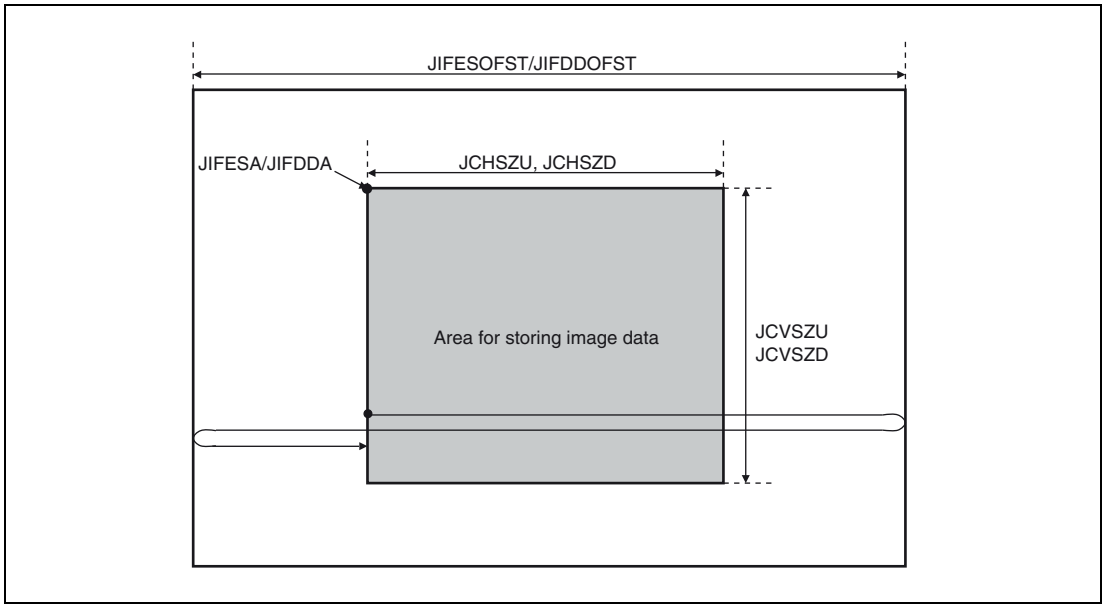


Figure 44.14 Image of Storing Image Data

44.5 Interrupts

Two types of interrupt requests, namely compression/decompression process interrupt request (JEDI) and data transfer interrupt request (JDTI), are available in this module. The two types of interrupt requests are each related to multiple sources. The interrupt request cancellation methods differ depending on the source of the interrupt request.

44.5.1 Compression/Decompression Process Interrupt Request (JEDI)

The flags in JINTS0 indicate compression/decompression-related sources. The interrupt requests asserted by these interrupt sources cannot be negated by clearing the corresponding interrupt status bits to 0. Issue an interrupt request clear command (by setting the JEND bit in JCCMD to 1) to clear the interrupt request. When a flag in JINTS0 is set to 1, a compression/decompression process interrupt request is sent to the interrupt controller.

44.5.1.1 Compression

- JPEG compression process end
  When the INS6 bit in JINTS0 is 1, the JPEG compression process has been successfully completed. After all of the coded data is transferred, the JCUA completes compression.

44.5.1.2 Decompression

- JPEG decompression process end

When the INS6 bit in JINTS0 is 1, the JPEG decompression process has been successfully completed. After all of the image data is transferred, JCUA completes decompression.

- JPEG decompression error occurrence
  When the INS5 bit in JINTS0 is 1, the input JPEG coded data has an error and the JCUA has stopped the decompression process. Read the error code (ERR[3:0] bits in JCDERR) and identify the error source. This interrupt occurs when any of the INT7 to INT5 bits in JINTE0 is 1.
- Request for reading the image size and pixel format
  When the INS3 bit in JINTS0 is 1, JPEG coded data has been input and information regarding the image size and pixel format can be read. Since the JPEG decompression process is suspended, resume the JPEG decompression process by setting the process stop clear command after accessing the necessary registers. This interrupt occurs when the INT3 bit in JINTE0 is 1.

44.5.2 Data Transfer Interrupt Request (JDTI)

The flags in JINTS1 are the interrupt sources for transferring the image data and coded data. The interrupt requests asserted by these interrupt sources can be negated by clearing the corresponding interrupt status bits to 0.

44.5.2.1 Compression

- Interrupt request generated after the specified number of input image data lines has been read.
  When the DINLF bit in JINTS1 is 1, the number of image data lines specified by JIFESLC has been transferred; transfer the rest of the image data to the external buffer and resume transferring the data from the external buffer. A data transfer interrupt request is sent when the DINLEN bit in JINTE1 is 1.
- Interrupt request generated after the specified amount of output coded data have been written to.
  When the JOUTF bit in JINTS1 is 1, the amount of coded data specified by JIFEDDC has been transferred. Secure a space for the next coded data in the external buffer, and resume transfer process. The data transfer interrupt request is sent when the JOUTEN bit in JINTE1 is 1.
- Interrupt request generated after all processes are completed
  When the CBTF bit in JINTS1 is 1, the JCUA has completed compression and transferred all of the coded data. The data transfer interrupt request is sent when the CBTEN bit in JINTE1 is set to 1.

44.5.2.2 Decompression

- Interrupt request generated after the specified number of output image data lines has been written to.
  When the DOUTLF bit in the JINTS1 is 1, the number of image data lines specified by JIFDDLC has been transferred. Secure a space for the next coded data in the external buffer, and resume transfer process. A data transfer interrupt request is sent when the DOUTLEN bit in JINTE1 is 1.
- Interrupt request generated after the specified amount of input coded data has been read.
  The JINF bit in JINTS1 becomes 1 when the amount of coded data specified by JIFDSDC has been transferred. Secure the next coded data in the external buffer, and resume transfer process. A data transfer interrupt is also sent at this time if the JINEN bit in JINTE1 is 1.
- Interrupt request generated after all processes are completed
  The DBTF bit in JINTS1 becomes 1 when the JCUA has completed decompression and transferred all of the coded data. A data transfer interrupt request is also sent at this time if the DBTEN bit in JINTE1 is set to 1.

44.5.2.3 Procedure of checking JINTS1

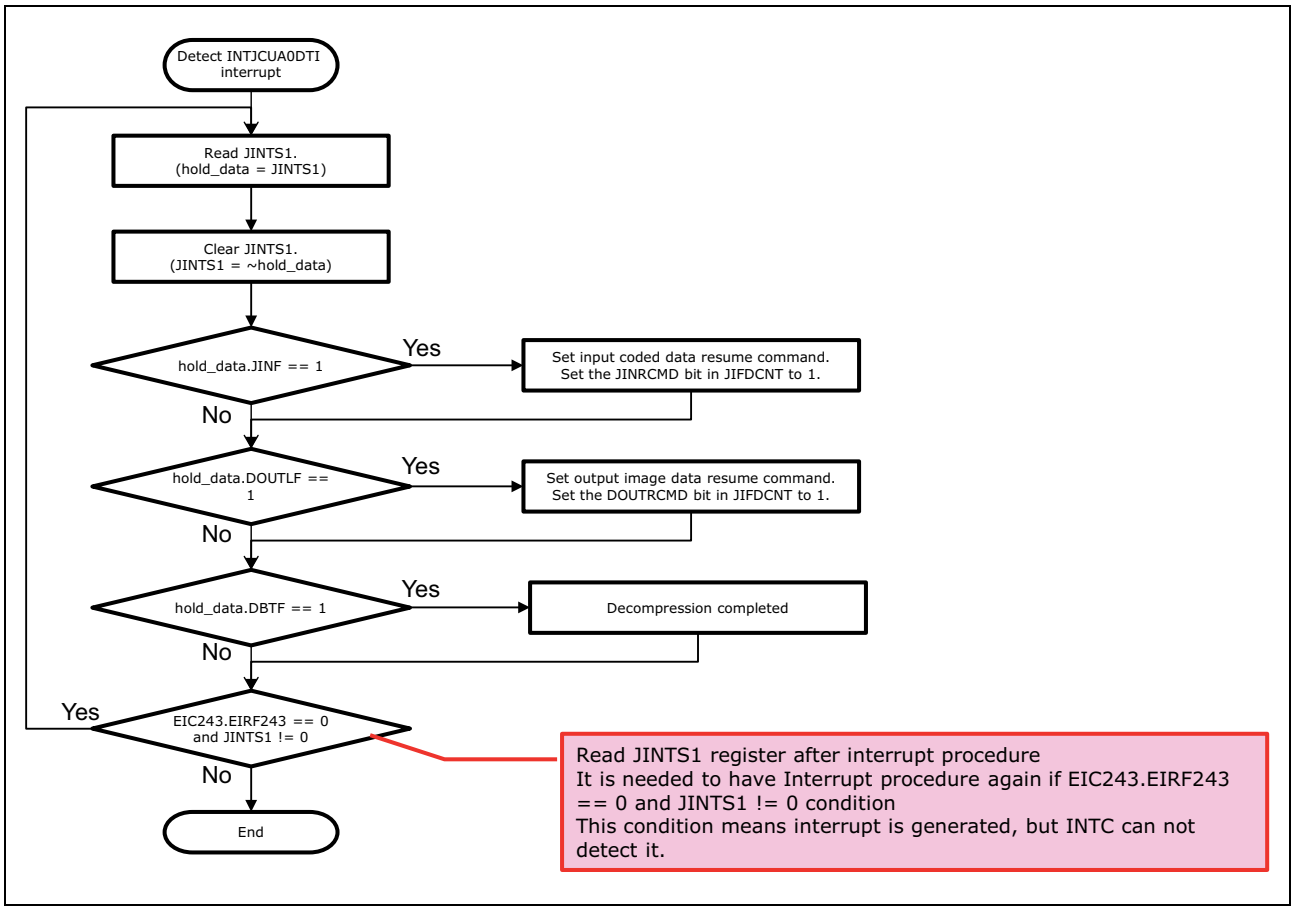


Figure 44.15 Procedure of checking JINTS1

44.6 Software Reset Processing

Setting of the JCUA0RES bit in the JCSWRST causes a software reset.

This reset is available after JCUA stop or error status.

Use the following software reset procedure to reset the JCUA unit:

1. Confirm any stop or error status is asserted in JINTS0 or JINTS1.
2. Wait for 30us.
3. Set JCSWRST.JCUA0RES bit to 1, to generate the software reset condition.
4. Read JCSWRST.JCUA0RES bit until bit is set to 1, to wait for software reset is active.
5. Set JCSWRST.JCUA0RES bit to 0, to release software reset condition.

By this sequence all JCUA registers with exception of the JCSWRST register are initialized by a software reset.

CAUTION

It is prohibit that JCSWRST.JCUA0RES is set to 1 during JPEG processing.

NOTE

A waiting time for 30 us is needed in order to wait that the remaining data from JCUA have been output to RAM. To archive this 30 µs waiting time dummy read and SYNCM operation should be inserted.

44.7 Usage Notes

44.7.1 Pixel Format YCbCr

This module treats the range from -128 to 127 as input/output values for YCbCr values in the YCbCr422 or YCbCr420 pixel format. On the other hand, video display controller treats the range from 0 to 255 as input/output values for YCbCr. Therefore, when bidirectionally transferring data in YCbCr pixel formats between this module and video display controller, correct YCbCr values by adding or subtracting 128.

Section 45 A/D Converter (ADCE)

This section contains a generic description of the A/D Converter (ADCE).

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

45.1 Overview of RH850/D1L/D1M A/D Converter (ADCE)

45.1.1 Units

This microcontroller has the following number of units of the A/D Converter (ADCE).

Table 45.1 Units

A/D Converter (ADCE)	
Units	1
Names	ADCE0

Units index n

Throughout this section, the individual units of an A/D Converter is identified by the index "n" (n = 0), for example ADCEnADCR for the ADCEn control register.

45.1.2 Channels and scan groups

Throughout this section, the individual physical A/D Converter channels are identified by the index “m”.
The available number of A/D Converter channels depends on the device:

Table 45.2 A/D Converter channels

ADCEn unit	D1L1	D1L2(H)	D1M1(H) D1M1-V2 D1M1A	D1M2(H)
ADCE0	m = 0 to 15	m = 0 to 15	m = 0 to 15	m = 0 to 19

Table 45.3 Index

Index	Meaning
n	Throughout this section, the individual ADCE units are identified by the index “n” (n = 0); for example, ADCEnPWDVCR indicates the PWM-Diag virtual channel register.
m	Throughout this section, the individual physical channels of ADCEn are identified by the index “m”; for example, ANInm indicates an analog input pin.
j	Throughout this section, the individual virtual channels of ADCEn are identified by the index “j” (j = 0 to 20); for example, ADCEnVCRj indicates the virtual channel register j.
x	Throughout this section, the individual general purpose scan groups (SG) of ADCEn are identified by the index “x” (x = 1 to 3); for example, ADCEnSGSTCRx indicates the scan group x start control register.

Isolated-Area supply voltage monitoring

Table 45.2, A/D Converter channels shows the number of A/D converter channels that are available as external pin.

All devices feature an additional channel (m = 20) that internally connects to the supply voltage of the Isolated-Area. This channel can be used as a consistency check measurement to confirm the A/D Converter reference voltage.

Scan groups overview

The following table gives an overview about the usage of the five scan groups.

Table 45.4 Scan groups overview

Scan group	Purpose	Hardware trigger sources		Scan group interrupts
		Internal	External*1	
0	Temperature measurement	TAUBn interrupts	–	INTADCE0SGTSN
x = 1	General purpose	TAUBn interrupts	• Trigger input ADCE0TRGI1 • Interrupts INTP0, INTP1, INTP2	INTADCE0SG1
x = 2		TAUBn interrupts	• Trigger input ADCE0TRGI2 • Interrupts INTP0, INTP1, INTP2	INTADCE0SG2
x = 3		TAUBn interrupts	• Trigger input ADCE0TRGI3 • Interrupts INTP0, INTP1, INTP2	INTADCE0SG3
4	PWM Diagnosis	Fixed for PWM Diagnosis	–	–

Note 1. When the external trigger pin is used, the noise filter for the port needs to be set. For details, refer to Section 2.6, Noise Filter and Edge Level Detection Circuit.

45.1.3 Register addresses

All A/D Converters register addresses are given as address offsets from the individual base addresses <ADCEn_base>.
The <ADCEn_base> addresses of each ADCEn are listed in the following table:

Table 45.5 Register base addresses <ADCEn_base>

Base address name	Base address
<ADCE0_base>	FFF2 0000$_H$

45.1.4 Clock supply

All A/D Converters provide one clock input.

Table 45.6 Clock supply

ADCEn unit	ADCEn clock	Connected to
ADCE0	PCLK	Clock Controller C_ISO_PCLK
	CLKAD	Clock Controller C_ISO_ADCE

45.1.5 Interrupts and DMA

The A/D Converters can generate the following interrupt and DMA requests:

Table 45.7 ADCEn interrupt and DMA requests

ADCEn signals	Function	Connected to
ADCE0:		
INT_ADE	Error interrupt	Error Control Module INTADCE0ERR
INT_TSN	Temperature Sensor interrupt	Interrupt Controller INTADCE0TSN
INT_SG1	Scan group 1 interrupt	Interrupt Controller INTADCE0I1 DMA Controller trigger ID 123
INT_SG2	Scan group 2 interrupt	Interrupt Controller INTADCE0I2 DMA Controller trigger ID 124
INT_SG3	Scan group 3 interrupt	Interrupt Controller INTADCE0I3 DMA Controller trigger ID 125

45.1.6 Reset sources

The A/D Converters and their registers are initialized by the following reset signal:

Table 45.8 Reset sources

ADCEn unit	Reset signal
ADCE0	• Reset Controller SYSRES • Reset Controller ADCE0RES • reset upon wake-up from DEEPSTOP mode

CAUTION

By default the ADCE0RES reset is active.
Thus before accessing this module this reset must be released via the MRSTC register of the Reset Controller. Refer to Section 9, Reset Controller for details.

45.1.7 I/O signals

The following table shows the I/O signals of the A/D Converters.

Table 45.9 ADCEn I/O signals

ADCEn signals	Function	Connected to
ADCE0		
ADCE0I0 to ADCE0Im	Analog input channels 0 to m	Port ADCE0I0 to ADCE0Im
SG1_TRG*1	Hardware trigger for scan group 1	Port ADCE0TRIGI1
SG2_TRG*1	Hardware trigger for scan group 2	Port ADCE0TRIGI2
SG3_TRG*1	Hardware trigger for scan group 3	Port ADCE0TRIGI3

Note 1. These hardware trigger signals are input to the Hardware Trigger Expansion.

45.1.8 Internal signal connections

The following table shows the internal signal connections of the A/D Converters.

Table 45.10 ADCEn internal signals

ADCEn signals	Function	Connected to
ADCE0		
ADCE0I20	Analog input channel 20	Isolated-Area power supply ISOVDD
ADCE0I37	Analog input channel 37	Temperatur Sensor output

45.2 Overview

45.2.1 Functional Overview

ADCE has the following features.

- 10-bit/12-bit resolution
- Successive approximation conversion method
- Minimum A/D conversion time per channel: 1.15 µs
- A/D conversion is available for up to 20 channels
- Two scan modes
  Multi-cycle scan mode: Specified number of scans are executed.
  Continuous scan mode: Scans are executed with no limit of numbers.
- Asynchronous/synchronous suspend and resume function
  A processing for a scan group can interrupt a running processing for another scan group.
- Start trigger for each scan group
  Software, hardware, and external trigger can start processing of each scan group.
- Scan end interrupt and DMA transfer are supported.
  For each scan group, an interrupt request to INTC can be issued or DMA transfer can be started, each time a processing for the virtual channel indicated by the end virtual channel pointer ends, or a virtual channel ends.
- A/D conversion channel repeat function
  A/D conversion is performed for the same channel twice or four times sequentially, and the result is stored in the data register.
  (Note that the result data is not an additional value.)
- Abundant safety functions
  Abundant safety functions are provided, such as A/D converter self-diagnostic, pin-level self-diagnostic, wiring-break detection, upper limit/lower limit check for the A/D converter, overwrite check for data registers, read and clear function for data registers, and PWM-Diag function.

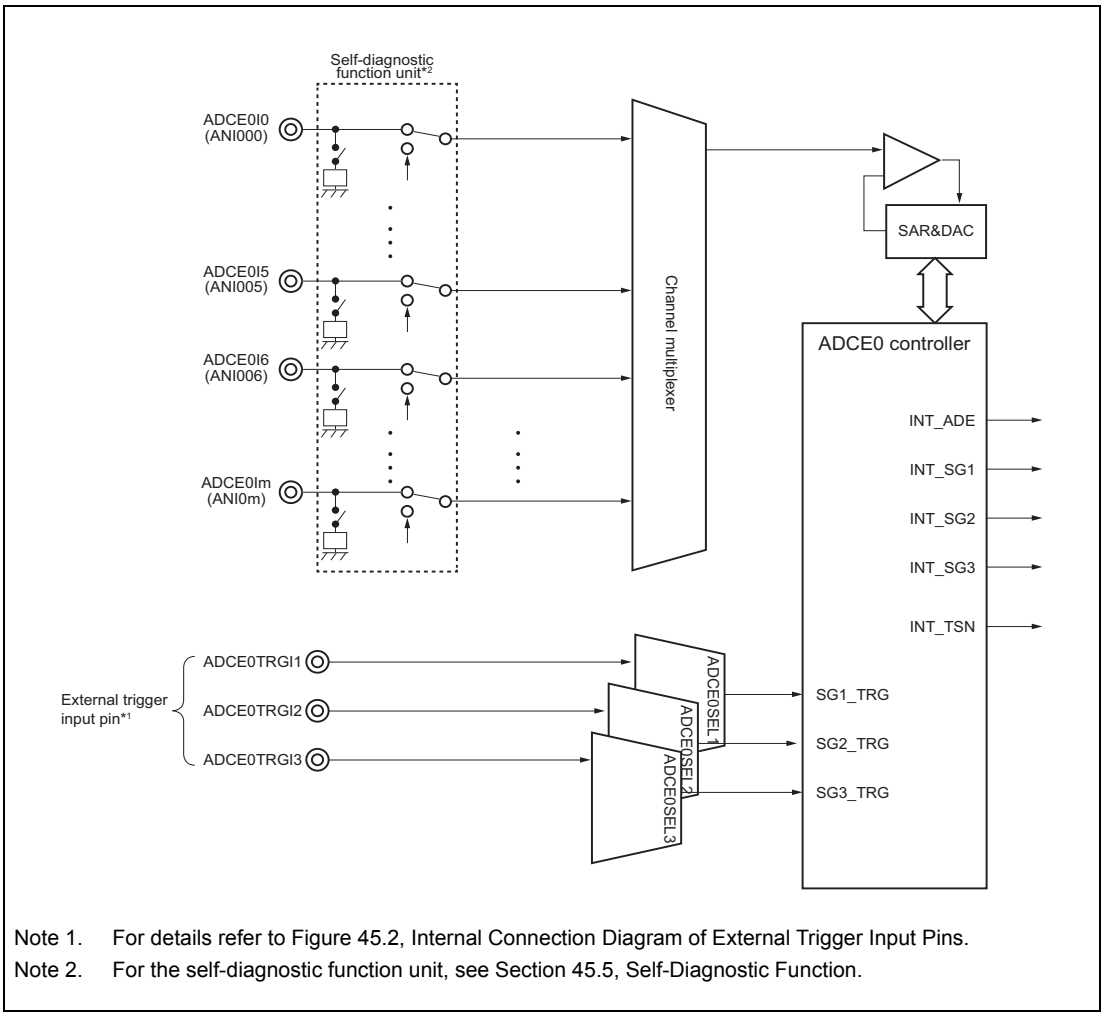
NOTE

- Physical channel (ANInm)
  Each analog input pin is assigned to a specific channel.
- Virtual channel (VCRj)
  ADCE0 has 21 virtual channels. The virtual channel specifies the physical address to be scanned.
  Scans are executed in the order from the small virtual channel number. The scan order can be arbitrarily-specified with using virtual channels. In addition, the scanned result is stored in the data register corresponding to the virtual channel.
- Scan group (SGx)
  ADCE has five scan groups:
  - SG0: dedicated scan group for temperature measurement
  - SG1, SG2, and SG3: general purpose scan group with external analog inputs
  - SG4: dedicated scan group for PWM diagnostic.

45.2.2 Block Diagram

The block diagram of ADCE0 is shown in **Figure 45.1**.

(1) Configuration of ADCE0



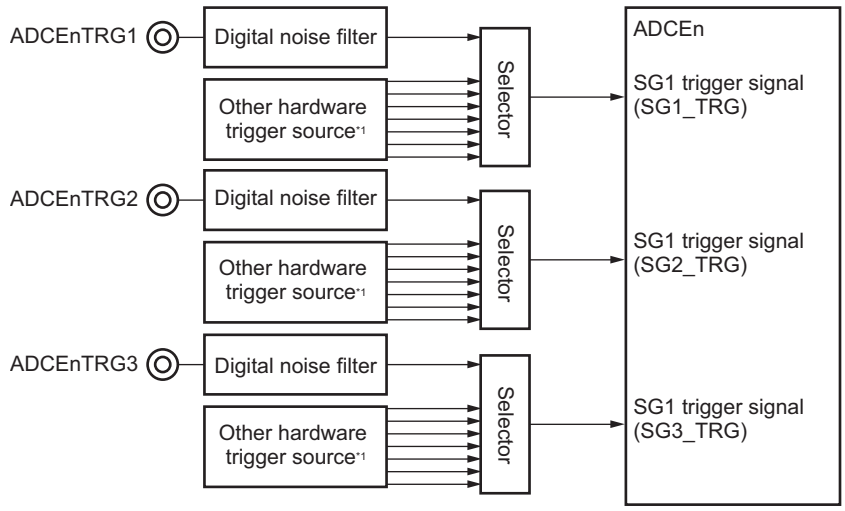
Note 1. For details refer to Figure 45.2, Internal Connection Diagram of External Trigger Input Pins.
Note 2. For the self-diagnostic function unit, see Section 45.5, Self-Diagnostic Function.

Figure 45.1 ADCE0 Block Diagram

(2) Configuration of External Trigger Input Pins

An external trigger input pin is a hardware trigger source to activate ADCEn.

The configuration of external trigger input pins is shown below.

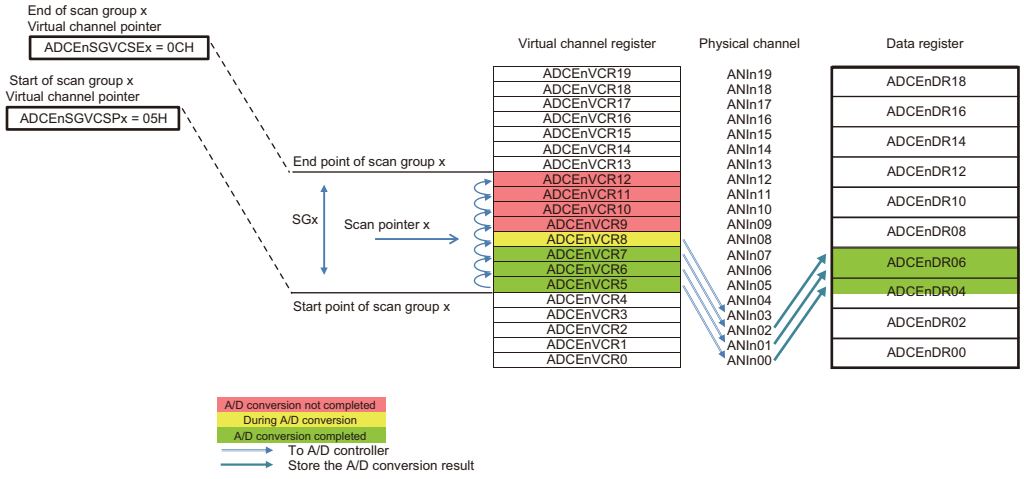


Note 1. As for the hardware trigger sources, refer to Table 45.44, List of A/D Conversion Hardware Triggers.

Figure 45.2 Internal Connection Diagram of External Trigger Input Pins

(3) Virtual Channel

The virtual channel specifies the physical address to be scanned.
The virtual channel is controlled by the ADCEnVCRj register.
A usage example of the virtual channel is shown below.



Example) When scan group x starts scanning 5th to 12th virtual channels

Note 1. The ADCEnVCRj register specifies the physical address (ANInm) to which virtual channel j executes the A/D conversion. The A/D conversion pin can be set in any order. (The figure above shows an example when ANIn0 to ANIn2 are selected.)

Note 2. The scan pointer x reads the physical address specified in the ADCEnVCRj register from “start point of scan group x” to “end point of scan group x”, and starts the A/D conversion for the read physical address.

Note 3. The scanned result of virtual channel j is stored in data register j.

Figure 45.3 Usage Example of Virtual Register

45.3 Registers

45.3.1 List of Registers

ADCE registers are listed in the following table.

Table 45.11 Register Address List (1/2)

Module Name	Register Name	Symbol	Address
• ADCE specific registers (virtual channel)			
	Virtual channel register j	ADCEnVCRj	<ADCEn_base> + j × 4_H
	PWM-Diag virtual channel register	ADCEnPWDVCR	<ADCEn_base> + $0F4_H$
	Temperature Sensor virtual channel register	ADCEnTSNVCR	<ADCEn_base> + $0F0_H$
	Data register j	ADCEnDRj	<ADCEn_base> + 100_H + j × 2_H
	Data supplementary information register j	ADCEnDIRj	<ADCEn_base> + 200_H + j × 4_H
	PWM-Diag/Temperature Sensor data register	ADCEnPWDTSNDR	<ADCEm_base> + 178_H
	PWM-Diag data supplementary information register	ADCEnPWDDIR	<ADCEn_base> + $2F4_H$
	Temperature Sensor data supplementary information register	ADCEnTSNDIR	<ADCEn_base> + $2F0_H$
• ADCE specific registers (control)			
	A/D force halt register	ADCEnADHALTR	<ADCEn_base> + 300_H
	A/D control register	ADCEnADCR	<ADCEn_base> + 304_H
	Sampling control register for SGx and PWM-Diag	ADCEnSMPCR	<ADCEn_base> + 380_H
	Sampling control register for Temperatur Sensor	ADCEnTSNSMPCR	<ADCEn_base> + 384_H
• ADCE specific registers (safety-related)			
	Safety control register	ADCEnSFTCR	<ADCEn_base> + 334_H
	Upper limit/lower limit table register 0	ADCEnULLMTBR0	<ADCEn_base> + 338_H
	Upper limit/lower limit table register 1	ADCEnULLMTBR1	<ADCEn_base> + $33C_H$
	Upper limit/lower limit table register 2	ADCEnULLMTBR2	<ADCEn_base> + 340_H
	Error clear register	ADCEnECR	<ADCEn_base> + 344_H
	Upper limit/lower limit error register	ADCEnULER	<ADCEn_base> + 348_H
	Overwrite error register	ADCEnOWER	<ADCEn_base> + $34C_H$
	Scan Groups Priority level control register	ADCEnSGPRCR	<ADCEn_base> + $38C_H$
• Scan group unique registers			
	Scan Group x start control register	ADCEnSGSTCRx	<ADCEn_base> + x × 40_H + 400_H
	Scan Group x control register	ADCEnSGCRx	<ADCEn_base> + x × 40_H + 408_H
	PWM-Diag scan group control register	ADCEnPWDSGCR	<ADCEn_base> + 508_H
	Temperature Sensor scan group control register	ADCEnTSNSGCR	<ADCEn_base> + 408_H
	Scan Group x start virtual channel pointer	ADCEnSGVCSPx	<ADCEn_base> + x × 40_H + $40C_H$
	Scan Group x end virtual channel pointer	ADCEnSGVCEPx	<ADCEn_base> + x × 40_H + 410_H
	Scan Group x multicycle register	ADCEnSGMCYCRx	<ADCEn_base> + x × 40_H + 414_H
	Scan Group x end flag clear register	ADCEnSGSEFCRx	<ADCEn_base> + x × 40_H + 418_H
	PWM-Diag scan end flag clear register	ADCEnPWDSGSEFCR	<ADCEn_base> + 518_H
	Temperature Sensor end flag clear register	ACDEnTSNSGSEFCR	<ADCEn_base> + 418_H
	Scan Group status register	ADCEnSGSTR	<ADCEn_base> + 308_H

Table 45.11 Register Address List (2/2)

Module Name	Register Name	Symbol	Address
• H/W trigger specific register			
	A/D Conversion SGx trigger select control register	ADCEnSGTSELx	<ADCEn_base> + x × 40_H + $41C_H$
SELB	External interrupts edge selection register	ADCEnINTPEDGCTL	FFC0 $603C_H$
	A/D Conversion temperature measurement trigger select control register	ADCEnTSNSGTSEL	<ADCEn_base> + $41C_H$
• Self-Diagnostic specific registers			
	Self-Diagnostic voltage level control register	ADCEnDGCTL0	<ADCEn_base> + 350_H
	Self-Diagnostic control register	ADCEnDGCTL1	<ADCEn_base> + 354_H
	Pull Down control register 2	ADCEnPDCTL2	<ADCEn_base> + $35C_H$
• Emulation specific register			
	Emulation control register	ADCEnEMU	<ADCEn_base> + 388_H

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

Note that index j is representing a double-digit number for this register.

45.3.2 ADCE Specific Registers (Virtual Channel)

45.3.2.1 ADCEnVCRj — Virtual Channel Register j

This register specifies the physical channel to be scanned to the virtual channel.

Access: This register can be read/written in 32/16-bit units
ADCEnVCRj can be read/written in 32-bit units
ADCEnVCRjL can be read/written in 16-bit units

Address: ADCEnVCRj:<ADCEn_base> + j × 4_H
ADCEnVCRjL:<ADCEn_base> + j × 4_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ADIE	ULS[1:0]		GCTRL[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.12 ADCEnVCRj register contents

Bit Position	Bit Name	Function
31 to 9	Reserved	When written, write the initial value.
8	ADIE	A/D Conversion End Interrupt Enable 0: INT_SGx is not output when A/D conversion for the virtual channel ends in SGx. 1: INT_SGx is output when A/D conversion for the virtual channel ends in SGx.
7 to 6	ULS[1:0]	Upper Limit/Lower Limit Table Select 00: Upper limit and lower limit are not checked. 01: Upper limit and lower limit are checked for ADCEnULLMTBR0. 10: Upper limit and lower limit are checked for ADCEnULLMTBR1. 11: Upper limit and lower limit are checked for ADCEnULLMTBR2.
5 to 0	GCTRL[5:0]	Physical Channel Select $0F_H$: Isolated-Area power supply ISOVDD is selected. 10_H to 23_H: Corresponding ANInm is selected. 24_H: Self-diagnostic channel is selected. 25_H: Temperature Sensor output is selected. Others: Setting prohibited **Note:** These bits shall set the physical channel which is supported. See Table 45.9, ADCEn I/O signals.

CAUTION

To prevent malfunction, ADCEnVCRj should be set when SGACT[5:0] of applicable scan groups is 0 (before scan groups are started) and TRGMD of applicable scan groups is 0.

Note that index j is representing a double-digit number for this register.

Table 45.13 Selection of Physical Channels

GCTRL5	GCTRL4	GCTRL3	GCTRL2	GCTRL1	GCTRL0	Analog Input Pin to be Selected
0	1	0	0	0	0	ADCEnI0 (Physical channel ANIn00)
0	1	0	0	0	1	ADCEnI1 (Physical channel ANIn01)
0	1	0	0	1	0	ADCEnI2 (Physical channel ANIn02)
0	1	0	0	1	1	ADCEnI3 (Physical channel ANIn03)
0	1	0	1	0	0	ADCEnI4 (Physical channel ANIn04)
0	1	0	1	0	1	ADCEnI5 (Physical channel ANIn05)
0	1	0	1	1	0	ADCEnI6 (Physical channel ANIn06)
0	1	0	1	1	1	ADCEnI7 (Physical channel ANIn07)
0	1	1	0	0	0	ADCEnI8 (Physical channel ANIn08)
0	1	1	0	0	1	ADCEnI9 (Physical channel ANIn09)
0	1	1	0	1	0	ADCEnI10 (Physical channel ANIn10)
0	1	1	0	1	1	ADCEnI11 (Physical channel ANIn11)
0	1	1	1	0	0	ADCEnI12 (Physical channel ANIn12)
0	1	1	1	0	1	ADCEnI13 (Physical channel ANIn13)
0	1	1	1	1	0	ADCEnI14 (Physical channel ANIn14)
0	1	1	1	1	1	ADCEnI15 (Physical channel ANIn15)
1	0	0	0	0	0	ADCEnI16 (Physical channel ANIn16)
1	0	0	0	0	1	ADCEnI17 (Physical channel ANIn17)
1	0	0	0	1	0	ADCEnI18 (Physical channel ANIn18)
1	0	0	0	1	1	ADCEnI19 (Physical channel ANIn19)
0	0	1	1	1	1	ADCEnI20 (Isolated-Area power supply ISOVDD)
1	0	0	1	0	1	ADCEnI37 (Temperature Sensor output)*1
1	0	0	1	0	0	Self-diagnostic channel
Other than above						Setting prohibited

Note 1. The temperature measurement uses a dedicated virtual channel, that is controlled by a separate register, refer to Section 45.3.2.3, ADCEnTSNVCR — Temperature Sensor Virtual Channel Register.

45.3.2.2 ADCEnPWDVCR — PWM-Diag Virtual Channel Register

This register controls the PWM-Diag virtual channel.

This register is written by the A/D Converter trigger control of the PWM Diagnostic module. Thus this register shows the A/D conversion parameters, sent by the PWM Diagnostic module.

Access: ADCEnPWDVCR can be read only in 32-bit units.
ADCEnPWDVCRL can be read only in 16-bit units.
ADCEnPWDVCRLL can be read only in 8-bit units.

Address: ADCEnPWDVCR: <ADCEn_base> + $0F4_H$
ADCEnPWDVCRL: <ADCEn_base> + $0F4_H$
ADCEnPWDVCRLL: <ADCEn_base> + $0F4_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ULS[1:0]		GCTRL[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.14 ADCEnPWDVCR register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, an initial value is returned. Writing is ignored.
7 to 6	ULS[1:0]	Upper Limit/Lower Limit Table selection of PWM Diagnostic A/D conversion 00: Upper limit and lower limit are not checked. 01: Upper limit and lower limit are checked for ADCEnULLMTBR0. 10: Upper limit and lower limit are checked for ADCEnULLMTBR1. 11: Upper limit and lower limit are checked for ADCEnULLMTBR2.
5 to 0	GCTRL[5:0]	Physical channel selection of PWM Diagnostic A/D conversion

CAUTION

To prevent malfunction, ADCEnPWDVCR should be set when SGACT[5:0] of applicable scan groups is 0 (before scan groups are started) and TRGMD of applicable scan groups is 0.

45.3.2.3 ADCEnTSNVCR — Temperature Sensor Virtual Channel Register

This register controls the Temperature Sensor virtual channel.

Access: ADCEnTSNVCR can be read/written in 32-bit units.
ADCEnTSNVCRL can be read/written in 16-bit units.
ADCEnTSNVCRLL can be read/written in 8-bit units.

Address: ADCEnTSNVCR: <ADCEn_base> + $0F0_H$
ADCEnTSNVCRL: <ADCEn_base> + $0F0_H$
ADCEnTSNVCRLL: <ADCEn_base> + $0F0_H$

Initial value: $0000\ 0025_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TSNULS[1:0]		TSNGCTRL[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.15 ADCEnTSNVCR register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When read, an initial value is returned. Writing is ignored.
7 to 6	TSNULS[1:0]	Upper Limit/Lower Limit Table selection of temperature measurement A/D Converter trigger control. 00: Upper limit and lower limit are not checked. 01: Upper limit and lower limit are checked for ADCEnULLMTBR0. 10: Upper limit and lower limit are checked for ADCEnULLMTBR1. 11: Upper limit and lower limit are checked for ADCEnULLMTBR2.
5 to 0	TSNGCTRL[5:0]	Physical channel selection of Temperature Sensor. The physical channel number for the Temperature Sensor output is 37, thus set GCTRL[5:0] = 100101_B.

CAUTION

To prevent malfunction, ADCEnTSNVCR should be set when SGACT[5:0] of applicable scan groups is 0 (before scan groups are started) and TRGMD of applicable scan groups is 0.

45.3.2.4 ADCEnDRj — Data Register j

This register is a 32-bit/16-bit read-only register, which stores the A/D conversion results corresponding to ADCEnVCRj and ADCEnVCR(j+1). As the A/D conversion results, the conversion result for ADCEnVCR(j+1) is stored in the upper bits (ADCEnDR(j+1)[15:0]), and the conversion result for ADCEnVCRj is stored in the lower bits (ADCEnDRj[15:0]).

Access: ADCEnDRj can be read only in 32-bit units.
ADCEnDRjL can be read only in 16-bit units.
ADCEnDRjH can be read only in 16-bit units.

Address: ADCEnDRj : <ADCEn_base> + 100H +j × 2_H
ADCEnDRjL : <ADCEn_base> + 100H + j × 2_H
ADCEnDRjH : <ADCEn_base> + 100H + j × 2_H + 2_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DR(j+1)[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DRj[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.16 ADCEnDRj register contents

Bit Position	Bit Name	Function
31 to 16	DR(j+1)[15:0]	These bits are used to store the A/D conversion result data. (The A/D conversion result for the channel set in ADCEnVCR(j+1)are transferred.)
15 to 0	DRj[15:0]	These bits are used to store the A/D conversion result data. (The A/D conversion result for the channel set in ADCEnVCRj are transferred.)

NOTE

By controlling ADCEnADCR.CRAC and ADCEnADCR:CTYP, the data format of this register becomes as follows:

- When ADCEnADCR.CRAC = 0 and ADCEnADCR.CRAC = 0, right alignment is used.→right alignment is used.
  →The A/D conversion result for ADCEnVCR(j+1) is transferred to bits 27 to 16, and the A/D conversion result for ADCEnVCRj is transferred to bits 11 to 0.
- When ADCEnADCR.CTYP = 0 and ADCEnADCR.CRAC = 1, left alignment is used.→left alignment is used.
  →The A/D conversion result for ADCEnVCR(j+1) is transferred to bits 31 to 20, and the A/D conversion result for ADCEnVCRj is transferred to bits 15 to 4.
- When ADCEnADCR.CTYP = 1 and ADCEnADCR.CRAC = 0, right alignment is used.→right alignment is used.
  →The A/D conversion result for ADCEnVCR(j+1) is transferred to bits 25 to 16, and the A/D conversion result for ADCEnVCRj is transferred to bits 9 to 0.
- When ADCEnADCR.CTYP = 1 and ADCEnADCR.CRAC = 1, left alignment is used.→left alignment is used.
  →The A/D conversion result for ADCEnVCR(j+1) is transferred to bits 31 to 22, and the A/D conversion result for ADCEnVCRj is transferred to bits 15 to 6.

Note that index j is representing a double-digit number for this register.

45.3.2.5 ADCEnDIRj — Data Supplementary Information Register j

This register is a 32-bit read-only register, which stores the A/D conversion result for ADCEnDRj and information incidental to the A/D converted value.

As the A/D conversion result, the ADCEnDRj value is transferred. As information incidental to the A/D converted value, information about the write flag (WFLG), and the physical channel (ID[5:0]) is transferred. The data format of the A/D conversion result stored in ADCEnDIRj is the same as the one for the ADCEnDRj register.

Access: This register can be read only in 32-bit units

Address: <ADCEn_base> + 200_H + j × 4_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	WFLG	—	—	—	ID[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.17 ADCEnDIRj register contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When written, write the initial value.
25	WFLG	Write Flag 0: The conversion result DR[15]. has been read via the ADCEnDRj or ADCEnDIRj register. 1: A new A/D conversion result has been stored in DR[15].
24 to 20	Reserved	When written, write the initial value.
21 to 16	ID[5:0]	These bits store the physical channel number (GCTRL) corresponding to the conversion result. The physical channel number to be stored is the number corresponding to the most recent conversion result.
15 to 0	DR[15:0]	These bits are used to store the A/D conversion result.

Note that index j is representing a double-digit number for this register.

45.3.2.6 ADCEnPWDTSNDR — PWM-Diag and temperature measurement Data Register

This register is a 32-bit/16-bit read-only register, which stores the A/D conversion results corresponding to the PWM-Diag and temperature measurement.

As the A/D conversion results, the conversion result for the PWM-Diag (PWDDR[15:0]) is stored in the upper bits.

The conversion result for the temperature measurement (TSNDR[15:0]) is stored in the lower bits.

Access: ADCEnPWDTSNDR can be read only in 32-bit units
ADCEnPWDTSNDRL can be read only in 16-bit units
ADCEnPWDTSNDRH can be read only in 16-bit units

Address: ADCEnPWDTSNDR: <ADCEn_base> + 178_H
ADCEnPWDTSNDRL : <ADCEn_base> + 178_H
ADCEnPWDTSNDRH : <ADCEn_base> + $17A_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	PWDDR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSNDR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.18 ADCEnPWDTSNDR register contents

Bit Position	Bit Name	Function
31 to 16	PWDDR[15:0]	These bits are used to store the A/D conversion result data for the PWM-Diag.
15 to 0	TSNDR[15:0]	These bits are used to store the A/D conversion result data for the temperature measurement.

NOTE

The data format of this register is controlled by ADCEnADCR.CRAC and ADCEnADCR.CTYP, as shown below.

- ADCEnADCR.CTYP = 0, ADCEnADCR.CRAC = 0; right alignment is used.→ right alignment is used.
  → The A/D conversion result for ADCEnPWDVCR is transferred to bits 27 to 16.
- ADCEnADCR.CTYP = 0, ADCEnADCR.CRAC = 1 → left alignment is used.→ left alignment is used.
  → The A/D conversion result for ADCEnPWDVCR is transferred to bits 31 to 20.
- ADCEnADCR.CTYP = 1, ADCEnADCR.CRAC = 0 → right alignment is used.→ right alignment is used.
  → The A/D conversion result for ADCEnPWDVCR is transferred to bits 25 to 16.
- ADCEnADCR.CTYP = 1, ADCEnADCR.CRAC = 1 → left alignment is used.→ left alignment is used.
  → The A/D conversion result for ADCEnPWDVCR is transferred to bits 31 to 22.

45.3.2.7 ADCEnPWDDIR — PWM-Diag Data Supplementary Information Register

This register is a 32-bit read-only register, which stores the A/D conversion result when PWM-Diag is used, and information incidental to the A/D converted value. As the A/D conversion result, the ADCEnPWDTSNDR.PWDDR[15:0] value is transferred. As supplementary information to the A/D converted value, the write flag (WFLG), and physical channel (ID[5:0]) are transferred. The format of the A/D conversion result stored in ADCEnPWDDIR is the same as that of the data in ADCEnPWDTSNDR.PWDDR[15:0].

Access: This register can be read only in 32-bit units

Address: <ADCEn_base> + $2F4_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	WFLG	—	—	—	ID[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PWDDR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.19 ADCEnPWDDIR Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When written, write the initial value.
25	WFLG	Write Flag 0: ADCEnPWDTSNDR or ADCEnPWDDIR is read (cleared when read). 1: The A/D converted value is stored in ADCEnPWDTSNDR (set when the value is stored).
24 to 22	Reserved	When written, write the initial value.
21 to 16	ID[5:0]	These bits store the physical channel number (GCTRL) corresponding to the conversion result. The physical channel number to be stored is the number corresponding to the most recent conversion result.
15 to 0	PWDDR[15:0]	These bits are used to store the A/D conversion result for PWM-Diag.

45.3.2.8 ADCEnTSNDIR — Temperature Measurement Data Supplementary Information Register

This register is a 32-bit read-only register, which stores the A/D conversion result when temperature measurement is used, and information incidental to the A/D converted value. As the A/D conversion result, the ADCEnPWDTSNDR.TSNDR[15:0] value is transferred. As supplementary information to the A/D converted value, the write flag (WFLG), and physical channel (ID[5:0]) are transferred. The format of the A/D conversion result stored in ADCEnTSNDIRm is the same as that of the data in ADCEnPWDTSNDR.TSNDR[15:0].

Access: This register can be read only in 32-bit units

Address: <ADCEn_base> + $2F0_H$

Initial value: $0025\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	WFLG	—	—	—	ID[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSNDR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.20 ADCEnTSNDIR Register Contents

Bit Position	Bit Name	Function
31 to 26	Reserved	When written, write the initial value.
25	WFLG	Write Flag 0: The conversion result TSNDR[15]. has been read via the ADCEnPWDTSNDR or ADCEnTSNDIR register. 1: A new A/D conversion result has been stored in TSNDR[15].
24 to 22	Reserved	When written, write the initial value.
21 to 16	ID[5:0]	Holds the physical channel number (GCTRL) corresponding to the conversion result. The physical channel number of the temperature measurement is 37, thus ID[5:0] = 100101_B.
15 to 0	TSNDR[15:0]	These bits are used to store the A/D conversion result for temperature measurement.

45.3.3 ADCE Specific Registers (Control)

45.3.3.1 ADCEnADHALTR — A/D Force Halt Register

This register is a 32/16/8-bit write-only register, which halts conversion for all SGs of ADCEn. The read value is always 0

Access: This register can be written only in 32/16/ 8-bit units
ADCEnADHALTR can be written only in 32-bit units
ADCEnADHALTRL can be written only in 16-bit units
ADCEnADHALTRLL can be written only in 8-bit units

Address: ADCEnADHALTR:<ADCEn_base> + 300_H
ADCEnADHALTRL:<ADCEn_base> + 300_H
ADCEnADHALTRLL:<ADCEn_base> + 300_H

Initial value: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value																
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	HALT
Initial value																
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 45.21 ADCEnADHALTR register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, an undefined value is returned. The write value should be 0.
0	HALT	ADCE Force Halt Trigger All scan groups are halted and initialized, and ADCE becomes idle state. Writing of 0: No effect Writing of 1: Scan groups are halted.

45.3.3.2 ADCEnADCR — A/D Control Register

This register provides common control settings for all channels and scan groups.

Access: ADCEnADCR can be read/written in 32-bit units
ADCEnADCRL can be read/written in 16-bit units
ADCEnADCRLL can be read/written in 8-bit units

Address: ADCEnADCR: <ADCEn_base> + 304_H
ADCEnADCRL: <ADCEn_base> + 304_H
ADCEnADCRLL: <ADCEn_base> + 304_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DGON	TSN SELF DIAG	CRAC	CTYP	—	—	SUSMTD[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Table 45.22 ADCEnADCR Register Contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When written, write the initial value.
7	DGON	Self-Diagnostic Voltage Standby Control 0: The self-diagnostic voltage circuit is turned off. 1: The self-diagnostic voltage circuit is turned on.
6	TSNSELFDIAG	Temperature Sensor self-diagnostic control 0: Temperature Sensor self-diagnostic is turned off. 1: Temperature Sensor self-diagnostic is turned on.
5	CRAC	Alignment Control 0: PWDDR[15:0] and ADCEnDRj are set to right align. 1: PWDDR[15:0] and ADCEnDRj are set to left align.
4	CTYP	12/10 Bit Select Mode 0: 12-bit mode 1: 10-bit mode
3, 2	Reserved	When written, write the initial value.
1, 0	SUSMTD [1:0]	Suspend Mode Select Selects the suspend method. 00_B: Synchronous suspend when a higher-priority SG or SVSTOP is generated 01_B: SG0 (TSN) and SG1 are suspended asynchronously, other scan groups are suspended synchronously. SVSTOP always interrupts synchronously. 10_B: Asynchronous suspend when a higher-priority SG or SVSTOP interrupts. 11_B: Setting prohibited

CAUTION

To prevent malfunction, perform the ADCEnADCR settings when SGACT[5:0] of all scan groups is 0 (before scan groups are started), and TRGMD of all scan groups is 0.

NOTE

Suspend Method
These bits are used to select the suspend method used when a higher-priority scan group interrupts a lower-priority scan group.

- Synchronous suspend:
  If a request from a higher-priority SG occurs while a lower-priority SG is being processed, the A/D conversion for the higher-priority SG is performed after the ongoing A/D conversion of a channel is completed. After processing for the higher-priority SG is completed, the suspended channel processing for the lower-priority SG is resumed.
- Asynchronous suspend:
  If a request from a higher-priority SG occurs while a lower-priority SG is being processed, the ongoing channel processing is suspended, and then the A/D conversion for the higher-priority SG is performed. After processing for the higher-priority SG is completed, the suspended A/D channel conversion for the lower-priority SG is resumed.

For details, see Figure 45.14, Example of Synchronous Suspend and Resume Operation and Figure 45.15, Example of Asynchronous Suspend and Resume Operation.

45.3.3.3 ADCEnSMPCR — Sampling Control Register for SGx and PWM-Diag

This register selects the sampling time of the general purpose scan groups SGx and the PWM-Diag scan group SG4.

Access: ADCEnSMPCR can be read or written in 32-bit units.
ADCEnSMPCRL can be read or written in 16-bit units.
ADCEnSMPCRLL can be read or written in 8-bit units.

Address: ADCEnSMPCR: <ADCEn_base> + 380_H
ADCEnSMPCRL: <ADCEn_base> + 380_H
ADCEnSMPCRLL: <ADCEn_base> + 380_H

Initial value: 0000 0018_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	SMPT[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.23 ADCEnSMPCR register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When written, write the initial value.
7 to 0	SMPT[7:0]	These bits are used to set the sampling time (the number of CLKAD cycles). 12_H: 18 cycles (CLKAD = 8 MHz to 32 MHz) 18_H: 24 cycles (CLKAD = 8 MHz to 40 MHz) Settings other than above are prohibited.

CAUTION

To prevent malfunction, perform ADCETLnSMPCR settings when SGACT[5:0] of all scan groups is 0 (before scan groups are started), and TRGMD of all scan groups is 0.

45.3.3.4 ADCEnTSNSMPCR — Sampling Control Register for Temperature Sensor

This register selects the sampling time of the temperature measurement.

Access: ADCEnTSNSMPCR can be read or written in 32-bit units.
ADCEnTSNSMPCRL can be read or written in 16-bit units.
ADCEnTSNSMPCRLL can be read or written in 8-bit units.

Address: ADCEnTSNSMPCR: <ADCEn_base> + 384_H
ADCEnTSNSMPCRL: <ADCEn_base> + 384_H
ADCEnTSNSMPCRLL: <ADCEn_base> + 384_H

Initial value: 0000 $00F0_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TSNSMPT[7:0]							
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.24 ADCEnTSNSMPCR register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When written, write the initial value.
7 to 0	TSNSMPT[7:0]	These bits are used to set the sampling time (the number of CLKAD cycles). The sampling time for the temperature measurement must be 6 µs. Examples: • for CLKAD = 40 MHz: 6.0 µs = 240/40 MHz, thus SMPT[7:0] = $F0_H$ • for CLKAD = 32 MHz: 6.0 µs = 192/32 MHz, thus SMPT[7:0] = $C0_H$ • C0H: 192 cycles, sampling time = 6.0 µs with CLKAD = 32 MHz

CAUTION

To prevent malfunction, perform ADCETLnTSNSMPCR settings when SGACT[5:0] of all scan groups is 0 (before scan groups are started), and TRGMD of all scan groups is 0.

45.3.4 ADCE Specific Registers (Safety-related)

45.3.4.1 ADCEnSFTCR — Safety Control Register

This register is a register regarding safety control.

Access: ADCEnSFTCR can be read or written in 32-bit units.
ADCEnSFTCRL can be read or written in 16-bit units.
ADCEnSFTCRLL can be read or written in 8-bit units.

Address: ADCEnSFTCR: <ADCEn_base> + 334_H
ADCEnSFTCRL: <ADCEn_base> + 334_H
ADCEnSFTCRLL: <ADCEn_base> + 334_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	RDCLRE	ULEIE	OWEIE	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 45.25 ADCEnSFTCR register contents

Bit Position	Bit Name	Function
31 to 5	Reserved	When written, write the initial value.
4	RDCLRE	Read & Clear Enable When the A/D conversion result is read, this bit selects whether the A/D conversion result is cleared by hardware. 0: ADCEnPWDTSNDR/ADCEnDRj and ADCEnPWDDIR/ADCEnTSNDIR/ADCEnDIRj are not cleared when ADCEnPWDTSNDR/ADCEnDRj or ADCEnPWDDIR/ADCEnTSNDIR/ADCEnDIRj is read. 1: ADCEnPWDTSNDR/ADCEnDRj and ADCEnPWDDIR/ADCEnTSNDIR/ADCEnDIRj are cleared when ADCEnPWDTSNDR/ADCEnDRj or ADCEnPWDDIR/ADCEnTSNDIR/ADCEnDIRj is read. **CAUTION** WFLG of ADCEnDIRj is cleared when ADCEnDRj or ADCEnDIRj is read.
3	ULEIE	Upper Limit/Lower Limit Error Interrupt Enable 0: Disabled 1: Enabled
2	OWEIE	Overwrite Error Interrupt Enable 0: Disabled 1: Enabled
1, 0	Reserved	When written, write the initial value.

CAUTION

To prevent malfunction, perform the ADCEnSFTCR settings when SGACT[5:0] of all scan groups is 0 (before scan groups are started), and TRGMD of all scan groups is 0.

45.3.4.2 ADCEnULLMTBR0 to 2 — Upper Limit/Lower Limit Table Register 0 to 2

These registers are used to set the upper limit and lower limit of an A/D converted value. Any of ADCEnULLMTBR0 to ADCEnULLMTBR2 is specified by setting ADCEnPWDVCR.ULS[1:0] and ADCEnVCRj.USL[1:0] and compared with ADCEnPWDTSNDR and ADCEnDRj.

Access: This register can be read or written in 32-bit units

Address: ADCEnULLMTBR0: <ADCEn_base> + 338_H
ADCEnULLMTBR1: <ADCEn_base> + $33C_H$
ADCEnULLMTBR2: <ADCEn_base> + 340_H

Initial value: $FFF0\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ULMTB[11:0]												—	—	—	—
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LLMTB[11:0]												—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R

Table 45.26 ADCEnULLMTBR register contents

Bit Position	Bit Name	Function
31 to 20	ULMTB[11:0]	Upper Limit Table Specify the upper limit of an A/D converted value. The upper limit error (ADCEnULER.UE) is set when the following condition is met: ULMTB[11:0] < A/D converted value
19 to 16	Reserved	When written, write the initial value.
15 to 4	LLMTB[11:0]	Lower Limit Table Specify the lower limit of A/D converted value. The lower limit error (ADCEnULER.LE) is set when the following condition is met: LLMTB[11:0] > A/D converted value
3 to 0	Reserved	When written, write the initial value.

CAUTION

- When A/D conversion is executed in 10-bit mode (ADCEnADCR.CTYP=1), ULMTB[1:0] and LLMTB[1:0] should be set to 11_B and 00_B, respectively.
- To prevent malfunction, perform the settings of ADCEnULLMTBRn when SGACT[5:0] of all scan groups that use the corresponding table (n) is 0 (before the scan groups are started), and TRGMD of all this scan groups is 0.
- The upper-limit table (ULMTB[11:0]) must be greater than the lower-limit table (LLMTB[11:0]).

45.3.4.3 ADCEnECR — Error Clear Register

This register controls error clear. The read value is always 0.

Access: ADCEnECR can be written in 32-bit units.
ADCEnECRL can be written in 16-bit units.
ADCEnECRLL can be written in 8-bit units.

Address: ADCEnECR: <ADCEn_base> + 344_H
ADCEnECRL: <ADCEn_base> + 344_H
ADCEnECRLL: <ADCEn_base> + 344_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ULEC	OWEC	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 45.27 ADCEnECR register contents

Bit Position	Bit Name	Function
31 to 4	Reserved	When read, an undefined value is returned. The write value should be 0.
3	ULEC	ULEC clear the error flags. ULEC affect to following error flags. • Upper Limit Error Flag (ADCEnULER.UE) • Lower Limit Error Flag (ADCEnULER.LE) • Upper/Lower Limit Error Capture Bits (ADCEnULER.ULECAP[5:0]) • Scan Group Bits (ADCEnULE.ULSG[1:0]) Writing of 0: No effect. Writing of 1: Cleared.
2	OWEC	OWEC clear the overwrite error flags. OWEC affect to following flags. • Overwrite Error (ADCEnOWER.OWE) • Overwrite Error (ADCEnOWER.OWECAP[5:0]) Writing of 0: No effect. Writing of 1: Cleared.
1, 0	Reserved	When read, the value returned is 0.

45.3.4.4 ADCEnULER — Upper Limit/Lower Limit Error Register

This register is a 32/16-bit read-only register, which indicates the upper limit/lower limit errors.

Access: ADCEnULER can be read only in 32-bit units.
ADCEnULERL can be read only in 16-bit units.

Address: ADCEnULER: <ADCEn_base> + 348_H
ADCEnULERL: <ADCEn_base> + 348_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	UE	LE	ULSG[1:0]		—	—	—	—	—	—	ULECAP[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.28 ADCEnULER register contents (1/2)

Bit Position	Bit Name	Function
31 to 16	Reserved	When written, write the initial value.
15	UE	Upper Limit Error Generation Flag 0: No upper limit error 1: Upper limit error. Setting condition: The A/D converted value exceeds the range of the specified upper limit table. Clearing condition: 1 is written to ADCEnECR.ULEC.
14	LE	Lower Limit Error Generation Flag 0: No lower limit error 1: Lower limit error. Setting condition: The A/D converted value exceeds the range of the specified lower limit table. Clearing condition: 1 is written to ADCEnECR.ULEC.
13, 12	ULSG[1:0]	Scan Group where an Upper Limit/Lower Limit Error Occurs 00: No upper limit/lower limit error occurred. 01: A scan group where an upper limit/lower limit error occurred is SG1 to SG3. 10: A scan group where an upper limit/lower limit error occurred is SG4 (PWM-Diag). 11: A scan group where an upper limit/lower limit error occurred is SG0 (TSN) Capturing condition: When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0. Clearing condition: 1 is written to ADCEnECR.ULEC.
11 to 7	Reserved	When written, write the initial value.
6	Reserved	When written, write the initial value.

Table 45.28 ADCEnULER register contents (2/2)

Bit Position	Bit Name	Function
5 to 0	ULECAP[5:0]	Upper Limit/Lower Limit Error Capture The physical channel is captured when an upper limit/lower limit error occurred. Capturing condition: When the A/D converted value exceeds the range of the specified upper- or lower-limit table while UE = 0 and LE = 0. Clearing condition: 1 is written to ADCEnECR.ULEC.

CAUTION

ADCEnULER is updated when the A/D converted value is written to ADCEnDRj or ADCEnPWDTSNDR.

45.3.4.5 ADCEnOWER — Overwrite Error Register

This register is a 32/16/8-bit read-only register, which indicates an overwrite error. The target of an overwrite error is SG0 (TSN), SG1 to SG3, and not SG4 (PWM-Diag).

Access: ADCEnOWER can be read in 32-bit units.
ADCEnOWERL can be read in 16-bit units.
ADCEnOWERLL can be read in 8-bit units.

Address: ADCEnOWER: <ADCEn_base> + $34C_H$
ADCEnOWERL: <ADCEn_base> + $34C_H$
ADCEnOWERLL: <ADCEn_base> + $34C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OWE	—	OWECAP[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.29 ADCEnOWER register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When written, write the initial value.
7	OWE	Overwrite Error Generation Flag 0: No Error 1: There is an error. Setting condition: WFLG = 1, and the A/D converted value is written to ADCEnDRj and ADCEnPWDTSNDR.TSNDR[15:0]. Clearing condition: 1 is written to OWEC.
6	Reserved	When written, write the initial value.
5 to 0	OWECAP[5:0]	Overwrite Error Capture The virtual channel is captured when an overwrite error occurs. Capturing condition: OWE = 0 and WFLG = 1, and the A/D converted value is written to ADCEnDRj and ADCEnPWDTSNDR.TSNDR[15:0]. Clearing condition: 1 is written to OWEC.

CAUTION

ADCEnOWER is updated when the A/D converted value is written to ADCEnDRj or ADCEnPWDTSNDR.TSNDR[15:0].

45.3.4.6 ADCEnSGPRCR — Scan Groups Priority Level Control Register

This register selects the priority level order of the scan groups.

Access: ADCEnSGPRCR can be read/written in 32-bit units.

Address: <ADCEn_base> + $38C_H$

Initial value: 0004 3210_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SGPR[18:16]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SGPR[15:0]															
Initial value	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.30 ADCEnOWER register contents

Bit Position	Bit Name	Function
31 to 19	Reserved	When written, write the initial value.
18 to 0	SGPR[18:0]	Scan groups priority selection 43210_H: priority order SG4 (PWM-Diag) > SG3 > SG2 > SG1 > SG0 (temperature measurement) 32401_H: priority order SG3 > SG2 > SG4 (PWM-Diag) > SG0 (temperature measurement) > SG1 All others: Setting prohibited

45.3.5 Scan Group (SG) Unique Registers

This section describes the registers provided for each scan group.

45.3.5.1 ADCEnSGSTCRx — Scan Group x Start Control Register

This register is an 32/16/8-bit write-only register, which controls the start of scan group x. The read value is always 0.

Access: ADCEnSGSTCRx can be written in 32-bit units.
ADCEnSGSTCRxL can be rwritten in 16-bit units.
ADCEnSGSTCRxLL can be written in 8-bit units.

Address: ADCEnSGSTCRx: <ADCEn_base> + 400_H + x × 40_H
ADCEnSGSTCRxL: <ADCEn_base> + 400_H + x × 40_H
ADCEnSGSTCRxLL: <ADCEn_base> + 400_H + x × 40_H

Initial value: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value																
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SGST
Initial value																
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 45.31 ADCEnSGSTCRx register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, an undefined value is returned. The write value should be 0.
0	SGST	Scan Group Start Trigger Condition of starting scan group x: SGACT[5:0] = 0, and 1 is written to SGST.

45.3.5.2 ADCEnSGCRx — Scan Group x Control Register

This register controls scan group x.

Access: ADCEnSGCRx can be read or written in 32-bit units.
ADCEnSGCRxL can be read or written in 16-bit units.
ADCEnSGCRxLL can be read or written in 8-bit units.

Address: ADCEnSGCRx: <ADCEn_base> + x × 40_H + 408_H
ADCEnSGCRxL: <ADCEn_base> + x × 40_H + 408_H
ADCEnSGCRxLL: <ADCEn_base> + x × 40_H + 408_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	SCANMD	ADIE	SCT[1:0]		—	TRGMD
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W

Table 45.32 ADCEnSGCRx register contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When written, write the initial value.
5	SCANMD	Scan Mode 0: Multicycle scan mode 1: Continuous scan mode
4	ADIE	Scan End Interrupt Enable 0: INT_SGx is not output when the scan for SGx ends. 1: INT_SGx is output when the scan for SGx ends.
3 to 2	SCT[1:0]	Channel repeat times selection bit 00: The selected number of channel repeat times is one. 01: The selected number of channel repeat times is two. 10: The selected number of channel repeat times is four. 11: Setting prohibited
1	Reserved	When written, write the initial value.
0	TRGMD	Trigger Mode 0: Trigger input to SGx_TRG is disabled (Hardware trigger disabled). 1: SGx_TRG start trigger is selected for the trigger input to SGx. **NOTE** The software trigger is valid regardless of the TRGMD bit setting.

CAUTION

To prevent malfunction, ADCEnSGCRx should be set (except clearing TRGMD upon completion of AD conversion) when SGACT[5:0] of all scan groups are 0 (before the scan group is started) and TRGMD of all scan group is 0.

45.3.5.3 ADCEnPWDSGCR — PWM-Diag Scan Group Control Register

This register is a 32/16/8-bit read/write register, which controls PWM-Diag.

Access: ADCEnPWDSGCR can be read or written in 32-bit units.
ADCEnPWDSGCRL can be read or written in 16-bit units.
ADCEnPWDSGCRLL can be read or written in 8-bit units.

Address: ADCEnPWDSGCR: <ADCEn_base> + 508_H
ADCEnPWDSGCRL: <ADCEn_base> + 508_H
ADCEnPWDSGCRLL: <ADCEn_base> + 508_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWDTRGMD
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 45.33 ADCEnPWDSGCR register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	PWDTRGMD	PWM-Diag Trigger Mode Select 0: PVCR_TRG trigger input is disabled. 1: PVCR_TRG is selected for the trigger input to the PWM-Diag scan group.

CAUTION

To prevent malfunction, perform ADCEnPWDSGCR settings when SGACT4 of the PWM-Diag scan group (SG4) is 0 (before the scan group is started), and PWDTRGMD of the PWM-Diag scan group is 0.

45.3.5.4 ADCEnTSNSGCR — Temperature Measurement Scan Group Control Register

This register is a 32/16/8-bit read/write register, which controls the Temperatur Sensor.

Access: ADCEnPWDSGCR can be read or written in 32-bit units.
ADCEnPWDSGCRL can be read or written in 16-bit units.
ADCEnPWDSGCRLL can be read or written in 8-bit units.

Address: ADCEnPWDSGCR: <ADCEn_base> + 408_H
ADCEnPWDSGCRL: <ADCEn_base> + 408_H
ADCEnPWDSGCRLL: <ADCEn_base> + 408_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSNTRGMD
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 45.34 ADCEnTSNSGCR register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When written, write the initial value.
0	TSNTRGMD	Temperature Sensor Trigger Mode Select 0: TVCR_TRG trigger input is disabled. 1: TVCR_TRG is selected for the trigger input to the temperature measurement scan group.

CAUTION

To prevent malfunction, perform ADCEnTSNSGCR settings when SGACT0 of the temperature measurement scan group (SG0) is 0 (before the scan group is started), and TSNTRGMD of the Temperature Sensor scan group is 0.

45.3.5.5 ADCEnSGVCSPx — Scan Group x Start Virtual Channel Pointer

This register specifies the start pointer of a virtual channel.

Access: ADCEnSGVCSPx can be read or written in 32-bit units.
ADCEnSGVCSPxL can be read or written in 16-bit units.
ADCEnSGVCSPxLL can be read or written in 8-bit units.

Address: ADCEnSGVCSPx: <ADCEn_base> + x × 40_H + $40C_H$
ADCEnSGVCSPxL: <ADCEn_base> + x × 40_H + $40C_H$
ADCEnSGVCSPxLL: <ADCEn_base> + x × 40_H + $40C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	VCSP[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.35 ADCEnSGVCSPx register contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When written, write the initial value.
5 to 0	VCSP[5:0]	Start Virtual Channel Pointer These bits are used to select the virtual channel from which the scan is to be started.

CAUTION

- ADCEnSGVCSPx must be equal to or smaller than ADCEnSGVCEPx.
- When writing to the channel pointers, be sure to write in the following order: ADCEnSGVCSPx → ADCEnSGVCEPx. When SGx is started, the A/D conversion for the virtual channels within the range specified in ADCEnSGVCSPx and ADCEnSGVCEPx is executed.
- Though ADCEnSGVCSPx can be rewritten during the A/D conversion, the register is updated at the time ADCEnSGVCEPx is written. The new setting is applied when SGn is started next time.
- When the hardware trigger is used, writing to this register during operation is prohibited.

45.3.5.6 ADCEnSGVCEPx — Scan Group x End Virtual Channel Pointer

This register specifies the start pointer of a virtual channel.

Access: ADCEnSGVCEPx can be read or written in 32-bit units.
ADCEnSGVCEPxL can be read or written in 16-bit units.
ADCEnSGVCEPxLL can be read or written in 8-bit units.

Address: ADCEnSGVCEPx: <ADCEn_base> + x × 40_H + 410_H
ADCEnSGVCEPxL: <ADCEn_base> + x × 40_H + 410_H
ADCEnSGVCEPxLL: <ADCEn_base> + x × 40_H + 410_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	VCEP[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.36 ADCEnSGVCEPx register contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When written, write the initial value.
5 to 0	VCEP[5:0]	End Virtual Channel Pointer These bits are used to select the virtual channel at which the scan is to be ended.

CAUTION

- ADCEnSGVCSPx must be equal to or smaller than ADCEnSGVCEPx.
- When SGx is started, processing for the virtual channels within the range specified in ADCEnSGVCSPx and ADCEnSGVCEPx is executed.
  ADCEnSGVCEPx can be rewritten even when SGx is being processed. The new setting is applied when SGx is started next time.

45.3.5.7 ADCEnSGMCYCRx — Scan Group x Multicycle Register

This register is a 32/16/8-bit read/write register, which indicates the number of times for scanning in multicycle scan mode.

Access: ADCEnSGMCYCRx can be read or written in 32-bit units.
ADCEnSGMCYCRxL can be read or written in 16-bit units.
ADCEnSGMCYCRxLL can be read or written in 8-bit units.

Address: ADCEnSGMCYCRx: <ADCEn_base> + x × 40_H + 414_H
ADCEnSGMCYCRxL: <ADCEn_base> + x × 40_H + 414_H
ADCEnSGMCYCRxLL: <ADCEn_base> + x × 40_H + 414_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MCYC[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 45.37 ADCEnSGMCYCRx register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When written, write the initial value.
1, 0	MCYC[1:0]	Multicycle Specification These bits are used to specify the number of times for scanning in multicycle scan mode. 00_B: Number of scans = 1 01_B: Number of scans = 2 10_B: Setting prohibited 11_B: Number of scans = 4

CAUTION

- To prevent malfunction, perform the ADCEnSGMCYCRx settings when SGACT[5:0] of scan group x is 0 (before the scan group is started), and TRGMD is 0.
- When SGx is started, the scan for the virtual channels within the range specified in ADCEnSGVCSPx and ADCEnSGVCEPx is repeatedly executed as many times as specified in ADCEnSGMCYCRx.

45.3.5.8 ADCEnSGSEFCRx — Scan Group x Scan End Flag Clear Register

This register is a write-only register, which controls the scan end flag (SEFx). The read value is always 0.

Access: ADCEnSGSEFCRx can be written in 32-bit units.
ADCEnSGSEFCRxL can be written in 16-bit units.
ADCEnSGSEFCRxLL can be written in 8-bit units.

Address: ADCEnSGSEFCRx: <ADCEn_base> + x × 40_H + 418_H
ADCEnSGSEFCRxL: <ADCEn_base> + x × 40_H + 418_H
ADCEnSGSEFCRxLL: <ADCEn_base> + x × 40_H + 418_H

Initial value: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value																
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEFC
Initial value																
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 45.38 ADCEnSGSEFCRx register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, an undefined value is returned. The write value should be 0.
0	SEFC	Scan End Flag Clear Trigger Writing of 0: No effect. Writing of 1: The flag is cleared.

45.3.5.9 ADCEnPWDSGSEFCR — PWM-Diag Scan End Flag Clear Register

This register is a 32/16/8-bit write-only register, which controls the clearing of PWM-Diag scan end flag (SEF4). The bits are always read as 0.

Access: ADCEnPWDSGSEFCR can be written in 32-bit units.
ADCEnPWDSGSEFCRL can be written in 16-bit units.
ADCEnPWDSGSEFCRLL can be written in 8-bit units.

Address: ADCEnPWDSGSEFCR: <ADCEn_base> + 518_H
ADCEnPWDSGSEFCRL: <ADCEn_base> + 518_H
ADCEnPWDSGSEFCRLL: <ADCEn_base> + 518_H

Initial value: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWDSEFC
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 45.39 ADCEnPWDSGSEFCR register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, an initial value is returned. The write value should be 0.
0	PWDSEFC	PWM-Diag Scan End Flag Clear Trigger 0: No effect. 1: The scan end flag is cleared.

45.3.5.10 ADCEnTSNSGSEFCR — Temperature Sensor Scan End Flag Clear Register

This register is a write-only register, which controls the scan end flag (SEF0). The read value is always 0.

Access: ADCEnTSNSGSEFCR can be written in 32-bit units.
ADCEnTSNSGSEFCRL can be written in 16-bit units.
ADCEnTSNSGSEFCRLL can be written in 8-bit units.

Address: ADCEnTSNSGSEFCR: <ADCEn_base> + 418_H
ADCEnTSNSGSEFCRL: <ADCEn_base> + 418_H
ADCEnTSNSGSEFCRLL: <ADCEn_base> + 418_H

Initial value: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value																
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSNSEFC
Initial value																
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 45.40 ADCEnTSNSGSEFCR register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, an undefined value is returned. The write value should be 0.
0	TSNSEFC	Temperature Sensor Scan End Flag Clear Trigger Writing of 0: No effect. Writing of 1: The flag is cleared.

45.3.5.11 ADCEnSGSTR — Scan Group Status Register

This register is a 32/16-bit read-only register, which indicates the statuses of scan group SGx, and PWM-Diag scan group SG4 and the Temperature Sensor scan group SG0. It is initialized to $0000\ 0000_H$ at reset. The SGACT[5:0] bits are cleared when HALT is executed.

Access: This register can be read only in 32/16-bit units
ADCEnSGSTR can be read only in 32-bit units
ADCEnSGSTRL can be read only in 16-bit units

Address: ADCEnSGSTR:<ADCEm_base> + 308_H
ADCEnSGSTRL:<ADCEm_base> + 308_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	SGACT[5:0]						—	—	—	SEF[4:0]				
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 45.41 ADCEnSGSTR register contents (1/2)

Bit Position	Bit Name	Function
31 to 14	Reserved	When read, an initial value is returned. Writing is ignored.
13	SGACT[5]	SVSTOP Status Flag 0: SVSTOP is canceled. 1: SVSTOP is accepted.
12	SGACT[4]	PWM-Diag Scan Group (SG4) Status Flag 0: A/D conversion for PWM-Diag (SG4) is completed. 1: A/D conversion for PWM-Diag (SG4) is in processing or suspension.
11	SGACT[3]	Scan Group 3 (SG3) Status Flag 0: A/D conversion for SG3 is completed. 1: A/D conversion for SG3 is in processing or suspension.
10	SGACT[2]	Scan Group 2 (SG2) Status Flag 0: A/D conversion for SG2 is completed. 1: A/D conversion for SG2 is in processing or suspension.
9	SGACT[1]	Scan Group 1 (SG1) Status Flag 0: A/D conversion for SG1 is completed. 1: A/D conversion for SG1 is in processing or suspension.
8	SGACT[0]	Temperature Sensor Scan Group 0 (SG0) Status Flag 0: A/D conversion for SG0 is completed. 1: A/D conversion for SG0 is in processing or suspension.
8 to 5	Reserved	When written, write the initial value.
4	SEF[4]	PWM-Diag Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: – ADCEnPWDTSNDR for PMW-Diag/Temperature Sensor is read. – ADCEnPWDDIR for PWM-Diag is read. – ADCEnPWDSGSEFCR.PWDSEFC is written as 1. 1: The A/D conversion result is written to ADCEnPWDTSNDR.PWDDR[15:0] for PWM-Diag.

Table 45.41 ADCEnSGSTR register contents (2/2)

Bit Position	Bit Name	Function
3	SEF[3]	SG3 Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: – ADCEnDRj for the virtual channel which ADCEnSGVCEP3 indicates is read. – ADCEnDIRj for the virtual channel which ADCEnSGVCEP3 indicates is read. – ADCEnSGSEFCRx.SEFC is written as 1. 1: The A/D conversion result is written to ADCEnDRj for the virtual channel which ADCEnSGVCEP3 indicates.
2	SEF[2]	SG2 Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: – ADCEnDRj for the virtual channel which ADCEnSGVCEP2 indicates is read. – ADCEnDIRj for the virtual channel which ADCEnSGVCEP2 indicates is read. – ADCEnSGSEFCRx.SEFC is written as 1. 1: The A/D conversion result is written to ADCEnDRj for the virtual channel which ADCEnSGVCEP2 indicates.
1	SEF[1]	SG1 Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: – ADCEnDRj for the virtual channel which ADCEnSGVCEP1 indicates is read. – ADCEnDIRj for the virtual channel which ADCEnSGVCEP1 indicates is read. – ADCEnSGSEFCRx.SEFC is written as 1. 1: The A/D conversion result is written to ADCEnDRj for the virtual channel which ADCEnSGVCEP1 indicates.
0	SEF[0]	Temperature Sensor Scan End Flag Indicates the status of the scan result data. 0: The flag is cleared when any of the following operations is performed: – ADCEnPWDTSNDR for PMW-Diag/Temperature Sensor is read. – ADCEnTSNDIR for temperature sensor is read. – ADCEnTSNSGSEFCR.TSNSEFC is written as 1. 1: The A/D conversion result is written to ADCEnPWDTSNDR.TSNDR[15:0] for temperature measurement.

45.3.6 Hardware Trigger Specific Register

45.3.6.1 ADCEnSGTSELx — A/D Conversion Trigger Select Control Register

This register is a 32/16-bit read/write register, which controls the A/D conversion trigger (H/W trigger) selection for SGx_TRG.

Access: ADCEnSGTSELx can be read or written in 32-bit units.
ADCEnSGTSELxL can be read or written in 16-bit units.

Address: ADCEnSGTSELx: <ADCEn_base> + x × 40_H + $41C_H$
ADCEnSGTSELxL: <ADCEn_base> + x × 40_H + $41C_H$

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TxSEL[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.42 ADCEnSGTSELx Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, an undefined value is returned. The write value should be 0.
15 to 0	TxSEL[15:0]	A/D Conversion Trigger (Hardware Trigger) Select 0: Hardware trigger is disabled. 1: Hardware trigger is enabled. **CAUTION** When setting TxSELp to 1, set only one of the bits to 1 .

45.3.6.2 ADCEnINTPEDGCTL — External interrupts edge selection Register

This register is used to select the edge of the external interrupts INTP0 to INTP2, which shall trigger an A/D conversion.

Access: This register can be read or written in 32-bit units.

Address: FFC0 603C$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	INTP2ESEL[1:0]		INTP1ESEL[1:0]		INTP0ESEL[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.43 ADCEnSGTSELx Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	When read, an undefined value is returned. The write value should be 0.
5, 4	INTP2ESEL[1:0]	INTP2 edge selection for A/D conversion hardware trigger 00_B: rising edge 01_B: falling edge $1x_B$: rising and falling edge
3, 2	INTP1ESEL[1:0]	INTP1 edge selection for A/D conversion hardware trigger 00_B: rising edge 01_B: falling edge $1x_B$: rising and falling edge
1, 0	INTP0ESEL[1:0]	INTP0 edge selection for A/D conversion hardware trigger 00_B: rising edge 01_B: falling edge $1x_B$: rising and falling edge

The list below shows the hardware triggers to be selected.

Table 45.44 List of A/D Conversion Hardware Triggers (1/2)

Unit	Control Register/Bit		Trigger Input Signal	
	Register Name	Bit Name	Symbol	Connection Destination Unit
ADCE0	ADCE0 SGTSEL1	T1SEL0	ADCE0TRGI1	External trigger pin
		T1SEL1	INTP0	External interrupt pin
		T1SEL2	INTP1	
		T1SEL3	INTP2	
		T1SEL4	INTTAUB0I2	TAUB0
		T1SEL5	INTTAUB0I4	
		T1SEL6	INTTAUB0I6	
		T1SEL7	INTTAUB0I8	
		T1SEL8	INTTAUB1I2	TAUB1
		T1SEL9	INTTAUB1I4	
		T1SEL10	INTTAUB1I6	
		T1SEL11	INTTAUB1I8	
		T1SEL12	INTTAUB2I0	TAUB2
		T1SEL13	INTTAUB2I1	
		T1SEL14	INTTAUB2I2	
		T1SEL15	INTTAUB2I3	
ADCE0	ADCE0 SGTSEL2	T2SEL0	ADCE0TRGI2	External trigger pin
		T2SEL1	INTP0	External interrupt pin
		T2SEL2	INTP1	
		T2SEL3	INTP2	
		T2SEL4	INTTAUB0I2	TAUB0
		T2SEL5	INTTAUB0I4	
		T2SEL6	INTTAUB0I6	
		T2SEL7	INTTAUB0I8	
		T2SEL8	INTTAUB0I10	
		T2SEL9	INTTAUB0I12	
		T2SEL10	INTTAUB0I14	
		T2SEL11	INTTAUB1I2	TAUB1
		T2SEL12	INTTAUB1I4	
		T2SEL13	INTTAUB1I6	
		T2SEL14	INTTAUB1I8	
		T2SEL15	INTTAUB1I10	

Table 45.44 List of A/D Conversion Hardware Triggers (2/2)

Unit	Control Register/Bit		Trigger Input Signal	
	Register Name	Bit Name	Symbol	Connection Destination Unit
ADCE0	ADCE0 SGTSEL3	T3SEL0	ADCE0TRGI3	External trigger pin
		T3SEL1	INTP0	External interrupt pin
		T3SEL2	INTP1	
		T3SEL3	INTP2	
		T3SEL4	INTTAUB0I2	TAUB0
		T3SEL5	INTTAUB0I4	
		T3SEL6	INTTAUB0I6	
		T3SEL7	INTTAUB0I8	
		T3SEL8	INTTAUB1I2	TAUB1
		T3SEL9	INTTAUB1I4	
		T3SEL10	INTTAUB1I6	
		T3SEL11	INTTAUB1I8	
		T3SEL12	INTTAUB2I0	TAUB2
		T3SEL13	INTTAUB2I1	
		T3SEL14	INTTAUB2I2	
		T3SEL15	INTTAUB2I3	

CAUTION

To prevent malfunction, perform ADCEnSGTSELx settings when SGACT[5:0] of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

45.3.6.3 ADCEnTSNSGTSEL — A/D Conversion Temperature Measurement Trigger Select Control Register

This register is a 32/16-bit read/write register, which controls the A/D conversion trigger (H/W trigger) selection for the temperature measurement scan group.

Access: ADCEnTSNSGTSEL can be read or written in 32-bit units.
ADCEnTSNSGTSELL can be read or written in 16-bit units.

Address: ADCEnTSNSGTSEL: <ADCEn_base> + $41C_H$
ADCEnTSNSGTSELL: <ADCEn_base> + $41C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TTSNSEL[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.45 ADCEnTSNSGTSEL Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, an undefined value is returned. The write value should be 0.
15 to 0	TTSNSEL[15:0]	A/D Conversion Trigger (Hardware Trigger) Select 0: Hardware trigger is disabled. 1: Hardware trigger is enabled.

The list below shows the hardware triggers to be selected.

Table 45.46 List of A/D Conversion Temperature Measurement Hardware Triggers

	Control Register/Bit		Trigger Input Signal	
Unit	**Register Name**	**Bit Name**	**Symbol**	**Connection Destination Unit**
ADCE0	ADCE0 TSNSGTSEL	TTSNSEL0	—	not connected
		TTSNSEL1	—	
		TTSNSEL2	—	
		TTSNSEL3	—	
		TTSNSEL4	INTTAUB0I2	TAUB0
		TTSNSEL5	INTTAUB0I4	
		TTSNSEL6	INTTAUB0I6	
		TTSNSEL7	INTTAUB0I8	
		TTSNSEL8	INTTAUB1I2	TAUB1
		TTSNSEL9	INTTAUB1I4	
		TTSNSEL10	INTTAUB1I6	
		TTSNSEL11	INTTAUB1I8	
		TTSNSEL12	INTTAUB2I0	TAUB2
		TTSNSEL13	INTTAUB2I1	
		TTSNSEL14	INTTAUB2I2	
		TTSNSEL15	INTTAUB2I3	

CAUTION

To prevent malfunction, perform ADCEnTSNSGTSEL settings when SGACT[5:0] of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

45.3.7 Self-Diagnostic Specific Registers

45.3.7.1 ADCEnDGCTL0 — Self-Diagnostic Voltage Level Control Register

This register controls the self-diagnostic voltage level.

Access: ADCEnDGCTL0 can be read or written in 32-bit units.
ADCEnDGCTL0L can be read or written in 16-bit units.
ADCEnDGCTL0LL can be read or written in 8-bit units.

Address: ADCEnDGCTL0: <ADCEn_base> + 350_H
ADCEnDGCTL0L: <ADCEn_base> + 350_H
ADCEnDGCTL0LL: <ADCEn_base> + 350_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	PSEL2	PSEL1	PSEL0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 45.47 ADCEnDGCTL0 register contents

Bit Position	Bit Name	Function
31 to 3	Reserved	When written, write the initial value.
2 to 0	PSEL[2:0]	Self-Diagnostic Voltage level selection bit

ADCEnDGCTL0			Output signal			
PSEL2	PSEL1	PSEL0	AD DIAGOUT	DIAGOUT2	DIAGOUT1	DIAGOUT0
0	0	0	Hi-z	Hi-z	Hi-z	Hi-z
0	0	1	AVss	$2/3AnV_{REF}$	$1/2AnV_{REF}$	$1/3AnV_{REF}$
0	1	0	$1/3AnV_{REF}$	$1/3AnV_{REF}$	$2/3AnV_{REF}$	$1/2AnV_{REF}$
0	1	1	$1/2AnV_{REF}$	$1/2AnV_{REF}$	$1/3AnV_{REF}$	$2/3AnV_{REF}$
1	0	0	$2/3AnV_{REF}$	Hi-z	Hi-z	Hi-z
1	0	1	AnV_{REF}	$1/3AnV_{REF}$	$1/3AnV_{REF}$	$1/3AnV_{REF}$
1	1	0	AnV_{REF}	$1/2AnV_{REF}$	$1/2AnV_{REF}$	$1/2AnV_{REF}$
1	1	1	AnV_{REF}	$2/3AnV_{REF}$	$2/3AnV_{REF}$	$2/3AnV_{REF}$

45.3.7.2 ADCEnDGCTL1 — Self-Diagnostic Control Register

This register controls the self-diagnostic channel.

Access: ADCEnDGCTL1 can be read or written in 32-bit units.
ADCEnDGCTL1L can be read or written in 16-bit units.

Address: ADCEnDGCTL1: <ADCEn_base> + 354_H
ADCEnDGCTL1L: <ADCEn_base> + 354_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CDG[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.48 ADCEnDGCTL1 register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When written, write the initial value.
15,12,9,6,3, 0	CDG [15,12,9,6,3,0]	Self-Diagnostic Channel Select 0: ANInm is selected. 1: DIAGOUT0 is selected.
13,10,7,4,1	CDG [13,10,7,4,1]	Self-Diagnostic Channel Select 0: ANInm is selected. 1: DIAGOUT1 is selected.
14,11,8,5,2	CDG [14,11,8,5,2]	Self-Diagnostic Channel Select 0: ANInm is selected. 1: DIAGOUT2 is selected.

CAUTION

To prevent malfunction, perform ADCEnDGCTL1 settings when SGACT[5:0] of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

45.3.7.3 ADCEnPDCTL2 — Pull Down Control Register 2

This register specifies a channel which the pull down resistor is connected with.

For details, see Section 45.5.3, Diagnostic of Open Pins.

Access: ADCEnPDCTL2 can be read or written in 32-bit units.
ADCEnPDCTL2L can be read or written in 16-bit units.
ADCEnPDCTL2LL can be read or written in 8-bit units.

Address: ADCEnPDCTL2: <ADCEn_base> + $35C_H$
ADCEnPDCTL2L: <ADCEn_base> + $35C_H$
ADCEnPDCTL2LL: <ADCEn_base> + $35C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	PDNB[19:16]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PDNB[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 45.49 ADCEnPDCTL2 Register Contents

Bit Position	Bit Name	Function
31 to 20	Reserved	When written, write the initial value.
19 to 0	PDNB[19:0]	Pull Down Enable Control These bits control the on-chip pull-down resistor of the corresponding physical channel (ANI0[19:0]). 0: Pull-down enable is off 1: Pull-down enable is on

CAUTION

To prevent malfunction, perform ADCEnPDCTL2 settings when SGACT[5:0] of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

NOTE

For on-chip pull-down resistor values, see the Electrical Characteristics section in the Data Sheet document.

45.3.8 Emulation Specific Register

45.3.8.1 ADCEnEMU — Emulation Control Register

This register controls the SVSTOP disable signal.

Access: This register can be read/written in 8-bit units.

Address: <ADCEn_base> + 388_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	SVSDIS	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R

Table 45.50 ADCEnEMU Register Contents

Bit Position	Bit Name	Function
7	SVSDIS	SVSTOP Disable 0: SVSTOP is enabled 1: SVSTOP is disabled
6 to 0	Reserved	When written, write the initial value.

CAUTION

To prevent malfunction, perform SVSDIS settings when SGACT[5:0] of all scan groups is 0 (before scan groups are started) and TRGMD of all scan groups is 0.

45.4 Operation

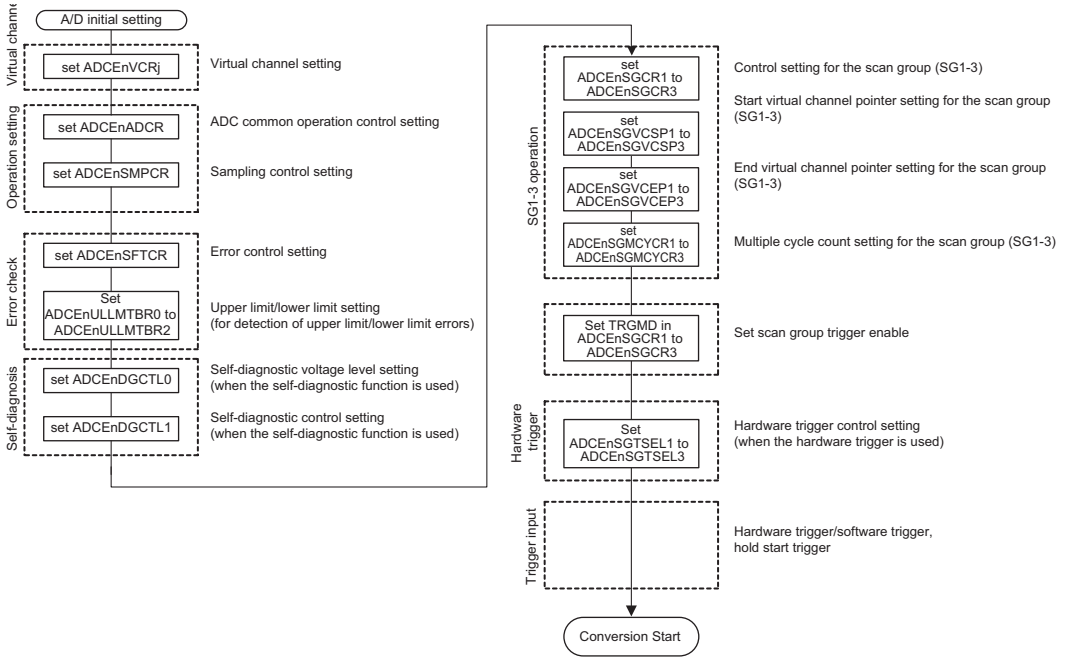
45.4.1 Priority Selection

The priority order of the scan groups can be selected from two sets via the scan groups priority level control register ADCEnSGPRCR:

- SGPR[18:0] = 43210_H: priority order
  SG4 (PWM-Diag, highest) > SG3 > SG2 > SG1 > SG0 (temperature measurement, lowest)
- SGPR[18:0] = 32401_H: priority order
  SG3 (highest) > SG2 > SG4 (PWM-Diag) > SG0 (temperature measurement) > SG1 (lowest)

45.4.2 Initial Setting

Table 45.47 shows an initial setting example of the A/D conversion. For trigger input, see **Figure 45.5** in the next section. For interrupt request signals, see Section 45.4.12, Scan End Interrupt Request.



Note: Set these registers as necessary.

Figure 45.4 Flowchart for Initial Setting

45.4.3 Trigger input

The following figure shows the flowchart for trigger input.

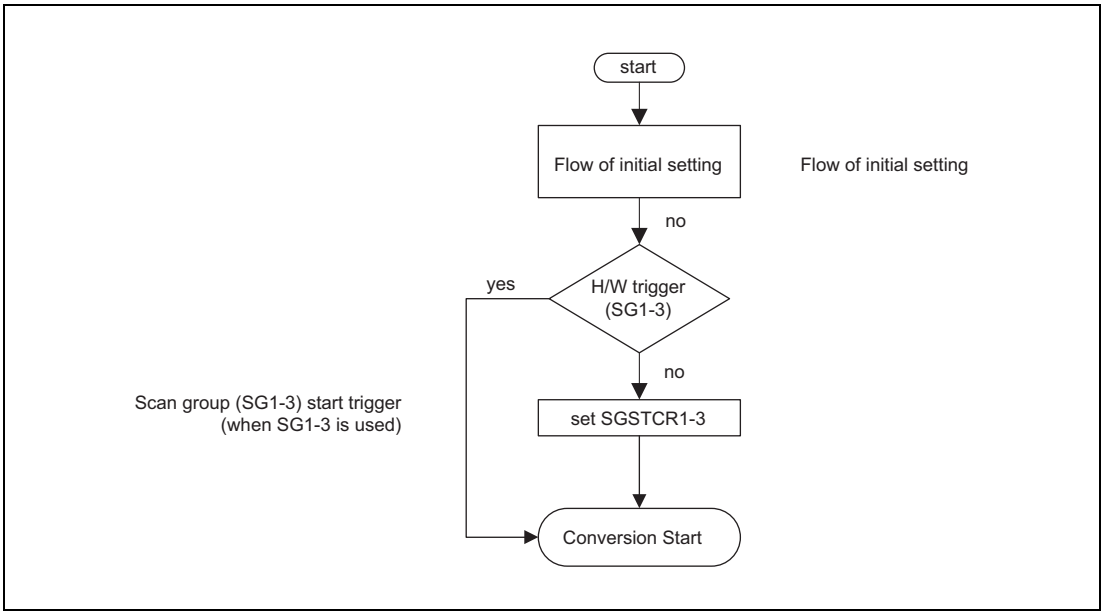


Figure 45.5 Flowchart for Trigger Input

NOTE

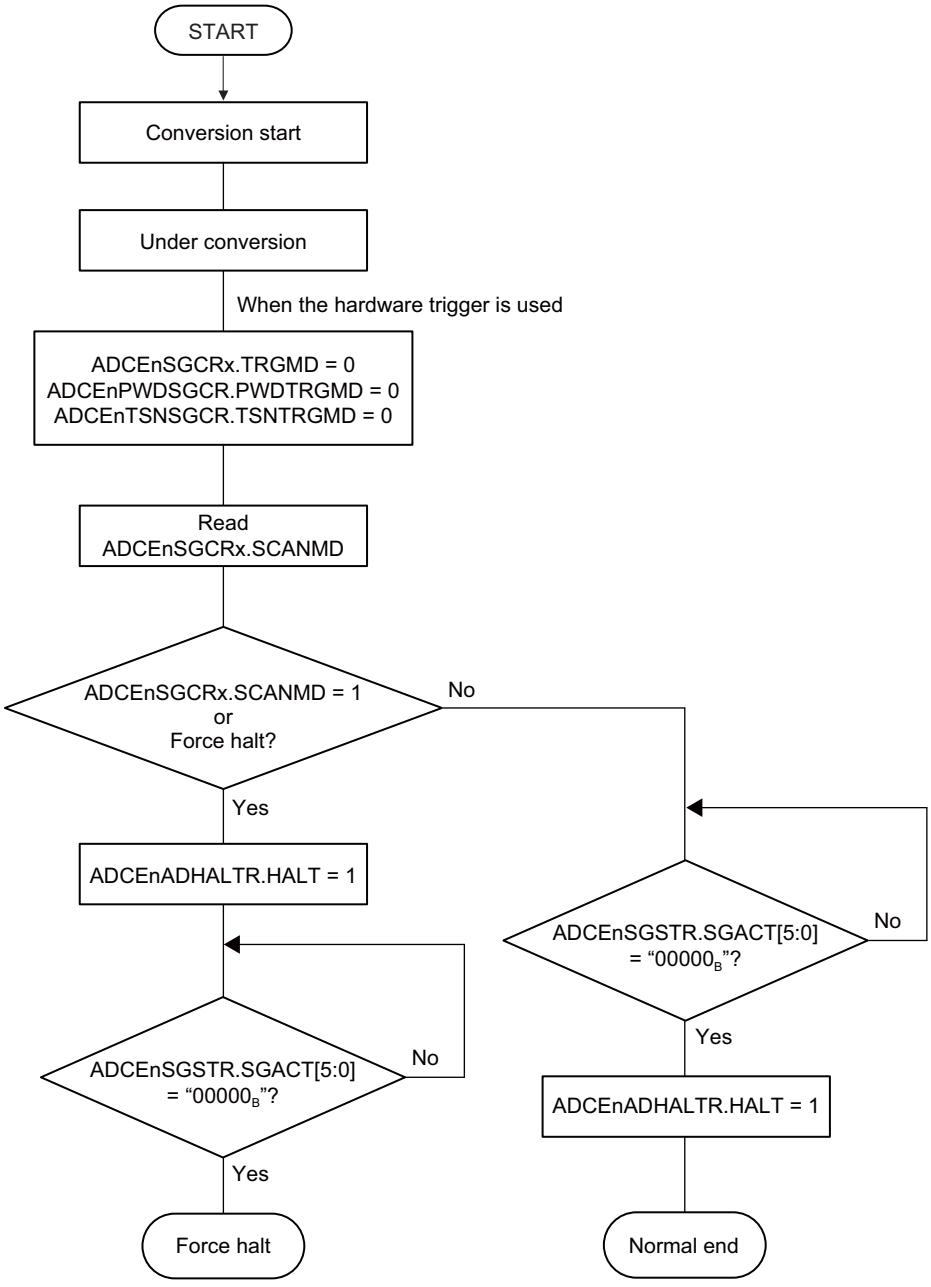
When an SG start trigger is generated during scanning, the SG start trigger is ignored.

For INTP0, INTP1 and INTP2 the scan groups SG1 to SG3 can be configured to start conversion on rising, falling or both edges of the input signal. The valid edge can be selected by the ADCEnINTPEDGCTL register.

All other HW triggers start conversion on the rising edge only.

45.4.4 Ending A/D conversion

The flow at the end of A/D conversion is shown below.



Note: The final setting of HALT when A/D conversion ends normally is done to clarify whether the status is "Conversion" or "IDLE". Whether HALT is set or not when A/D conversion ends normally does not effect the A/D conversion accuracy.

Figure 45.6 Flow at the End of A/D Conversion

45.4.5 Example of Scan Group Operation

(1) Multi cycle scan mode

The following figure illustrates an operation example where conversion is performed for four virtual channels, using the two cycle scan in multicycle scan mode as scan group 0.

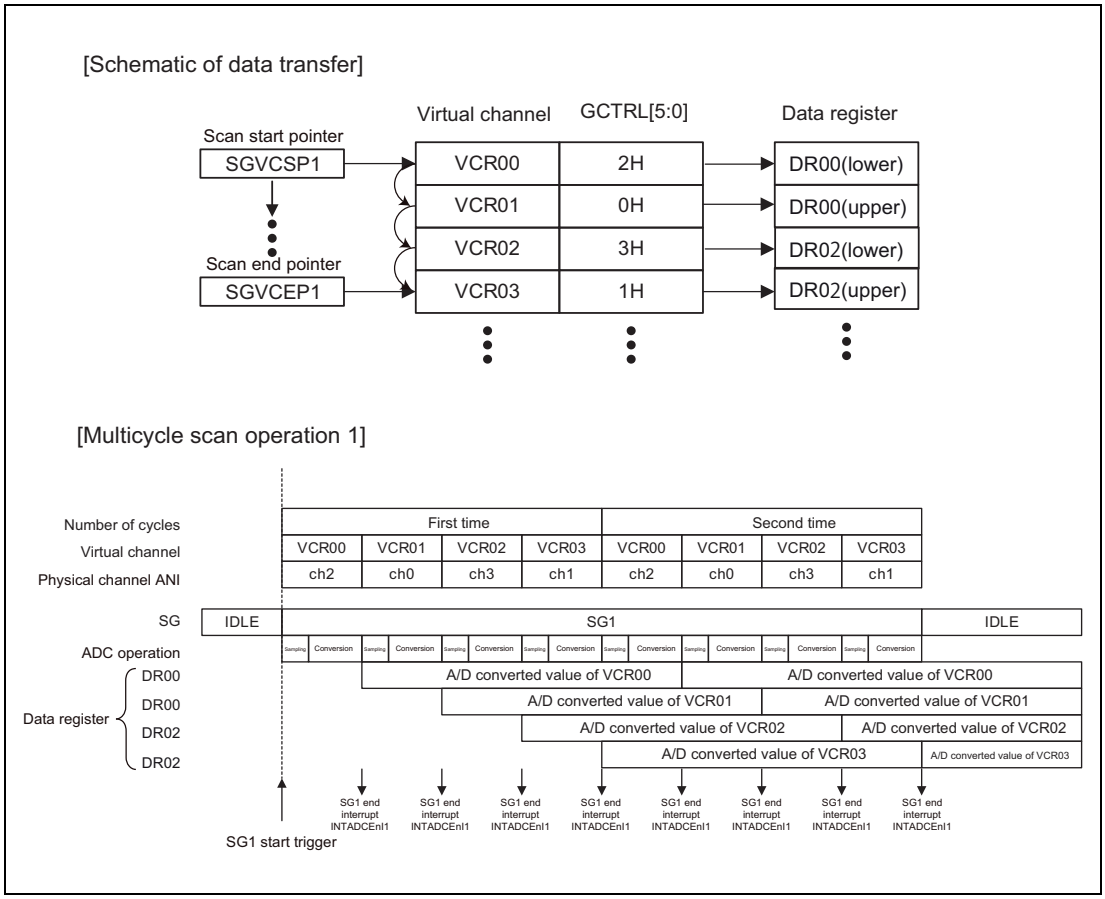


Figure 45.7 Example of Multicycle Scan Operation 1

The following figure illustrates an operation example where a pin is scanned once, using multicycle scan mode.

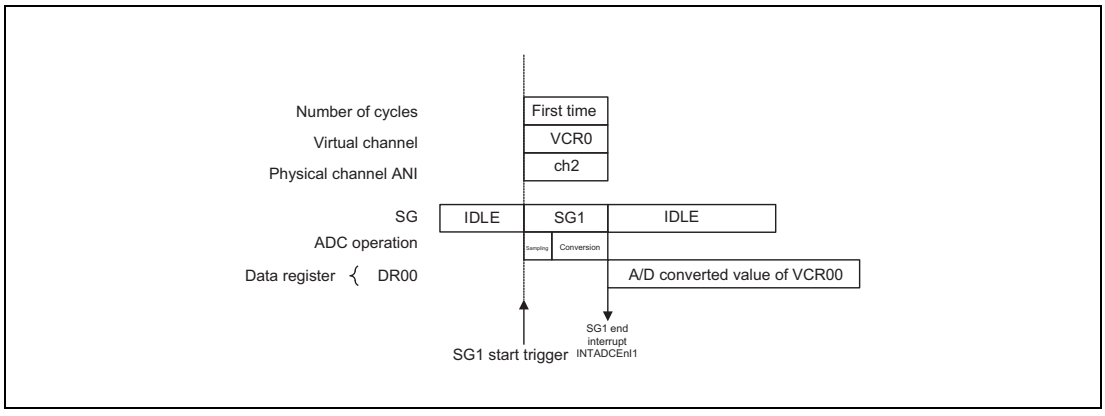


Figure 45.8 Example of Multicycle Scan Operation 2

(2) Continuous scan mode

Continuous scan mode allows A/D conversion of the SG channel indicated by the pointer specified by ADCEnSGVCSPx.VCSP[5:0] to ADCEnSGVCEPx.VCEP[5:0] (x = 1 to 3) to continue until ADCEnADHALTR: HALT is asserted. This mode operates exclusively in each SG. The following figure shows an example of operation in continuous scan mode.

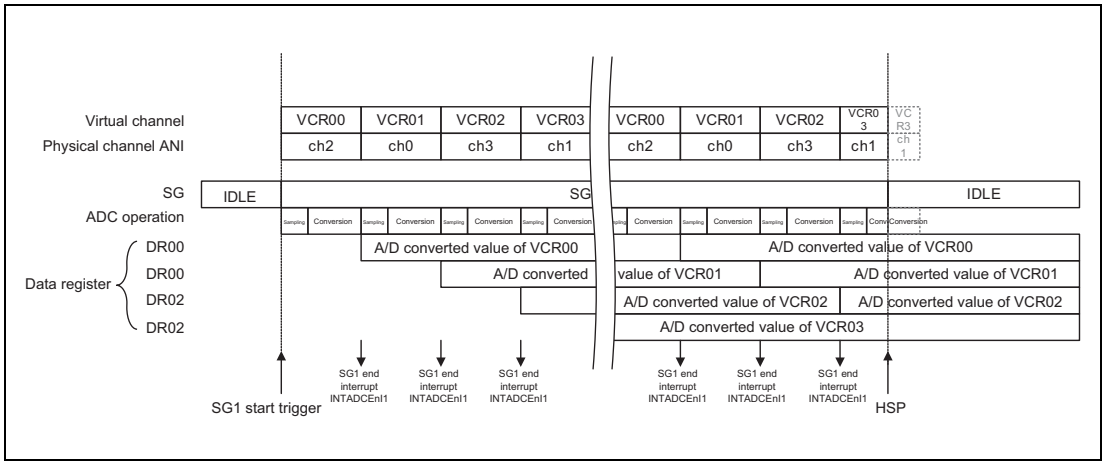


Figure 45.9 Example of Continuous Scan Operation

45.4.6 Channel repeat mode

Channel repeat mode allows A/D conversion of the SG channel indicated by the pointer specified by ADCEnSGVCSPm.VCSP[5:0] to ADCEnSGVCEPm.VCEP[5:0] to repeat number of channel repeat times specified by SGCRn (n = 1 to 3):SCTn[1:0]. This mode operates exclusively in each SG. The number of channel repeat times is selectable from 1, 2, and 4.

The following figures show examples of operation under respective conditions.

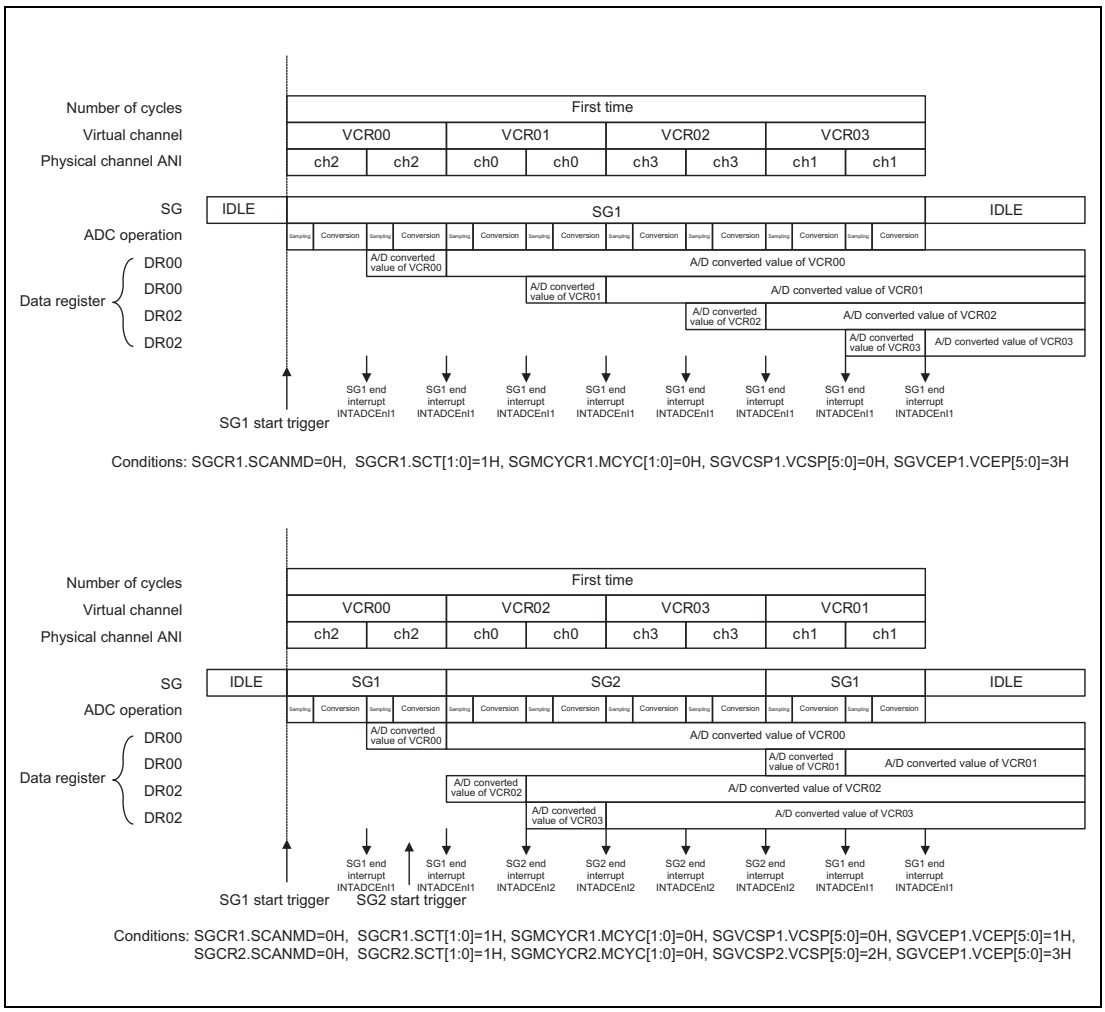
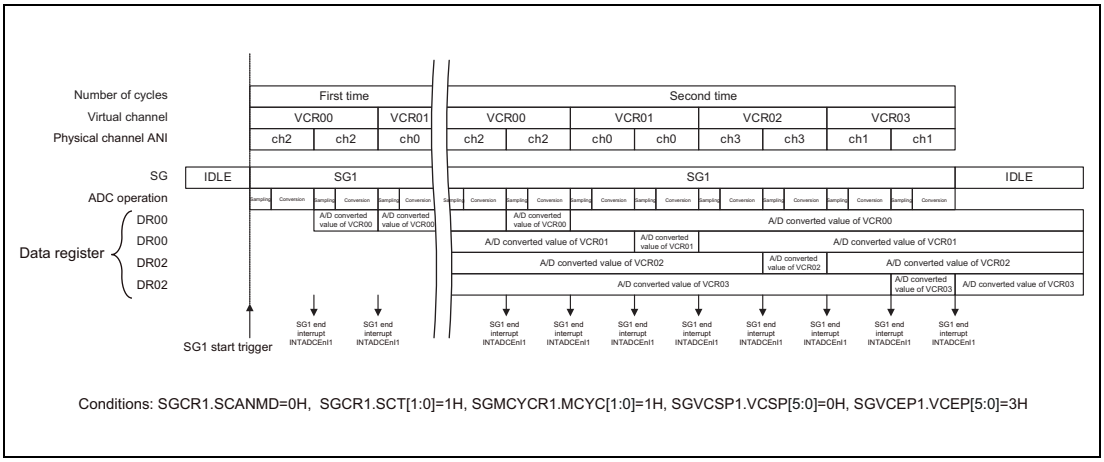
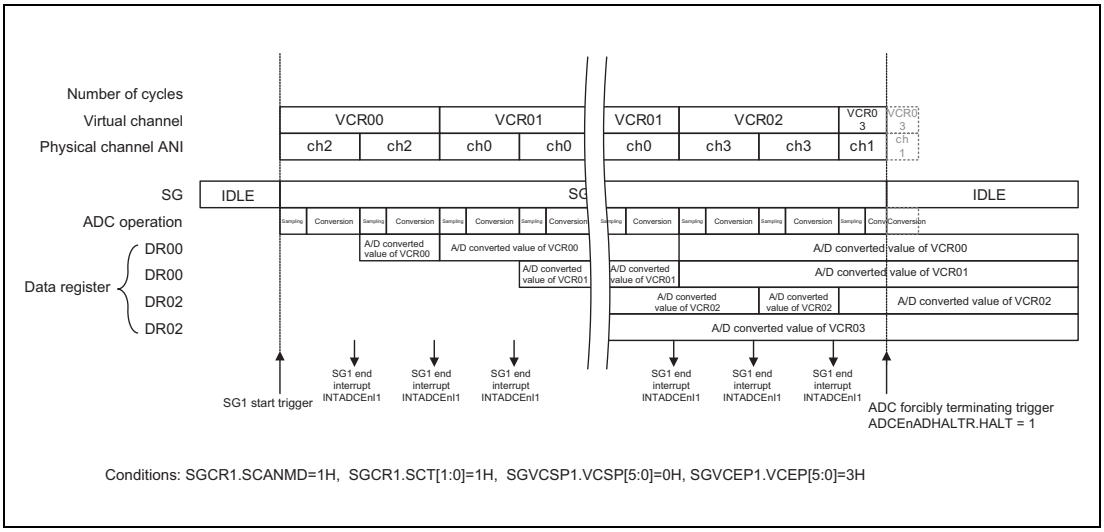


Figure 45.10 Example of Channel Repeat Operation 1



Conditions: SGCR1.SCANMD=0H, SGCR1.SCT[1:0]=1H, SGMCYCR1.MCYC[1:0]=1H, SGVCSP1.VCSP[5:0]=0H, SGVCEP1.VCEP[5:0]=3H

Figure 45.11 Example of Channel Repeat Operation 2



Conditions: SGCR1.SCANMD=1H, SGCR1.SCT[1:0]=1H, SGVCSP1.VCSP[5:0]=0H, SGVCEP1.VCEP[5:0]=3H

Figure 45.12 Example of Channel Repeat Operation 3

45.4.7 A/D Conversion with PWM-Diag Enabled

With the PWM-Diag function enabled, A/D conversion is performed by the signal from the PWM-Diag.

For details on the PWM-Diag function, refer to Section 32, PWM Generators and Diagnostic (PWM-Diag).

To control the A/D conversion, the A/D converter receives the setting information by the A/D conversion trigger select (PWSA) signal. The flow of A/D conversion with PWM-Diag is shown in the following figure.

CAUTION

If the trigger signal PVCR_TRG of PWM-Diag function has a higher-priority than SGx_TRG (x = 1 to 3), the operation of other scan groups may be kept waiting until PWM-Diag function is ended.

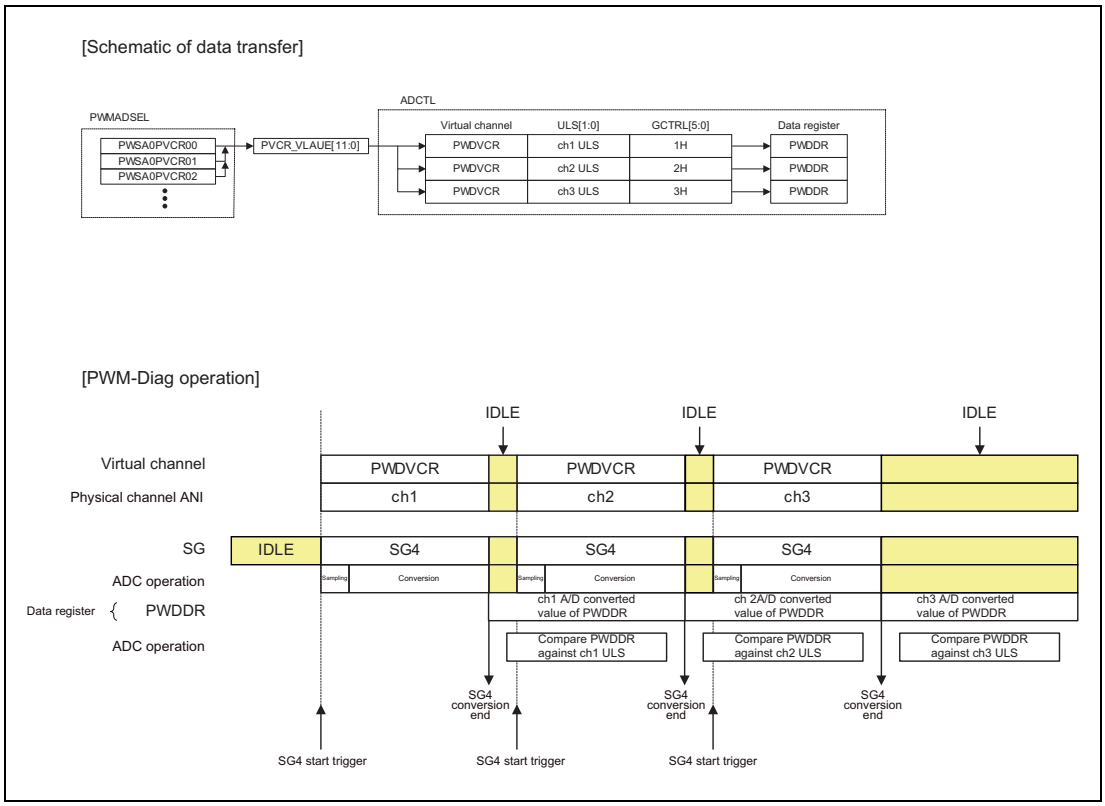


Figure 45.13 PWM-Diag Operation

45.4.8 Example of Synchronous Suspend and Resume Operation

Figure 45.14 shows an example of synchronous suspend and resume operation. In this example, a higher-priority SG interrupts a lower-priority SG.

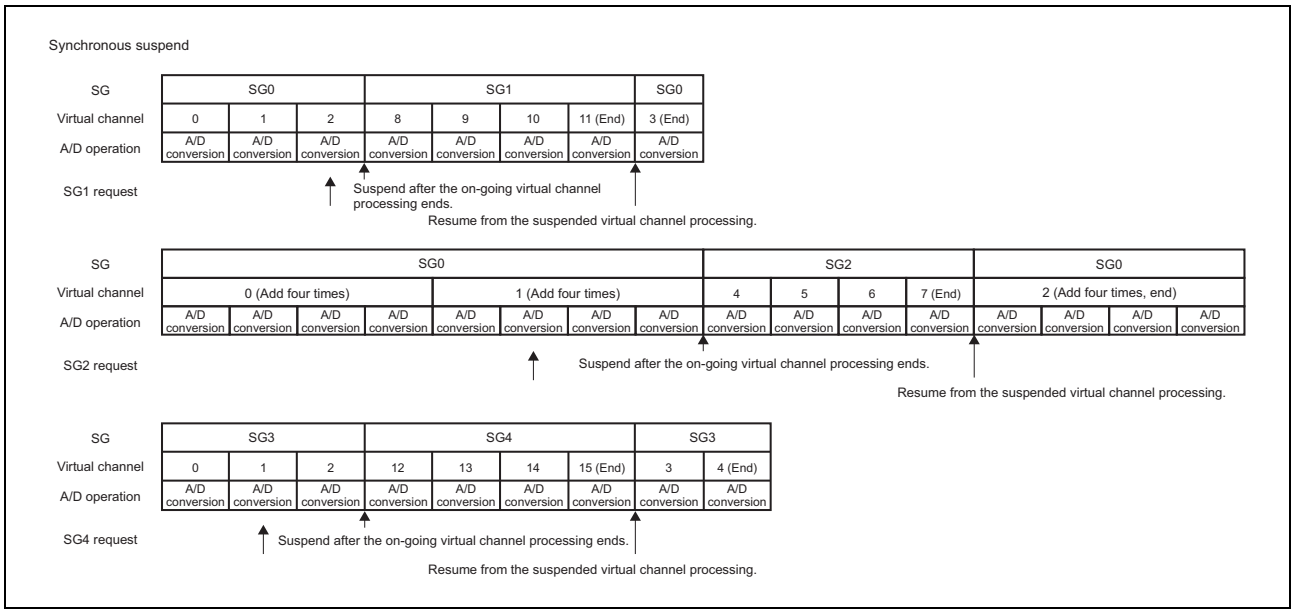


Figure 45.14 Example of Synchronous Suspend and Resume Operation

NOTE

The priority selection in the example above is ADCEnSGPRCR.SGPR[1:0] = 1, i.e.

Lower Higher

SG0 < SG1 < SG2 < SG3 < SG4

45.4.9 Example of Asynchronous Suspend and Resume Operation

Figure 45.15 shows an example of asynchronous suspend and resume operation. In this example, a higher-priority SG interrupts a lower-priority SG.

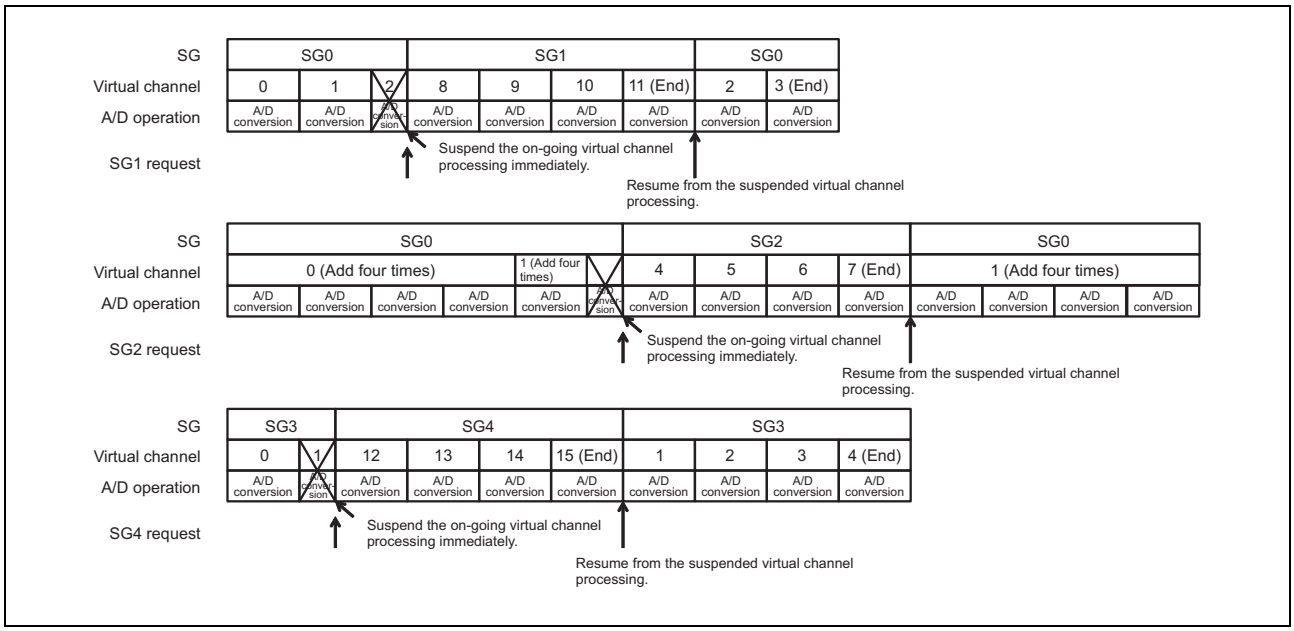


Figure 45.15 Example of Asynchronous Suspend and Resume Operation

NOTE

The priority selection in the example above is ADCEnSGPRCR.SGPR[1:0] = 1, i.e.

Lower Higher

SG0 < SG1 < SG2 < SG3 < SG4

45.4.10 Error Detecting Functions

Error detecting functions cover upper-limit error, lower-limit error, and overwrite error.

45.4.10.1 Upper-Limit/Lower-Limit Error Detecting Function

The upper-limit/lower-limit error detecting function determines whether the A/D converted data is larger than the upper-limit table ULMTB[11:0] or smaller than the lower-limit table LLMTB[11:0] at the end of A/D conversion.

45.4.10.2 Overwrite Error Detecting Function

When the ADCEnDRj (or ADCEnDIRj) of a virtual channel where A/D converted data is written and WFLG in ADCEnDIRj is 1 is not read but the next A/D converted data is written in the same ADCEnDRj, an overwrite error is detected.

45.4.10.3 SVSTOP Operation

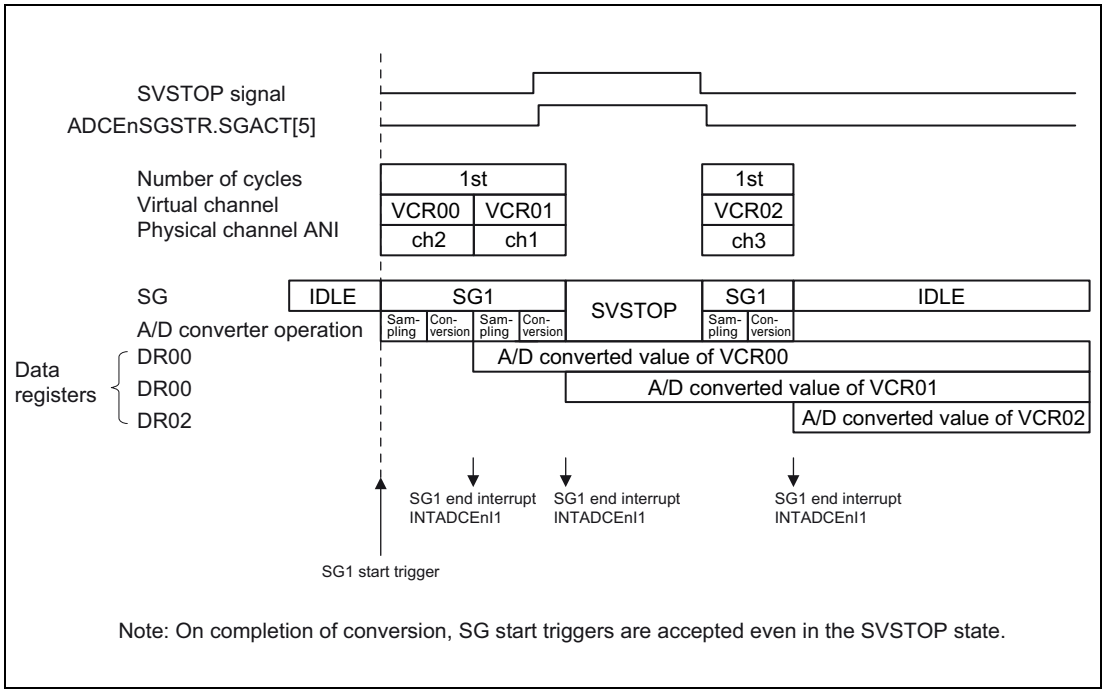
The SVSTOP function is supported by the SVSTOP signal sent from the on-chip debugger control unit.

The SVSTOP function stops conversion of the A/D converter when the SVSTOP signal is input during an emulation break. While the SVSTOP signal is high, reading registers ADCEnDRj, ADCEnDIRj, ADCEnSGSTR, ADCEnULER, and ADCEnOWER by the external access does not affect these registers.

ADCEnSGSTR.SGACT[5] is set to 1 when SVSTOP = High is input while ADCEnEMU.SVSDIS = 0 to make a transition to the SVSTOP state. Hardware triggers and software triggers are valid in the SVSTOP state. When SVSTOP = High is input while EMUCR:SVSDIS = 1’b1, the SAR-ADCE(SM) does not transition to the SVSTOP state. ADHALT (forced termination of A/D conversion) should not be performed in the SVSTOP state.

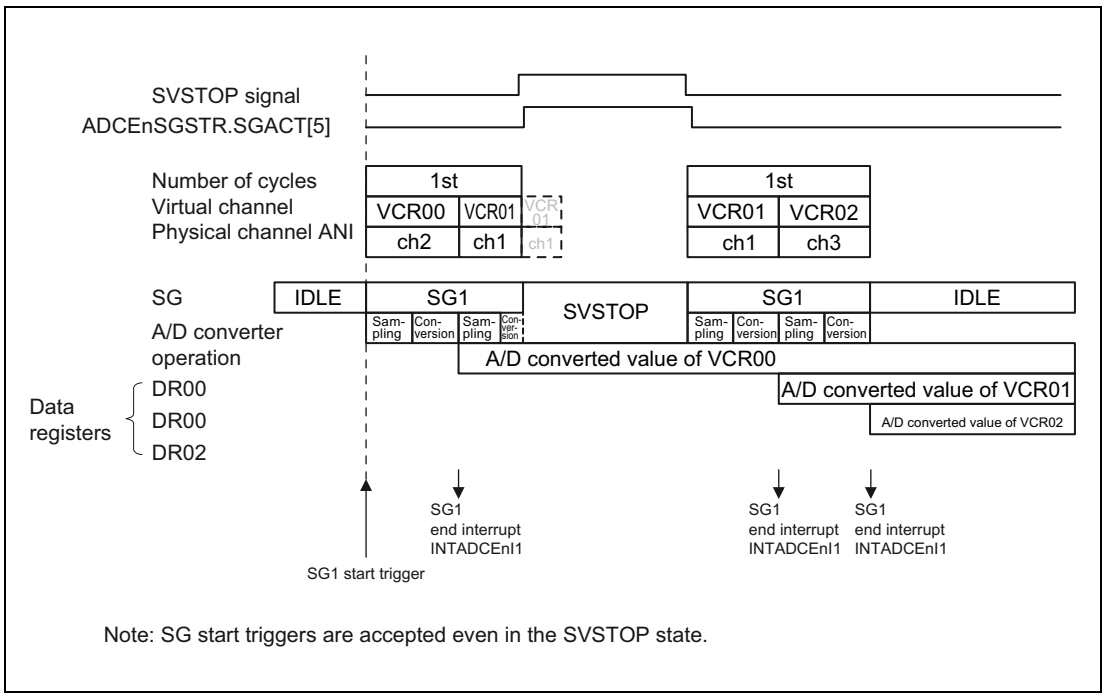
In operations for synchronous suspension, a new start trigger cannot be accepted over the time from input of the high level to SVSTOP to completion of conversion on the channel where conversion is currently proceeding. This time can be up to the time taken for one A/D conversion.

The following example illustrates a SVSTOP operation example.



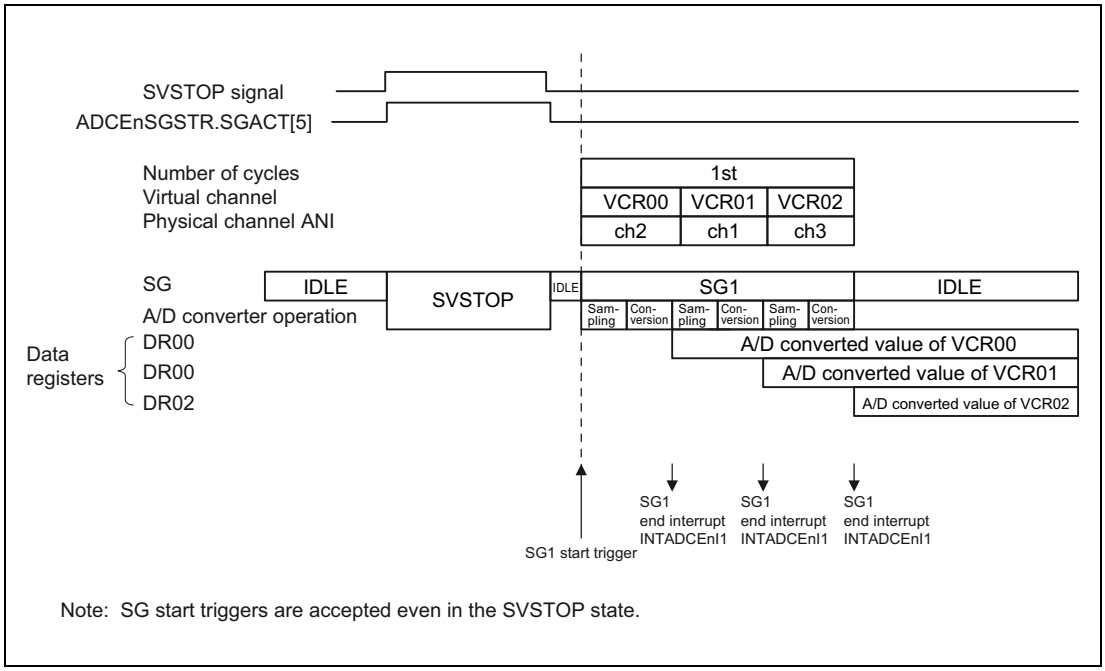
Note: On completion of conversion, SG start triggers are accepted even in the SVSTOP state.

Figure 45.16 Example of SVSTOP Operation (ADCEnADCR.SUSMTD[1:0] = 00 and ADCEnEMU.SVSDIS = 0)



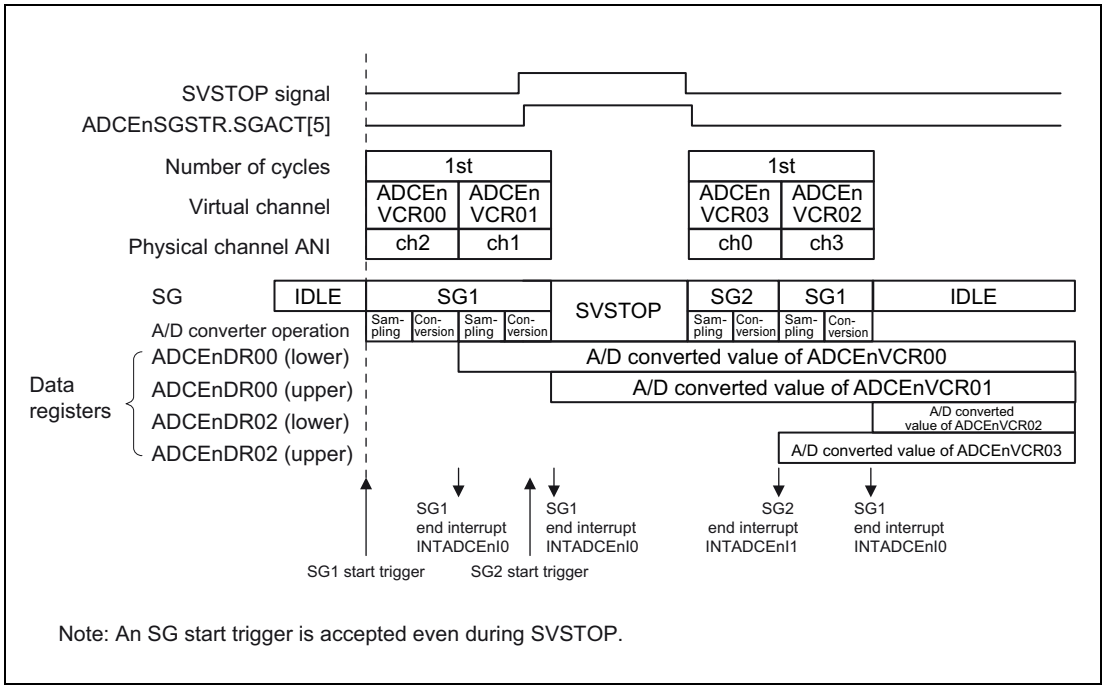
Note: SG start triggers are accepted even in the SVSTOP state.

Figure 45.17 Example of SVSTOP Operation (ADCEnADCR.SUSMTD[1:0] = 10 and ADCEnEMU.SVSDIS = 0)



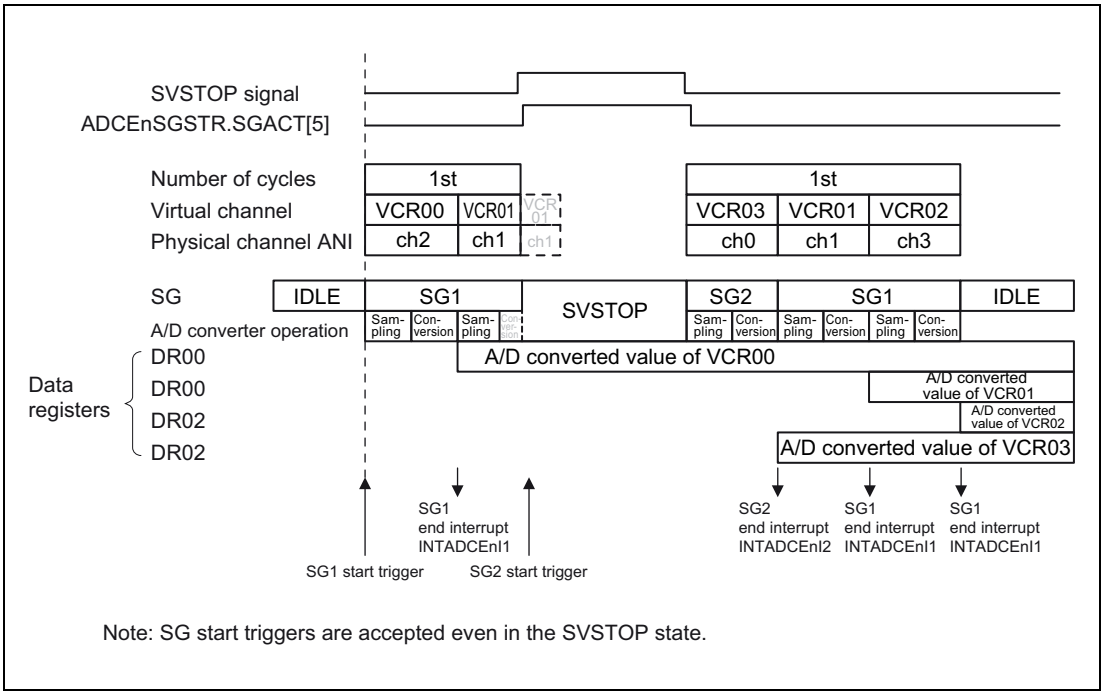
Note: SG start triggers are accepted even in the SVSTOP state.

Figure 45.18 Example of SVSTOP Operation in the IDLE State (ADCEnADCR.SUSMTD[1:0] = 00 and ADCEnEMU.SVSDIS = 0)



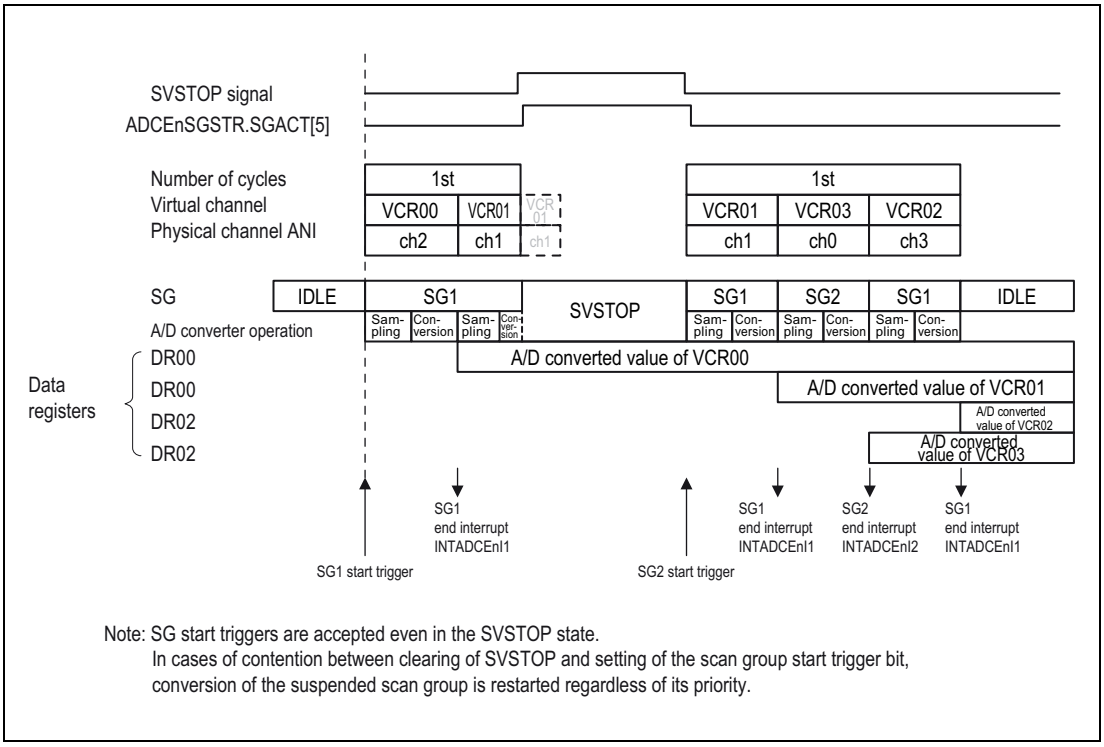
Note: An SG start trigger is accepted even during SVSTOP.

Figure 45.19 Conflict of SVSTOP Start and High-Priority SG Start Trigger (ADCEnADCR.SUSMTD[1:0] = 00, ADCEnEMU.SVSDIS = 0)



Note: SG start triggers are accepted even in the SVSTOP state.

Figure 45.20 Conflict of SVSTOP Start and High-Priority SG Start Trigger (ADCEnADCR.SUSMTD[1:0] = 10, ADCEnEMU.SVSDIS = 0)



Note: SG start triggers are accepted even in the SVSTOP state.
In cases of contention between clearing of SVSTOP and setting of the scan group start trigger bit, conversion of the suspended scan group is restarted regardless of its priority.

Figure 45.21 Conflict of SVSTOP Clear and High-Priority SG Start Trigger (ADCEnADCR.SUSMTD[1:0] = 10, ADCEnEMU.SVSDIS = 0)

45.4.11 Activating Scan Group by a Hardware Trigger

Scan group x can be activated by the hardware trigger input to SGx_TRG. As for the hardware trigger sources to be used, refer to Table 45.44, List of A/D Conversion Hardware Triggers. When activating SGx_TRG by the hardware trigger, set the peripheral function to be used by the trigger and set the start trigger in the A/D conversion trigger select control register (ADCEnSGTSELx). More than one start trigger can be specified.

45.4.11.1 Stopping Scan Group by ADHALT

Setting ADCEnADHALTR.HALT (A/D force halt trigger) to 1 forcibly halts the A/D conversion and clears the scan group x status register (ADCEnSGSTR). The error flag of ADCEnULER (upper limit/lower limit error register) is not cleared. When ADCEnADHALTR.HALT is set, make sure that ADCEnSGSTR.SGACT[5:0] has been cleared.

45.4.12 Scan End Interrupt Request

Scan group x can issue a scan end interrupt request (INT_SGx) to INTC. If ADIE of ADCEnSGCRx is set to 1, (INT_SGx) can be output after the SGx scan ends. If ADIE of ADCEnSGCRx is set to 0, the (INT_SGx) output when the SGx scan ends can be disabled. If ADIE of ADCEnVCRj is set to 1, (INT_SGx) can be output when A/D conversion for virtual channel n in SGx ends. If ADIE of ADCEnVCRj is set to 0, the (INT_SGx) output when A/D conversion for virtual channel n in SGx ends can be disabled. If ADIEs of both ADCEnSGCRx and ADCEnVCRj are set to 1 and SGx scan ending is simultaneous with A/D conversion ending for virtual channel m in SGx, the INT_SGx occurs only once.

Example 1) A scan is executed for virtual channel 0 or 1 in SG1 when ADIE of ADCEnSGCR1 is 0, ADIE of VCR0 is 1, and ADIE of VCR1 is 0.
INT_SG1 is output when A/D conversion ends for virtual channel 0.

Example 2) A scan is executed for virtual channel 0 or 1 in SG2 when ADIE of ADCEnSGCR2 is 0, ADIE of VCR0 is 1, and ADIE of VCR1 is 1.
INT_SG2 is output when A/D conversion ends for virtual channel 0 and virtual channel 1.

Example 3) A scan is executed for virtual channel 0 or 1 in SG3 when ADIE of ADCEnSGCR3 is 1, ADIE of VCR0 is 0, and ADIE of VCR1 is 0.
INT_SG3 is output when a scan ends (when A/D conversion for virtual channel 1 ends).

Furthermore, the DMAC can be started when scan ends.

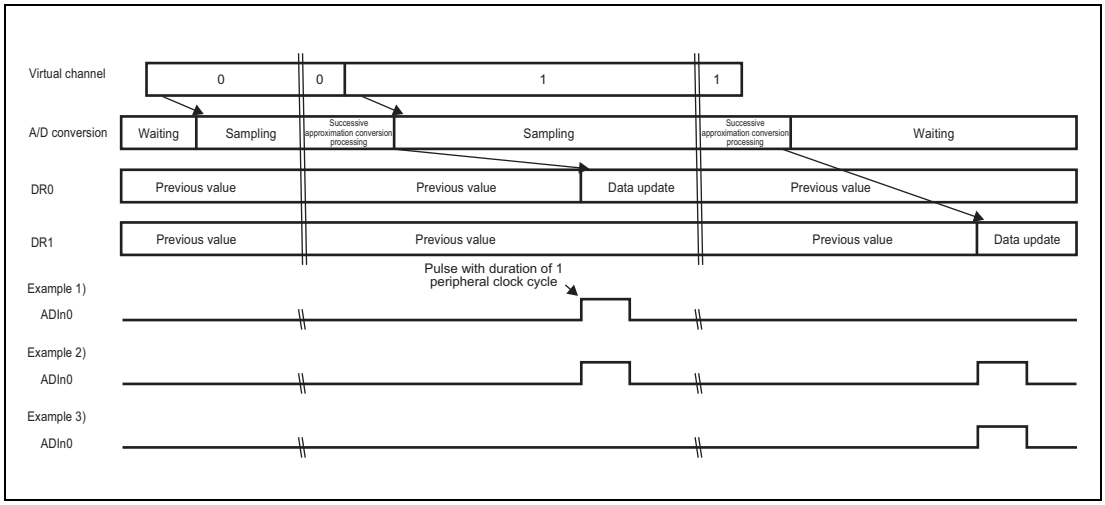


Figure 45.22 Example of a Scan Conversion End Interrupt Occurrence

NOTE

x = 1 to3

45.4.13 A/D Error Interrupt Request

ADCE can issue an A/D error interrupt request (ADEm) to ECM (Error Control Module). For an error source for which ULEIE and OWEIE of ADCEnSFTCR are set to 1, the OR condition of the error source is issued as ADEm. For an error source for which ULEIE and OWEIE of ADCEnSFTCR are set to 0, ADEm can be disabled.

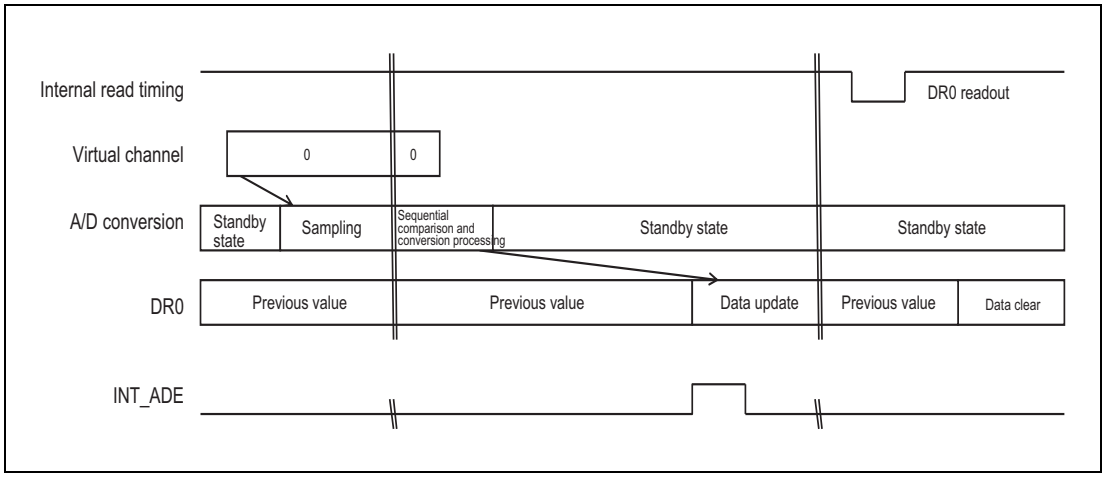


Figure 45.23 A/D Error Interrupt (Example: Overwrite Error)

45.5 Self-Diagnostic Function

To check the ADCEn function, the following self-diagnostic functions are available.

Section 45.5.1, Diagnostic of A/D Conversion Circuit

Section 45.5.2, Diagnostic of Channel Multiplexer

Section 45.5.3, Diagnostic of Open Pins

The overview of the self-diagnostic functions is shown in the figure below. The detailed description is given in the following sections.

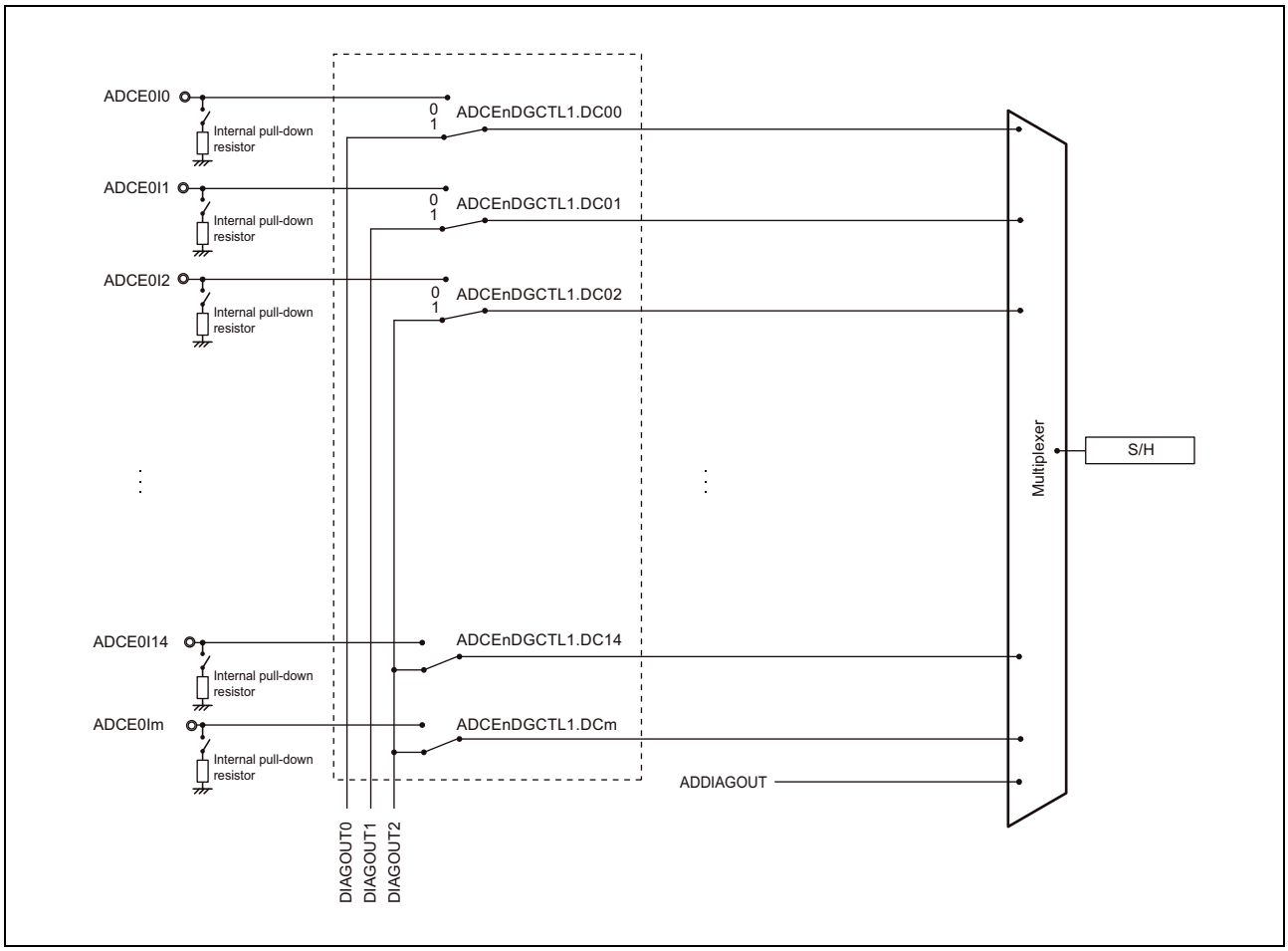


Figure 45.24 Overview of Self-Diagnostic Functions

NOTE

The functions in the dashed-line frames depend on the product.

45.5.1 Diagnostic of A/D Conversion Circuit

This function checks whether the A/D conversion is performed normally. When a voltage value is set by ADCEn.DGCTL0 and the A/D conversion result differs from an expected value, an internal circuit may be broken. The features of A/D conversion circuit self-diagnostic function is shown below.

- As the self-diagnostic voltage level, AnV_{REF}, $2/3AnV_{REF}$, $1/3AnV_{REF}$, $1/2AnV_{REF}$, and AVSS are selectable.
- The self-diagnostic of A/D conversion circuit is enabled by the A/D conversion of SG1-3.

45.5.1.1 Diagnostic procedure

1. Set ADCEnADCR.DGON = 1 to enable the self-diagnostic voltage circuit.
2. Wait for 500ns
3. Set ADCEnDGCTL0.PSEL[2:0] to select the self-diagnostic voltage level.
4. Set ADCEnADCR.DGON = 1 to update the voltage level.
5. Wait for 500ns.
6. Set arbitrary ADCEnVCRj.GCTRL[5:0] = 100100_B, to select the diagnosis channel.
7. Set ADCEnVCRj.ADIE = 1 to enable the A/D conversion end interrupt.
8. Set ADCEnSGVCSPx to specify the start pointer of virtual channel.
9. Set ADCEnSGVCEPx to specify the end Self-Diagnostic Control.
10. Generate the start trigger of scan group to perform the A/D conversion.
11. When the conversion interrupt occurs, read the result and compare it with the expected one.
12. If the result is the expected one, the A/D conversion was performed successfully.

NOTES

1. During A/D conversion, the self-diagnostic voltage level can be changed by writing to ADCEnDGCTL0.PSEL[2:0]. However, The value of ADCEnDGCTL0.PSEL[2:0] becomes effective from next A/D conversion.
2. When setting ADCEnADCR.DGON = 0, the following process needs to execute.
   1. Stop A/D conversion.
   2. Clear ADCEnDGCTL0.PSEL[2:0] bits.

45.5.2 Diagnostic of Channel Multiplexer

This function checks whether there is a problem in the route from the analog input to SAR-ADCE, by executing the A/D conversion, which is set for the different power supply, to the adjacent analog input. When the A/D conversion result is not static, an internal circuit may be broken. The different power supply is set by ADCEnDGCTL0. The features of channel multiplexer self-diagnostic function is shown below.

- Any of channels, ADCE0I0 to ADCE015, can be selected as the channel to be tested.
- As the self-diagnostic voltage level, $2/3AnV_{REF}$, $1/3AnV_{REF}$, and $1/2AnV_{REF}$ are selectable and one of the three reference voltage levels can be allocated to each channel.

Table 45.51 Selection of Channel to be Diagnosed

Connection	Select channel
DIAGOUT0	Channels 0, 3, 6, 9, 12, and 15
DIAGOUT1	Channels 1, 4, 7, 10, and 13
DIAGOUT2	Channels 2, 5, 8, 11, and 14

- Self-diagnosis of the channel multiplexer is enabled by performing A/D conversion on one of SG1 to SG3 by using multiple channels.

45.5.2.1 Diagnostic procedure

1. Set ADCEnADCR.DGON = 1 to enable the self-diagnostic voltage circuit.
2. Wait for 500ns.
3. Set ADCEnDGCTL0.PSEL[2:0] to select the self-diagnostic voltage level.
4. Set ADCEnADCR.DGON = 1 to update the voltage level.
5. Wait for 500ns.
6. Use two or more ADCEnVCRj.
   Set ADCEnVCRj.GCTRL[5:0] bits to select physical channel.
   Set ADCEnVCRj.ADIE bit to enable the A/D conversion end interrupt.
7. Set ADCEnSGVCSPx register to specify the start pointer of virtual channel.
8. Set ADCEnSGVCEPx register to specify the end pointer of virtual channel.
9. Set ADCEnDGCTL1 register to specify the Self-Diagnostic Control.
10. Generate the start trigger of scan group to perform the A/D conversion.
11. When the conversion interrupt occurs, read the result and compare it with the expected one.
12. If the result is the expected one, the A/D conversion was performed successfully.

NOTES

1. During A/D conversion, the self-diagnostic voltage level can be changed by writing to ADCEnDGCTL0.PSEL[2:0] bits. However, The value of ADCEnDGCTL0.PSEL[2:0] bits becomes effective from next A/D conversion.
2. When setting ADCEnADCR.DGON = 0, the following process needs to execute.
   1. Stop A/D conversion.
   2. Clear ADCEnDGCTL0.PSEL[2:0] bits.

45.5.3 Diagnostic of Open Pins

This function detects whether an analog input pins (ADCEnIm) is open or not by using the resistor for self-diagnostic function.

The internal pull-down resistors can be connected to diagnose the analog input pins.

Figure 45.25 Setting of On-chip Pull Down Resistor

When there is a disconnection, the conversion result is almost 0 V and it indicates an open detection.

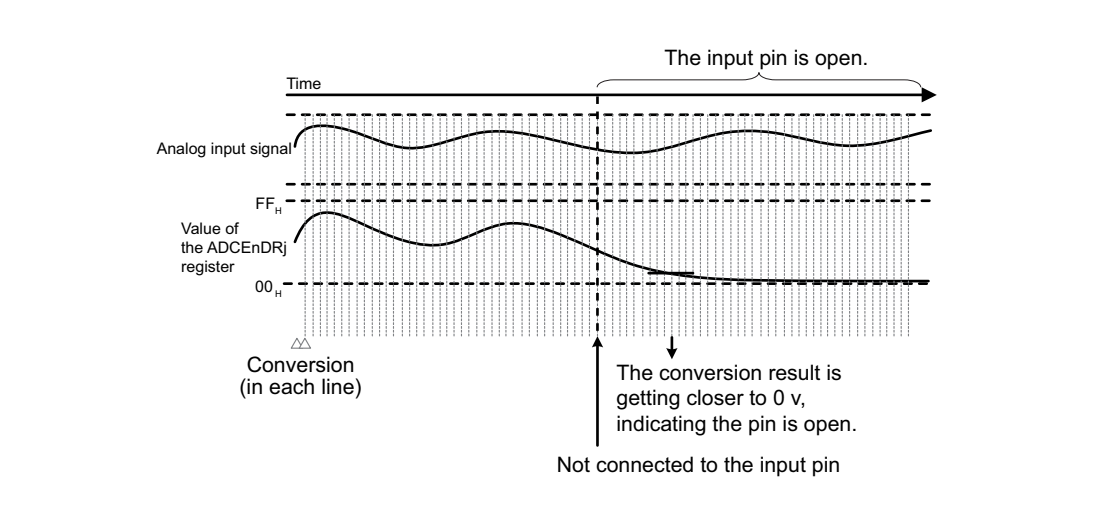


Figure 45.26 Analog Input Signal Disconnection Detection

CAUTIONS

1. **The pull-down resistors must not be connected during normal A/D conversion operation. Connected pull-down resistors may lead to a drop in the input voltage and result in erroneous A/D conversion results.**
2. **When the analog input voltage is nearly equal to the voltage level which is pulled down, a disconnection can not detected by this function.**

45.5.3.1 Diagnostic procedure

1. Set the ADCEnPDCTL2.PDNB[19:0] bits corresponding to an analog input pin (ADCEnIm) to be diagnosed to enable the pull down resistor.
2. Generate the start trigger of scan group to perform the A/D conversion.
3. Perform the A/D conversion multiple times on the same analog input.
4. Monitor the channel's A/D conversion results and check if any result declines to almost 0 V.

45.6 Definition of A/D Conversion Accuracy

A/D conversion accuracy is defined as follows:

- Resolution
  Digital output code value from the A/D converter
- Quantization error
  An error essentially contained in the A/D converter, which is assumed as 1/2 LSB (**Figure 45.27**).
- Offset error
  Deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from the minimum voltage value 000_H to 001_H. However, the quantization error is not included.
- Full scale error
  Deviation of the analog input voltage value from the ideal A/D conversion characteristics when the digital output changes from FFE_H to FFF_H. However, the quantization error is not included.
- DNL (Differential nonlinear error)
  Deviation between the ideal digital output code width (Vq) and the actual digital output code width (Va), which is assumed as (Va – Vq)/Vq. However, the offset error, the full scale error, and the quantization error are not included .
- INL (Integral nonlinear error)
  Deviation of the actual value from the ideal A/D conversion characteristics, from the zero voltage to the full scale voltage, which is assumed as an integral of DNL from 000_H to a digital output code. However, the offset error, the full scale error, and the quantization error are not included.
- Absolute accuracy
  Deviation between the digital value and the analog input value. The offset error, the full scale error, the quantization error, DNL, and INL are included .

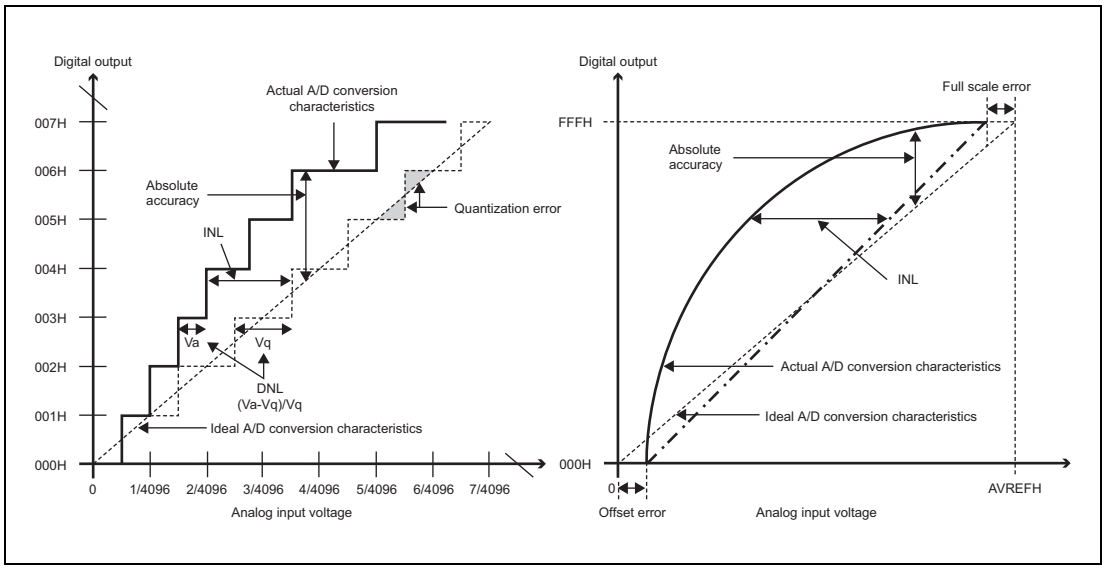


Figure 45.27 Definition of A/D Conversion Accuracy

45.7 Usage Notes

45.7.1 Range of Channel Input Voltage

CAUTION

ADCEnIm input voltage should be used within the specification range. If the channel input voltage exceeds AnVREF or falls below AVSS, the converted value of the channel is saturated and may influence electric characteristics of other channels.

When an over-voltage is applied to an ADCEnIm pin and at the same pin the ADC self-diagnosis (diagnosis of channel multiplexer) is executed, an offset voltage to the diagnosis voltage can be measured. According to the diagnosis setting, adjacent pins are affected from the pin which has over-voltaged. There are two advices for avoiding the problem. The group which have no injected current can be measured with no influence.

Case 1: Injected current is applied at one pin where the ADC self-diagnosis is executed

First ADC self-diagnosis conversion cycle:

→ only ADCE0I0 is selected (CDG0=1, CDG1-15 = 0) ← injected current

Second ADC self-diagnosis conversion cycle:

→ ADCE0I1-15 are selected (CDG0=0, CDG1-15 = 1) ← no injected current

Case2: Injected current is applied at multiple pins where the ADC self-diagnosis is executed

First ADC self-diagnosis conversion cycle:

→ only ADCE0I0 is selected (CDG0 = 1, CDG1-15 = 0) ← injected current

Second ADC self-diagnosis conversion cycle:

→ only ADCE0I1 is selected (CDG0 = 0, CDG1 = 1, CDG2-15 = 0) ← injected current

Third ADC self-diagnosis conversion cycle:

→ only ADCE0I2 is selected (CDG0 = 0, CDG1 = 0, CDG2 = 1, CDG3-15 = 0) ← injected current

n ADC self-diagnosis conversion cycle:

→ ADCE0In-15 are selected (CDGn-1 = 0, CDGn to CDG15 = 1) ← no injected current

45.7.2 Usage Notes for Analog Input Pins

- The write access to the PWM-Diag related registers is prohibited unless the PWM-Diag is used. If read, the read value is undefined.
- If digital input through the pin is not essential, disable digital input.
- If a digital signal is input to such a pin, the signal should not include overshot or undershot. Such overload condition may lead injected current.
- Set the digital input voltage to A0VCC.
  Operation of input voltages outside the range of -0.3 V < A0VCC < +0.3 V may negatively affect the reliability of the chip.
- If digital signal is output to such a pin, reduce the output load capacitance as much as possible. Such increased discharge currents from digital output pin load capacitance, may decrease the ADC accuracy.

45.7.3 Relationship between analog input voltage and A/D conversion result

The relationship between the analog input voltage input to the analog input pin(ADCEnIm) and the A/D conversion results (ADCEnDRj[31:16], ADCEnDRj[15:0], ADCEnDIRj[15:0]) is expressed as follows.

$$A/D\ conversion\ result\ value = INT\left(\frac{VIAN - A0GND}{A0VREF0 - A0GND} \times 2^k + 0.5\right)$$

INT( ): Function that returns the integer part of the value in parentheses

VIAN: Analog input voltage

A0VREF: A0VREF pin voltage

A0GND: A0GND pin voltage

k: Resolution

Section 46 Intelligent Stepper Motor Driver (ISM)

This section contains a generic description of the Intelligent Stepper Motor Driver (ISM).

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

46.1 Overview of RH850/D1L/D1M Intelligent Stepper Motor Driver (ISM)

46.1.1 Units

This microcontroller has the following number of units of the Intelligent Stepper Motor Driver (ISM).

Table 46.1 Number of Units

Product Name	All products
Units	1
Names	ISM0

Units index n

Throughout this section, the individual units of an Intelligent Stepper Motor Driver is identified by the index "n" (n = 0), for example ISMnCCMRm for the ISMn channel control register.

46.1.2 Channel index m

Throughout this section, the individual motor channels are identified by the index “m”, for example ISMnCCMRm for the ISM channel control register.

The number of channels for each device is given in the following table:

Table 46.2 Motor control channels

	D1L1	D1L2(H)	D1M1(H) D1M1-V2 D1M1A	D1M2	D1M2H
Number of channels	6	6	6	4	6
Indices	m = 0 to 5	m = 0 to 5	m = 0 to 5	m = 0, 1, 4, 5	m = 0 to 5

46.1.3 Register addresses

All Intelligent Stepper Motor Drivers register addresses are given as address offsets from the individual base addresses <ISMn_base>.
The <ISMn_base> addresses of each ISMn are listed in the following table:

Table 46.3 Register base addresses <ISMn_base>

ISMn unit	<ISMn_base> address
ISM0	$FFF0\ 0000_H$

46.1.4 Clock supply

All Intelligent Stepper Motor Drivers provide one clock input.

Table 46.4 Clock supply

ISMn unit	ISMn clock	Connected to
ISM0	PCLK	Clock Controller C_ISO_ISM

46.1.5 Interrupts

The Intelligent Stepper Motor Drivers can generate the following interrupt requests:

Table 46.5 ISMn interrupt requests

ISMn signals	Function	Connected to
ISM0:		
IRQ_REACHED	Indicates that a motor has reached its target/zero position	Interrupt Controller INTISM0REACHED
IRQ_DONE	Indicates that the sequencer has returned into idle state for the current channel data update interval	Interrupt Controller INTISM0DONE
IRQ_ZPDAD	Indicates the actual start of ZPD measurements	Interrupt Controller INTISM0ZPDAD
IRQ_ZPD	Indicates that a motor has *not* reached its zero position.	Error Control Module

46.1.6 Reset sources

The Intelligent Stepper Motor Drivers and their registers are initialized by the following reset signal:

Table 46.6 Reset sources

ISMn unit	Reset signal
ISMn	• Reset Controller SYSRES • Reset Controller ISM0RES • reset upon wake-up from DEEPSTOP mode

CAUTION

By default the ISM0RES reset is active.
Thus before accessing this module this reset must be released via the MRSTC register of the Reset Controller. Refer to Section 9, Reset Controller for details.

46.1.7 I/O signals

The following table shows the I/O signals of the Intelligent Stepper Motor Drivers.

Table 46.7 ISMn I/O signals

ISM0 signals	Function	Connected to
Motor Driver 0:		
SINP0	Plus side of vertical coil	Port ISM11
SINM0	Minus side of vertical coil	Port ISM12
COSP0	Plus side of horizontal coil	Port ISM13
COSM0	Minus side of horizontal coil	Port ISM14
Motor Driver 1:		
SINP1	Plus side of vertical coil	Port ISM21
SINM1	Minus side of vertical coil	Port ISM22
COSP1	Plus side of horizontal coil	Port ISM23
COSM1	Minus side of horizontal coil	Port ISM24
Motor Driver 2 (not available in D1M2 devices):		
SINP2	Plus side of vertical coil	Port ISM31
SINM2	Minus side of vertical coil	Port ISM32
COSP2	Plus side of horizontal coil	Port ISM33
COSM2	Minus side of horizontal coil	Port ISM34
Motor Driver 3 (not available in D1M2 devices):		
SINP3	Plus side of vertical coil	Port ISM41
SINM3	Minus side of vertical coil	Port ISM42
COSP3	Plus side of horizontal coil	Port ISM43
COSM3	Minus side of horizontal coil	Port ISM44
Motor Driver 4		
SINP4	Plus side of vertical coil	Port ISM51
SINM4	Minus side of vertical coil	Port ISM52
COSP4	Plus side of horizontal coil	Port ISM53
COSM4	Minus side of horizontal coil	Port ISM54
Motor Driver 5		
SINP5	Plus side of vertical coil	Port ISM61
SINM5	Minus side of vertical coil	Port ISM62
COSP5	Plus side of horizontal coil	Port ISM63
COSM5	Minus side of horizontal coil	Port ISM64

NOTE

If the zero point detection (ZPD) function is used, the ports, used for the ZPD input signals have to be configured for that purpose. Refer to Section 46.2, ZPD I/O buffer setting register ISMCHNCFG for details.

46.2 ZPD I/O buffer setting register ISMCHNCFG

This register configures the port I/O buffers to be used by the ZPD function.

Access: This register can be read/written in 32-bit units.

Address: FFC0 6028_H

Initial Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	PRMR06SSW	0	PRMR05SSW	0	PRMR04SSW	0	PRMR03SSW	0	PRMR02SSW	0	PRMR01SSW	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.8 ISMCHNCFG register contents

Bit position	Bit name	Function
31 to 12	Reserved	When written, write the initial value.
11	PRMR06SSW	Usage of P17_11 to P17_8 for ZPD function 0: ports are not used for ZPD input 1: ports are used for ZPD input
10	Reserved	When written, write the initial value.
9	PRMR05SSW	Usage of P17_7 to P17_4 for ZPD function 0: ports are not used for ZPD input 1: ports are used for ZPD input
8	Reserved	When written, write the initial value.
7	PRMR04SSW	Usage of P17_3 to P17_0 for ZPD function 0: ports are not used for ZPD input 1: ports are used for ZPD input
6	Reserved	When written, write the initial value.
5	PRMR03SSW	Usage of P16_11 to P16_8 for ZPD function 0: ports are not used for ZPD input 1: ports are used for ZPD input
4	Reserved	When written, write the initial value.
3	PRMR02SSW	Usage of P16_7 to P16_4 for ZPD function 0: ports are not used for ZPD input 1: ports are used for ZPD input
2	Reserved	When written, write the initial value.
1	PRMR01SSW	Usage of P16_3 to P16_0 for ZPD function 0: ports are not used for ZPD input 1: ports are used for ZPD input
0	Reserved	When written, write the initial value.

NOTE

In the header files the module name of the above register is defined as:
SELB.

46.3 Functional Overview

The Intelligent Stepper Motor Driver provides up to 6 drivers for stepper motors that are connected to external meters. Calibration of the meters is supported by the built-in zero point detection (ZPD) function.

Channel management feature, together with predetermined RAM tables, can be used in motor driving operation mode and in ZPD operation mode to carry out calculations with predefined parameters.

NOTE

The number of stepper motor drivers is defined in the first section of this chapter under the key word *"Channel index m"*.

Features summary

General features of the Intelligent Stepper Motor Driver:

- Control of up to 6 individual stepper motors
- Generation of pulse width modulated (PWM) output signals with the following characteristics:
  - Configurable PWM frequency
  - Pulse width with a precision of 10 bits
  - Configurable delay of PWM output signals for reducing fluctuation of power supply and for reducing the susceptibility to electromagnetic interference

Features for driving stepper motors:

- Generation of four output signals to drive a stepper motor's coils
- Half-bridge circuit or full-bridge circuit configuration compatibility (see I/O characteristics of device)
- Recirculation function to reduce the inductive load in the stepper motor's coils
- Advanced features of channel management:
  - Automatic calculation of intermediate positions from a target position
    Appropriate PWM signal information is taken from configurable PWM value look-up tables.
  - Storage of the stepper motor's characteristics in movement parameters
  - Configurable data update intervals
  - Interrupt request generation when a motor has reached its target position

Features for ZPD:

- Generation of four output signals to drive the stepper motor, with selectable signal generation method:
  - PWM signal generation as for motor driving
  - Direct control
- ZPD measurements with the following features:
  - Adjustable measurement clock and measurement delays to adjust ZPD measurement speed and window
  - Selection of the input to be measured
  - Comparison of input voltage with one of two configurable reference voltages
  - Digital noise filtering of comparison results with configurable filter parameters
  - Status flags and interrupt request to evaluate ZPD measurement results
- Power save mode
- Advanced features of channel management:
  - Automatic control of driving and measurements until zero point is detected
  - Configurable update intervals
  - Interrupt request generation when the zero position is reached

46.4 Functional Description

The Intelligent Stepper Motor Driver (ISMn) can control up to 6 independent stepper motors via motor channels.

Each of the six motor channels can be used to drive a stepper motor or for zero point detection.

46.4.1 Functional description stepper motor driving

Typically, a stepper motor has a horizontal and a vertical coil, each with two poles that are arranged opposite each other. Applying current to the coils in a specific pattern forces the rotor to move.

The Intelligent Stepper Motor Driver generates PWM signals to supply the horizontal and vertical coil with current. The vertical PWM signal comprises the sine value and the horizontal PWM signal comprises the cosine value of the desired rotor position (angle).

Since the PWM signals do not inherit a sign, quadrant information indicates whether the sine or cosine side is negative or positive.

Block diagram

Each stepper motor is driven individually. The following figure illustrates how ISMn drives *one* stepper motor.

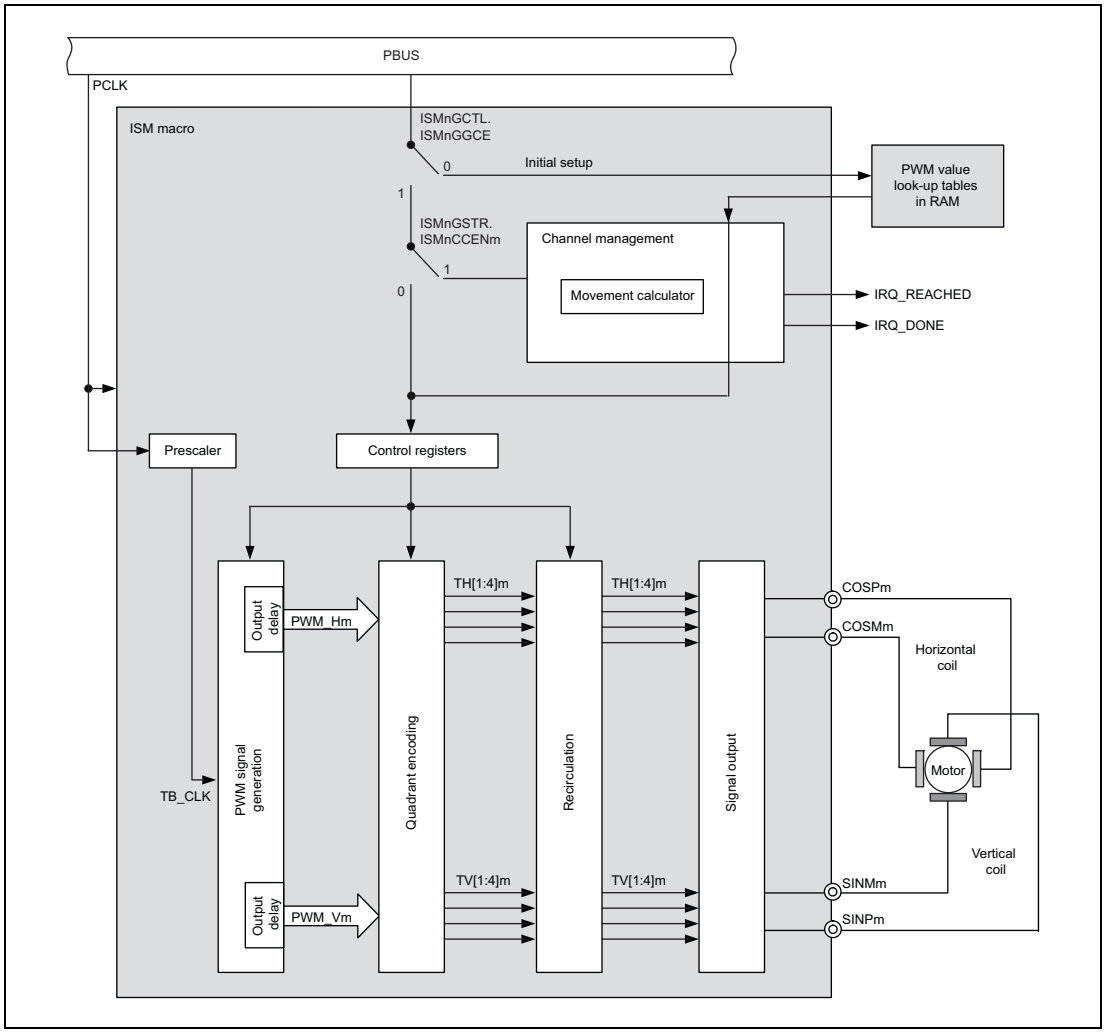


Figure 46.1 Block diagram of stepper motor driving of motor channel m

PWM signal generation

ISMn generates PWM signals. The PWM signal information (e.g. pulse width and quadrant information) can be applied in two different ways: with or without channel management.

- *With* channel management, only the desired target position of the motor needs to be supplied to ISMn. The rotor is moved step-wise to the target position whereby ISMn calculates intermediate positions using pre-determined movement parameters. Appropriate PWM signal information is taken from configurable PWM value look-up tables.

  An interrupt request and status flags indicate when a motor has reached its target position.
- *Without* channel management, the application software provides the PWM signal information.

Output delay

To reduce electromagnetic interferences (EMI noise), the PWM output can be delayed so that, for example, only one PWM edge can be seen at one time.

Quadrant encoding

Each PWM signal is encoded to 4 signals according to its quadrant. This allows the sides of both coils to be driven as required.

Recirculation

Recirculation can be used to prevent the generation of inductive voltage when a PWM signal is low and cuts-off the motor coil from current.

Signal output

The signals are output as required by the type of bridge circuit provided in the I/O buffers of the device. In a full-bridge circuit configuration, for example, the signal level is automatically adapted according to the P and N-channel FET types.

The bridge circuit of the port I/O structures then supplies the sides of the coils with current in the required manner.

46.4.2 Functional description zero point detection

A motor typically needs to be positioned at the “home” or zero position during power-up or after a position fault (e.g. after the motor power has been cut off).

Zero point detection

When the motor stops rotating, for example because the meter hits a mechanical stop, the back electromotive force (back EMF) of the motor declines to zero.

This principle can be used to bring a motor into its zero position: the rotor is rotated step-wise while the voltage is measured at an open end of an un-driven coil at specified times. The measured voltage is compared to a pre-determined reference voltage. When the measured voltage is below the threshold, the motor has reached its zero position.

Block diagram

Each stepper motor is driven individually. The following figure illustrates how ZPD is performed for *one* stepper motor.

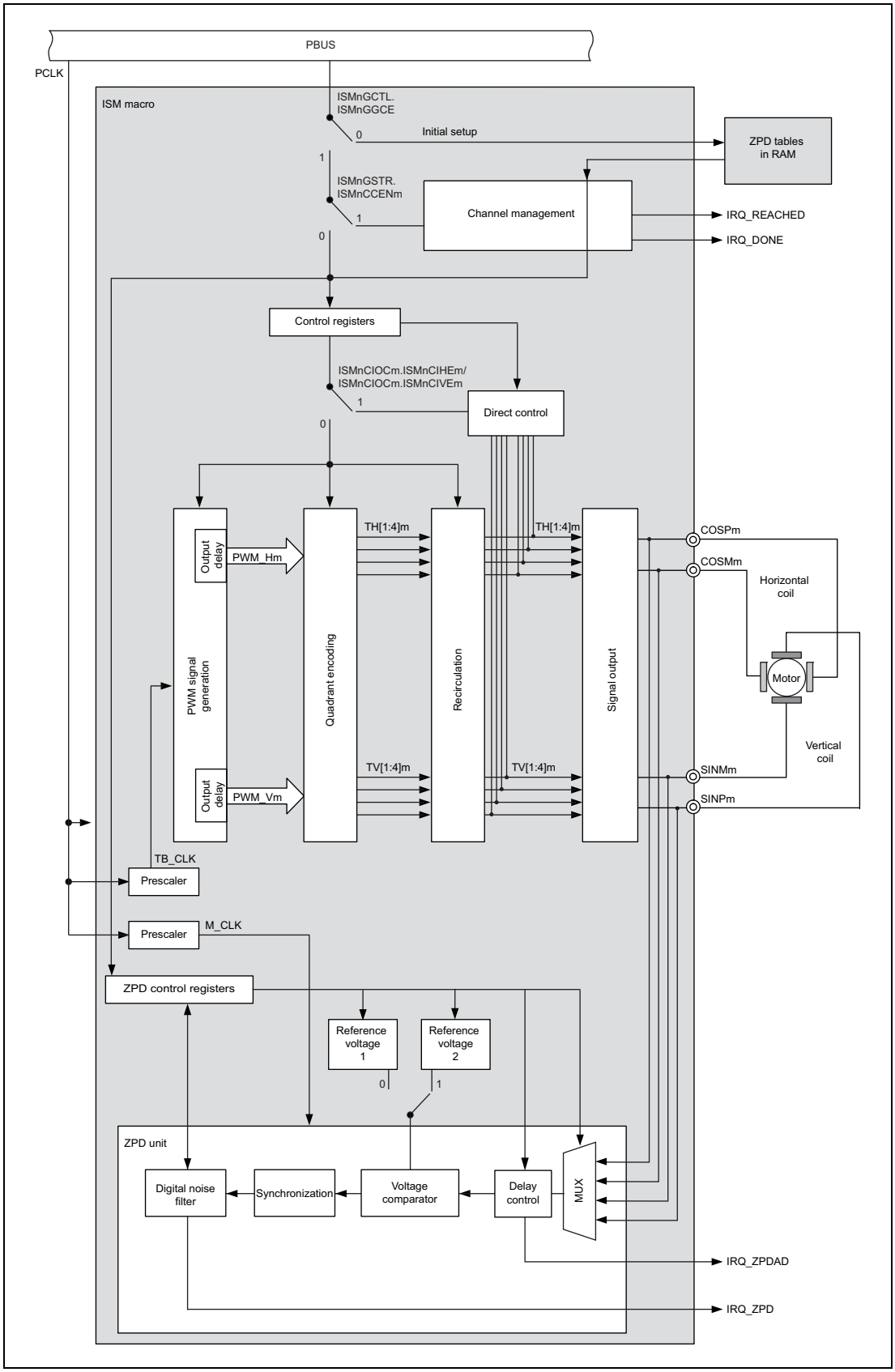


Figure 46.2 Block diagram of zero point detection of channel m

Zero point detection can be performed in two different ways: with or without channel management.

- *With* channel management, a pre-determined zero point detection procedure is started only once.

  ISMn moves the rotor and performs voltage measurements according to the step and measurement procedure stored in a RAM table.
  An interrupt request indicates that the motor has reached its zero position.
- *Without* channel management, the application provides all control necessary for motor driving and voltage measurements.

46.4.2.1 Motor driving

As in motor driving mode, PWM signals can be used to move the rotor.

Direct control

ISMn also allows the bridges to be driven directly. Direct control can be enabled/disabled for every motor channel individually.

46.4.2.2 ZPD measurements for position verification (ZPD unit)

Both ends of the horizontal and vertical coils can be measured (plus side and minus side).

The ZPD measurement frequency is configurable.

Blanking delay and measurement delay

Additionally, ZPD measurements can be delayed allowing a very precise timing of ZPD measurements, for example to start measuring at times that best suit a motor's characteristics.

Both a blanking delay at the start of a ZPD measurement and a measurement delay between measurement sequences are configurable.

Voltage comparator & synchronization

The voltage measured for a motor channel is compared to one of two pre-determined reference voltages.

The output of the voltage comparator is synchronized with the measurement clock M_CLK and forwarded to the digital noise filter.

Digital noise filtering

For reliable results, voltage measurements of a short pulse and spikes can be filtered out using digital noise filtering.

The interrupt IRQ_ZPD indicates when the zero position has not been reached during a measurement sequence.

Power save mode

To reduce power consumption, the comparators and reference voltage dividers can be disabled during that time, the ZPD operation mode is not used.

46.4.3 Clocking

The timebase for all channel operations is TB_CLK.

The PCLK prescaler for TB_CLK is configured in ISMnGCFG.ISMnGGTB[3:0].

TB_CLK = PCLK/N

where N = ISMnGCFG.ISMnGGTB[3:0] + 1

NOTE

TB_CLK can only be modified when channel management is switched off (ISMnGCTL.ISMnGGCE = 0).

46.4.4 Channel management

As already stated in the functional descriptions, channel management can be used in motor driving and ZPD operation modes.

No matter whether channel management is used by a motor channel or not, the PWM signals that drive the stepper motor are generated and processed in the same way. ZPD measurements are also performed regardless.

The procedures differ in how or “by whom”, and how often the position, movement, and measurement parameters need to be supplied to ISMn.

- Without channel management, the application software provides all information necessary for every single motor step and voltage measurement.
  PWM signal and ZPD measurement information are read from designated registers and become effective immediately.
  An interrupt request and status flags can indicate that a motor has not reached its zero position yet and further steps must be taken.
- When channel management is set for a motor channel that is operated in motor driving mode, only the desired target position of the motor needs to be supplied to ISMn.
  ISMn automatically controls the movement of the rotor until the rotor has reached the desired target position. Substeps are calculated using configurable PWM value look-up tables and movement parameters. Depending on a motor’s configured precision, a full rotation is divided into 512 or 128 steps. An interrupt request and status flags indicate when a rotor has reached its target position.
- When channel management is set for a motor channel that is operated in ZPD operation mode, the pre-determined zero point detection procedure only needs to be started.
  ISMn moves the rotor and performs voltage measurements according to the step and measurement procedure stored in a RAM table.
  When a motor has reached its zero position, channel management automatically disables ZPD operation mode for the respective channel. Additionally an interrupt request can be generated.

Advantages

Channel management reduces the number of calculations and the amount of configuration the application has to perform:

- Typically, most of the configuration needs to be done only once (after enabling/resetting ISMn).
- Multiple motors with the same or similar design and characteristics can share a PWM value look-

up table and/or a ZPD control sequence.

46.4.4.1 Channel management start/stop

The use of channel management can be enabled/disabled for every motor channel individually in ISMnGSTR.ISMnCCENm. Channel management itself is switched on/off by ISMnGCTL.ISMnGGCE.

- When ISMnGCTL.ISMnGGCE = 1, channel management is switched on.
  All motor channels where ISMnGSTR.ISMnCCENm = 1 are supplied by channel management.
- When ISMnGCTL.ISMnGGCE = 0, channel management is switched off for *all* channels although ISMnGSTR.ISMnCCENm = 1.
  For example, the rotors hold their position unless the pulse width, quadrant information, etc. are updated by the application.

The following table shows when a motor channel is supplied by channel management and when not.

Table 46.9 Channel management

ISMnGCTL.ISMnGGCE	ISMnGSTR.ISMnCCENm	Supplied by channel management?
1	1	Yes
	0	No
0	1	No
	0	No

NOTE

Modifying general configuration data requires channel management to be switched off. This includes the general settings and the RAM tables.
Modifying a channel's specific settings, usually only requires the usage of channel management to be disabled for that channel.

For details about the recommended procedures refer to

- Section 46.4.12.2, Basic procedures: Motor driving with channel management
- Section 46.4.12.4, Basic procedures: ZPD with channel management.

46.4.4.2 Channel data update interval

The data of motor channels for which channel management is set, is automatically updated in a configurable interval.

The so-called channel data update interval is a multiple of PWM cycles. It is therefore related to the PWM frequency f_{PWM}. The update is triggered by a START event which is issued at the beginning of every channel data update interval.

The new values take effect at the beginning of the next PWM cycle (see Section 46.4.5.1, PWM cycle time and PWM frequency for details).

The following figure illustrates the relation between the PWM cycle time (1) and the channel data update interval (2).

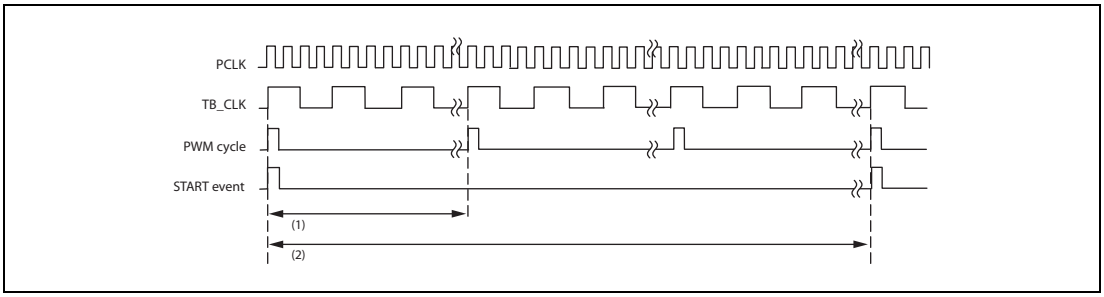


Figure 46.3 PWM cycle time and channel data update interval

NOTE

Values for channels where channel management is not set can be modified at any time. The new values take effect at the beginning of the next PWM cycle also – independent of START events.

Configuration of update interval

The channel data update interval can be calculated by the following formula:

$$\text{Update interval} = M / f_{PWM}$$

$$M = \text{Update interval} \times f_{PWM}$$

$$= (\text{Update interval} \times \text{TB\_CLK}) / (2^{10} - 1)$$

M = ISMnGCFG.ISMnGGUD[7:0] + 1.

Examples

The channel data update interval should be about 1.1 ms while TB_CLK is 8 MHz.

$$M = 1.1\ \text{ms} \times 8{,}000\ \text{kHz} / 1{,}023$$

$$= 8.602$$

The channel data update interval is 1.125 ms when M = 9 (ISMnGCFG.ISMnGGUD[7:0] = 8_H).

The following table provides M values for a channel data update interval of approximately 1.1 ms for recommended f_{PWM}.

Table 46.10 Recommended settings

f_{PWM}	M (ISMnGCFG.ISMnGGUD[7:0] + 1)	Update interval
8 kHz	9	1.125 ms
16 kHz	18	1.125 ms
32 kHz	35	1.093 ms
64 kHz	70	1.093 ms

Refer to Section 46.4.9.1, Channel data update timing to see when new values are applied.

46.4.5 Generation and processing of PWM signals to drive a motor

To drive a stepper motor, two synchronized PWM signals are generated according to the desired rotor position's sine and cosine values.

Provision of PWM signal information

The PWM signal information is supplied differently depending on whether channel management is set for a motor channel or not, and the channel's operation mode.

Exception

Direct control in ZPD operation mode allows the transistors that supply the horizontal and vertical coil with current to be activated directly. Refer to Section 46.4.7.1, Motor driving for details.

46.4.5.1 PWM cycle time and PWM frequency

The PWM cycle time and PWM frequency take effect for all channels:

- The PWM cycle time is 2^{10}-1 (1,023) cycles of the scalable timebase TB_CLK.
  All channels are processed once within *one* PWM cycle – at the beginning of a new PWM cycle.
- The PWM frequency f_{PWM} is given by the following formula:
  f_{PWM} = TB_CLK / (2^{10} - 1)
  Reasonable f_{PWM} values are 8 kHz, 16 kHz, 32 kHz, and 64 kHz.

Refer to Section 46.4.3, Clocking for details about configuring TB_CLK.

46.4.5.2 Duty cycle time and duty factor of PWM signals

The 10-bit free running up counter ISMnGCNT.ISMnGGTR[9:0] counts TB_CLK. The counter's maximum value is 2^{10} - 2.

The horizontal and vertical PWM signals are set to high at the beginning of a PWM cycle and are reset to low when the counter matches the configured duty cycle time.

Configuration of duty cycle time

The duty cycle times for the horizontal and vertical PWM signals can be set individually, each to a value between 0 and 2^{10} - 1 cycles of TB_CLK.

- ISMnCCMPm.ISMnCCVPm[9:0] defines the value for a channel's *vertical* PWM signal (corresponds to the sine value of the rotor's position).
- ISMnCCMPm.ISMnCCHPm[9:0] defines the value for a channel's *horizontal* PWM signal (corresponds to the cosine value of the rotor's position).

ISMnCCMPm.ISMnCCVPm[9:0] and ISMnCCMPm.ISMnCCHPm[9:0] are supplied differently depending on whether channel management is set for a motor channel or not, and the channel's operation mode.

NOTES

1. The output of a PWM signal is switched off when the configured value is "0".

2. The PWM signal is steady high when the maximum value "2^{10} - 1" is configured. With this value set, the counter never matches the configured duty cycle time and the PWM is not reset to low.

Duty cycle

The duty cycle of the PWM can be calculated by the following formula:

Duty cycle % = [D / (2^{10} - 1)] x 100 %

where D = ISMnCCMPm.ISMnCCHPm[9:0] or D = ISMnCCMPm.ISMnCCVPm[9:0].

Examples

The following figure illustrates some examples of horizontal PWM signals with different duty cycles.

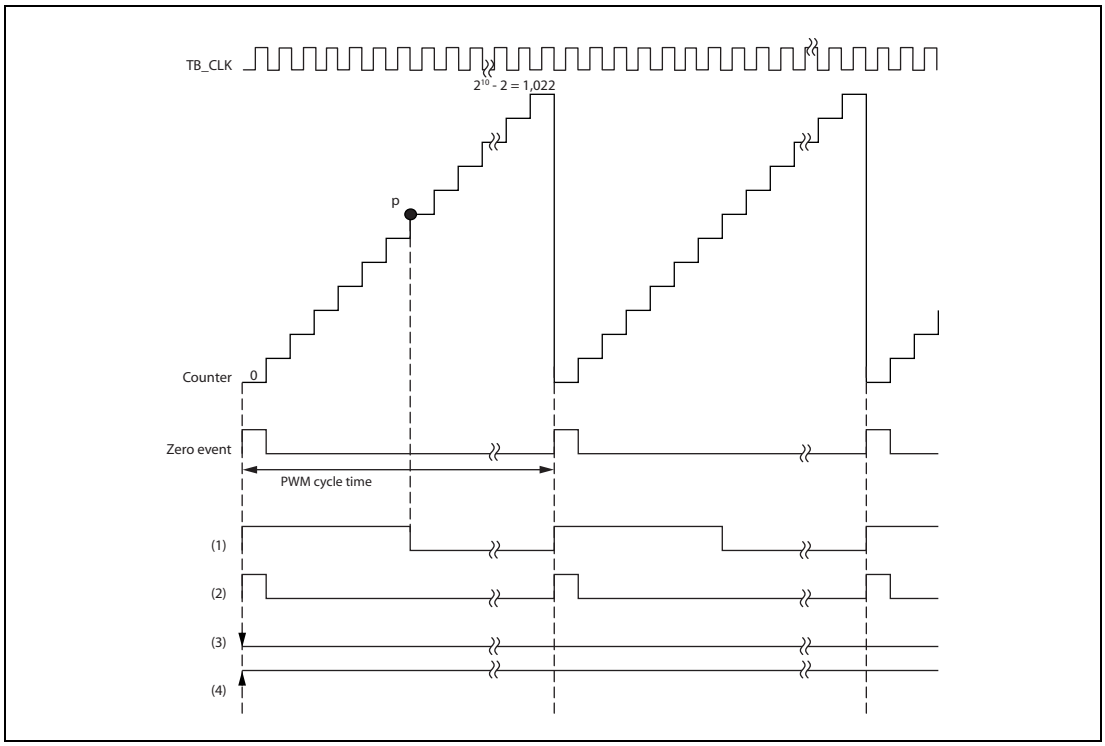


Figure 46.4 Examples of different PWM signals

The following settings are valid in the figure above:

	ISMnCCMPm.ISMnCCHPm[9:0]	Duty cycle
(1)	p	[p / (2^{10} - 1)] x 100 %
(2)	1	[1 / (2^{10} - 1)] x 100 % = 0,097 %
(3)	0 (min.)	0 %
(4)	2^{10} - 1 (max.)	100 %

The PWM signal is output without delay in all cases (ISMnCCMRm.ISMnCCDHm[3:0] = 0).

46.4.5.3 Output delay

Simultaneous switching of sine and cosine output can increase the susceptibility to electromagnetic interference (EMI noise).

To reduce EMI noise, the output of PWM signals can be delayed so that, for example, only one PWM signal edge can be seen at one time.

The output can be delayed by 1 to 15 clock cycles of TB_CLK.

- ISMnCCMRm.ISMnCCDHm[3:0] specifies the output delay for the *horizontal* PWM signal (corresponds to the cosine value of the rotor's position).
- ISMnCCMRm.ISMnCCDVm[3:0] specifies the output delay for the *vertical* PWM signal (corresponds to the sine value of the rotor's position).

46.4.5.4 Encoding of PWM signals according to quadrant

The PWM signals are encoded into 4 signals each as required by the type of bridge circuit implemented in the port structure.

- The *horizontal* (cosine) PWM signal is encoded to TH[1:4]m.
- The *vertical* (sine) PWM signal is encoded to TV[1:4]m.

Encoding

Since the PWM signals do not inherit a sign, the encoding depends on the quadrant information in ISMnCCMPm.ISMnCCQIm[1:0] and follows the logic in the following table:

Table 46.11 Encoding of PWM signals according to quadrant

			Output Result							
Quadrant information			Encoding of horizontal PWM signal				Encoding of vertical PWM signal			
ISMn CCQIm[1:0]	Quadrant	Angle	TH1m	TH2m	TH3m	TH4m	TV1m	TV2m	TV3m	TV4m
00_B	0	0° - 90°	PWM_H	0	0	1	PWM_V	0	0	1
01_B	1	90° - 180°	0	1	PWM_H	0	PWM_V	0	0	1
10_B	2	180° - 270°	0	1	PWM_H	0	0	1	PWM_V	0
11_B	3	270° - 360°	PWM_H	0	0	1	0	1	PWM_V	0

ISMnCCMPm.ISMnCCQIm[1:0] is supplied differently depending on whether channel management is set for a motor channel or not, and the channel's operation mode.

NOTE

The encoding is the same in a half-bridge or full-bridge circuit implementation. Depending on the implementation, the resulting signals are reshaped.

Refer to Section 46.4.5.6, Signal output for details.

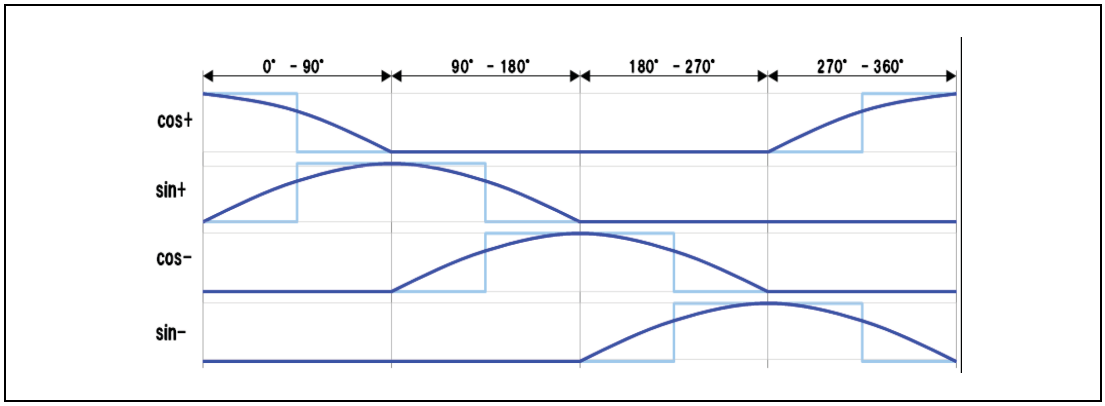


Figure 46.5 Reshaping of motor signals by microstep movement

46.4.5.5 Recirculation

Current flowing through a coil creates a magnetic field which crashes at the instant the current is switched off, for example because the corresponding PWM signal is in its low phase. The crash of the magnetic field induces a high load across the coil for a short time which can damage the transistors of the bridge.

Recirculation discharges inductive load when the horizontal and/or vertical PWM signals are low. The magnetic field dies away rather than crashing instantly. This prevents the induced load becoming too high.

CAUTION

Recirculation must be enabled (ISMnCIVRm and ISMnCIHRm set) when using the ISMn with inductive loads (such as the coils of the stepper motors).
Otherwise, severe damage to the output stages of the I/O ports may result.

Redirection to ground connection or power supply

The current is redirected reducing the effect of flyback voltage caused by switching the inductive load.

- The current is directed via the ground connection $ISMV_{SS}$ when ISMnCOPT.ISMnCIRSm = 0 (default).
- The current is directed via the power supply $ISMV_{CC}$ when ISMnCOPT.ISMnCIRSm = 1.

NOTE

ISMnCOPT.ISMnCIRSm can only be modified when channel management is switched off (ISMnGCTL.ISMnGGCE = 0).

Enabling/disabling recirculation

Recirculation can be enabled/disabled for each channel individually:

- ISMnCIOCm.ISMnCIHRm enables/disables the recirculation of a channel's *horizontal* (cosine) PWM signal.
- ISMnCIOCm.ISMnCIVRm enables/disables the recirculation of a channel's *vertical* (sine) PWM signal.

ISMnCIOCm.ISMnCIHRm and ISMnCIOCm.ISMnCIVRm are supplied differently depending on whether channel management is set for a motor channel or not, and the channel's operation mode.

46.4.5.6 Signal output

Bridge circuits supply current to the positive and negative sides of the horizontal and vertical coils. ISMn supports a half-bridge and a full-bridge circuit implementation.

Depending on the bridge type implemented, different signals are used to drive the transistors of the bridges.

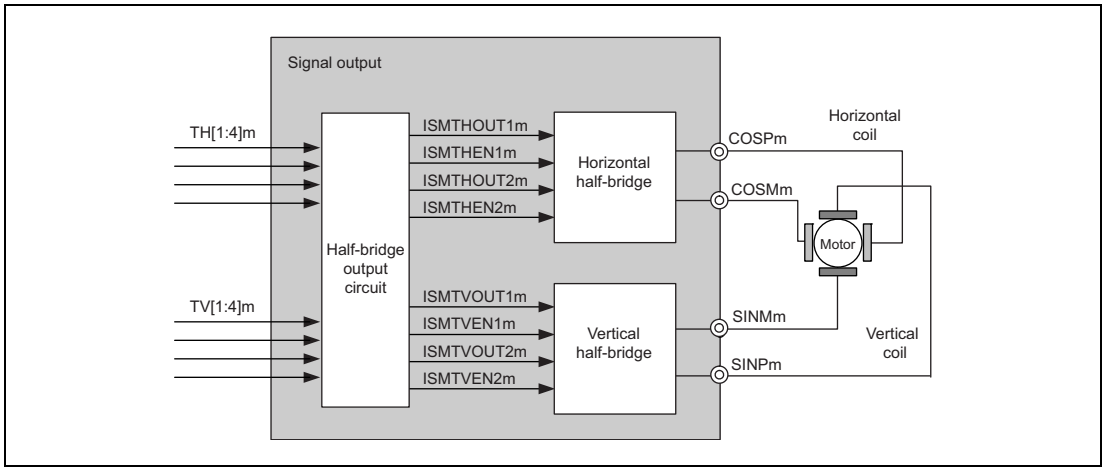


Figure 46.6 Signal output in half-bridge circuit configuration

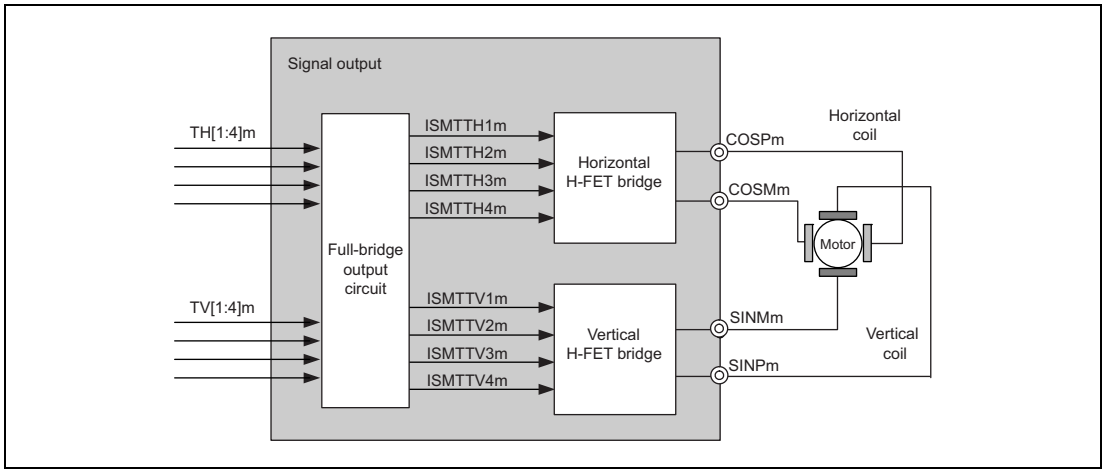


Figure 46.7 Signal output in full-bridge circuit configuration

Half-bridge circuit configuration

To match with a half-bridge circuit structure, the signals TH[1:4]m and TV[1:4]m are output according to the table below.

Table 46.12 Output signals in a half-bridge circuit configuration

Input conditions*1					Output to bridge circuit				Output to coil	
Horizontal coil	**TH1m**	**TH2m**	**TH3m**	**TH4m**	**ISMTHOUT1m**	**ISMTHEN1m**	**ISMTHOUT2m**	**ISMTHEN2m**	**COSPm**	**COSMm**
Vertical coil	**TV1m**	**TV2m**	**TV3m**	**TV4m**	**ISMTVOUT1m**	**ISMTVEN1m**	**ISMTVOUT2m**	**ISMTVEN2m**	**SINPm**	**SINMm**
	PWM low*2		0	1	0	Recirculation: - Off: 0 - On: 1	0	1	PWM	0
	PWM high*2		0	1	1	1	0	1		
	0	1	PWM low*2		0	1	0	Recirculation: - Off: 0 - On: 1	0	PWM
	0	1	PWM high*2		0	1	1	1		

Note 1. To prevent the bridge circuit from being damaged, neither TH1m & TH2m, TV1m & TV2m, TH3m & TH4m, nor TV3m & TV4m can be high at the same time. Quadrant encoding blocks all input conditions that are not given in this table.

Note 2. The shape of “PWM low” is (0,0), if recirculation is disabled, and (0,1), if recirculation is enabled. The shape of “PWM high” is always (1,0).

I/O signals to half-bridge circuits

The I/O signals to the half-bridge circuits are listed in the table below:

Table 46.13 Half-bridge circuits I/O signals

Signal name	I/O	Function
Half-bridge circuit configuration		
ISMTHOUT1m	O	Channel m horizontal output 1
ISMTHEN1m	O	Channel m horizontal enable 1
ISMTHOUT2m	O	Channel m horizontal output 2
ISMTHEN2m	O	Channel m horizontal enable 2
ISMTVOUT1m	O	Channel m vertical output 1
ISMTVEN1m	O	Channel m vertical enable 1
ISMTVOUT2m	O	Channel m vertical output 2
ISMTVEN2m	O	Channel m vertical enable 2

Full-bridge circuit configuration

In a full-bridge circuit configuration, the signals TH[1:4]m and TV[1:4]m are output according to the following table:

Table 46.14 Output signals in a full-bridge circuit configuration

	Input conditions*1				Output to bridge circuit				Output to coil	
Horizontal coil	TH1m	TH2m	TH3m	TH4m	ISMTTH1m	ISMTTH2m	ISMTTH3m	ISMTTH4m	COSPm	COSMm
Vertical coil	TV1m	TV2m	TV3m	TV4m	ISMTTV1m	ISMTTV2m	ISMTTV3m	ISMTTV4m	SINPm	SINMm
	PWM	Recirculation: - Off: 0 - On: PWM	0	1	PWM	Recirculation: - Off: 0 - On: $\overline{\text{PWM}}$	1	1	PWM	0
	0	1	PWM	Recirculation: - Off: 0 - On: PWM	1	1	PWM	Recirculation: - Off: 0 - On: $\overline{\text{PWM}}$	0	PWM

Note 1. To prevent the bridge circuit from being damaged, neither TH1m & TH2m, TV1m & TV2m, TH3m & TH4m, nor TV3m & TV4m can be high at the same time. Quadrant encoding blocks all input conditions that are not given in this table.

The signals TH[1:4]m and TV[1:4]m are passed through to the H-FET bridges but the level of the PWM signals is inverted to serve the P-channel type transistors that require a low-level PWM signal.

The following figure illustrates how the signals ISMTTH[1:4]m/ISMTTV[1:4]m relate to the transistors of an H-FET bridge.

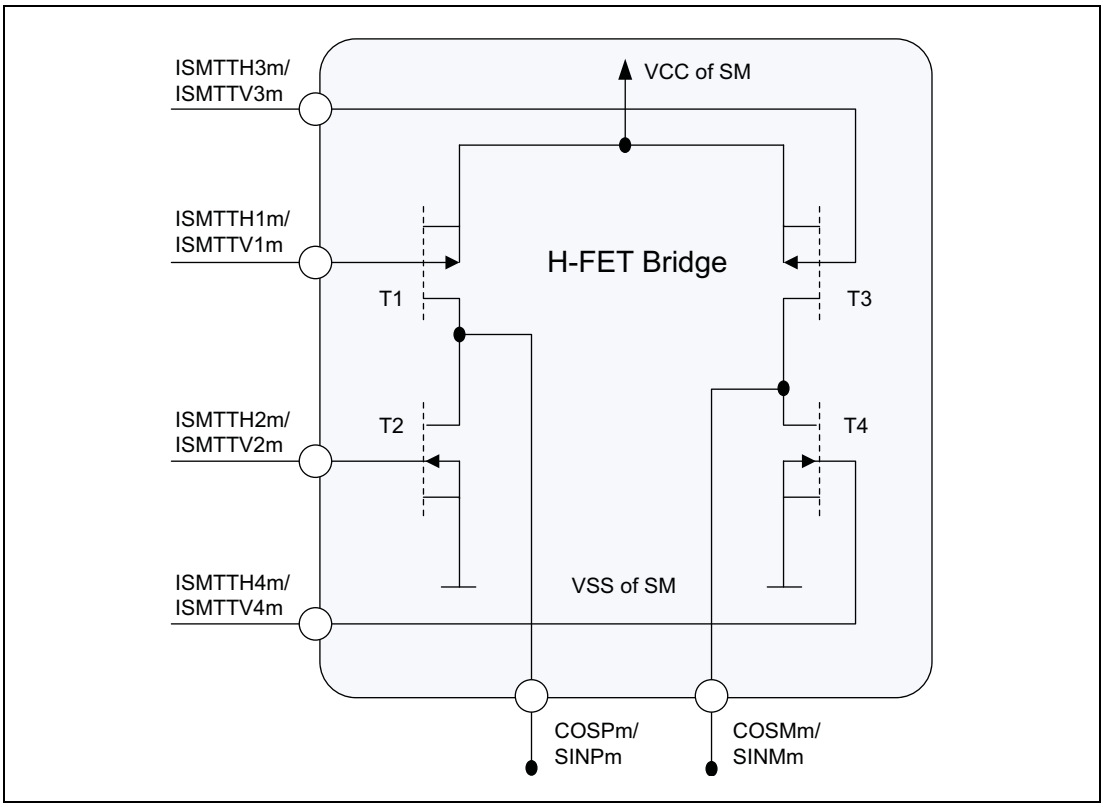


Figure 46.8 Structure of an H-FET full-bridge

I/O signals to full-bridge circuits

The I/O signals to the full-bridge circuits are listed in the table below:

Table 46.15 Full-bridge circuits I/O signals

Signal name	I/O	Function
Full-bridge circuit configuration		
ISMTTH1m	O	Channel m horizontal drive 1
ISMTTH2m	O	Channel m horizontal drive 2
ISMTTH3m	O	Channel m horizontal drive 3
ISMTTH4m	O	Channel m horizontal drive 4
ISMTTV1m	O	Channel m vertical drive 1
ISMTTV2m	O	Channel m vertical drive 2
ISMTTV3m	O	Channel m vertical drive 3
ISMTTV4m	O	Channel m vertical drive 4

46.4.6 Stepper motor driving

The Intelligent Stepper Motor Driver can drive up to 6 independent stepper motors by generating PWM signals.

A motor channel is operated in motor driving operation mode, when ISMnCCMRm.ISMnCCZPm = 0.

A basic description of the generation and processing of PWM signals can be found in **Section 46.4.5, Generation and processing of PWM signals to drive a motor**.

The following sections provide the details that are essential for using ISMn for motor driving with and without channel management.

46.4.6.1 Stepper motor driving with channel management

The following describes the details for motor driving *with* channel management.

The basic procedure is described in Section 46.4.12.2, Basic procedures: Motor driving with channel management.

(1) Provision of PWM signal information

With channel management, only the desired target position of the motor needs to be supplied to ISMn after the motor channel's parameters have been configured.

ISMn automatically moves the motor step by step until it has reached the desired target position (ISMnPAR0CFGm):

- The required intermediate positions, speed, and direction of rotation are calculated using the configured movement parameters ISMnPAR[9:1]CFGm (e.g., damping factor, acceleration limit, speed limit, etc.) and the saved variable parameters ISMnVAR[6:1]CFGm (e.g. actual position and speed).
- The PWM signal information of the calculated positions (such as pulse width, quadrant information, and recirculation enable flag) is taken from the assigned PWM value look-up table (ISMnCCMRm.ISMnCCCTm[2:0]) and fed into the corresponding registers (see Section 46.4.8.1, PWM value look-up tables for motor driving).
- The output delay for the horizontal and vertical PWM signal can be specified in

ISMnCCMRm.ISMnCCDHm[3:0] and ISMnCCMRm.ISMnCCDVm[3:0].

- ISMn can issue the interrupt request IRQ_REACHED, when a motor has reached its target position.

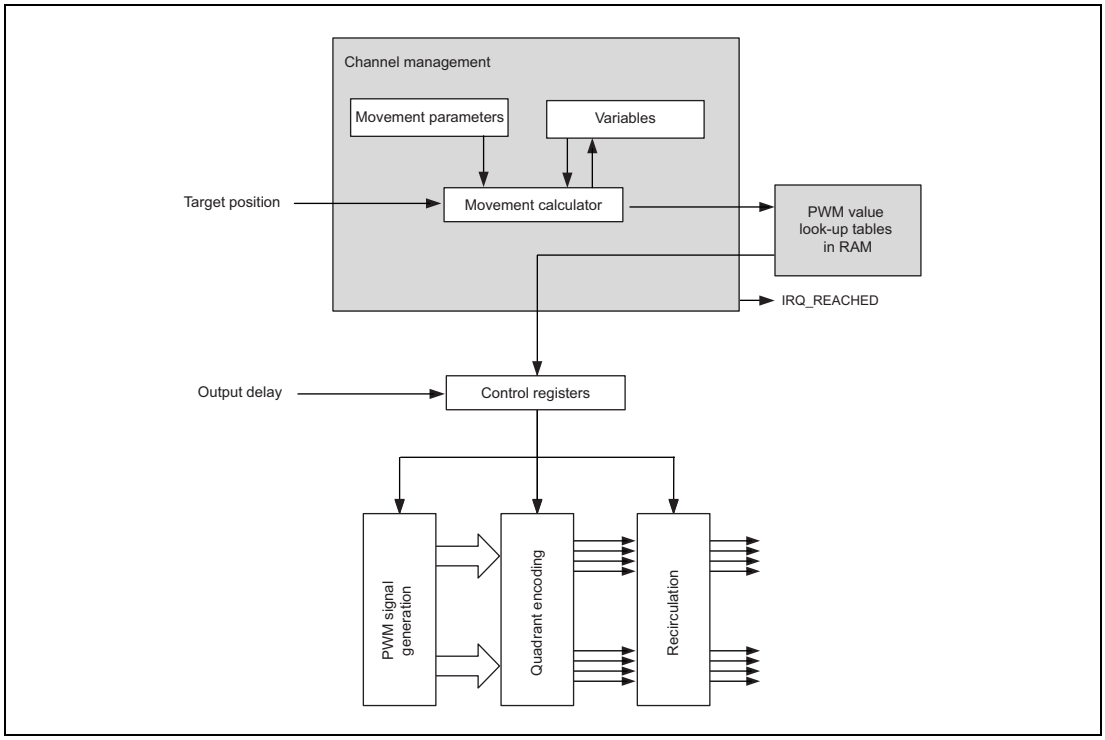


Figure 46.9 Provision of PWM signal information by channel management

(2) Modification of motor channel data

- The target position ISMnPAR0CFGm can be modified at any time.

  The new value takes effect after the next START event which is issued in a configurable interval (see Section 46.4.4.2, Channel data update interval).

- The modification of the movement parameters ISMnPAR[9:1]CFGm or the variable parameters ISMnVAR[6:1]CFGm requires the usage of channel management to be disabled for the dedicated channel (ISMnGSTR.ISMnCCENm = 0) or channel management to be switched off (ISMnGCTL.ISMnGGCE = 0).
  The new values take effect when channel management is enabled/switched on again (after the next START event).

- The modification of all other settings (including the PWM value look-up tables in the RAM) require channel management to be switched off (ISMnGCTL.ISMnGGCE = 0).
  The new values take effect when channel management is switched on again (after the next START event).

(3) Algorithm of the movement calculator

The movement calculator uses the given parameter sets and variables to calculate positions of the ISM channels. Each calculated position is an intermediate step of the movement from the current position towards the specified target position of ISMnPAR0CFGm.

Within one movement step of each channel, the movement calculator performs the calculation of the next following intermediate position, by using the algorithm below. Within this algorithm, the values are having the correspondences of the associated registers, as specified in **Table 46.32, ISMnPARiCFGm parameter registers overview**.

VPT:	ISMnVAR3CFGm	Temporary calculator register for PT1
PMP:	ISMnPAR0CFGm	Target motor position CHm
PDF:	ISMnPAR1CFGm	Damping factor CHm
VAS:	ISMnVAR1CFGm	Actual speed CHm
VAP:	ISMnVAR4CFGm	Actual position, current result CHm
VAX:	ISMnVAR0CFGm	Actual acceleration/deceleration CHm
VPS:	ISMnVAR2CFGm	Previous iteration speed CHm

```
/* Calculation of new acceleration and speed */

VPT       =    VPT + ( ( PMP - VPT) >> PDF )
VAS       =    ( VPT - VAP ) >> PDF
VAX       =    VAS - VPS

/* Limitation of the acceleration and speed */

if ( VPS > 0 )
{
     if ( VAX > PAL )
     {
          VAS = VPS + PAL
     }

     if ( VAX < ( - PDL ) )
     {
          VAS = VPS - PDL
     }
}
else
{
     if ( VAX < ( - PAL ) )
     {
          VAS = VPS - PAL
     }

     if ( VAX > PDL )
     {
          VAS = VPS + PDL
     }
}

if ( VAS >= 0 )
{
     if ( VAS > PMS )
    {
          VAS = PMS
     }
}
else
{
     if ( ( -VAS ) > PMS )
    {
          VAS = ( - PMS )
     }
}

/* Result output for Microstep & store old speed */
```

```
VAP       =    VAP + VAS
VPS       =    VAS

/* Calculate SW displayed position */

if ( ( VAP - VVP ) > PHC )
{
     VVP = VAP - PHC
}

if ( ( VAP - VVP ) < ( - PHC ) )
{
     VVP = VAP + PHC
}

/* Select Table according to speed and direction */

if ( VAS > 0 )
{
     VDR = 0

     if ( ( VAS <= PS1 ) or ( VAS >= PS2 ) )
     {
          VSP = 0
     }

     else
     {
          VSP = VSP
     }

     if( ( VAS >= PS3 ) and( VAS <= PS4 ) )
     {
          VSP = 1
     }

     else
     {
          VSP = VSP
     }
}
else
{
     if ( VAS < 0 )
     {

          VDR = 1

          if ( ( ( -VAS ) <= PS1 ) or ( ( -VAS ) >= PS2 ) )
          {
               VSP = 0
          }

          else
          {
               VSP = VSP
          }

          if ( ( ( -VAS ) >= PS3 ) and( ( -VAS ) <= PS4 ) )
          {
               VSP = 1
          }

          else
          {
               VSP = VSP
          }
     }
}
```

```
if ( VAS == 0 )
{
     VSP = 0
}

/* Negative Position suppression */

if ( VAP < 0 )
{
     VAP = 0
}
```

(4) Interrupt requests and status flags

IRQ_REACHED

If configured (ISMnGCTL.ISMnGIEREm = 1), ISMn generates the interrupt request IRQ_REACHED, when the rotor has reached its target position.

IRQ_REACHED is a general interrupt and is generated when any motor with ISMnGCTL.ISMnGIEREm = 1 has reached its target or zero position.

The status flags in ISMnGIP allow to verify the interrupt status at any time as well as which motor has caused the interrupt.

Refer to Section 46.4.10.1, IRQ_REACHED: Target/zero position reached for details.

IRQ_DONE

If configured (ISMnGCTL.ISMnGIEDO = 1), ISMn generates the interrupt request IRQ_DONE, when the sequencer has returned to the idle state (until the next START event) after all channels' data has been updated.

Refer to Section 46.4.10.2, IRQ_DONE: Sequencer idle interrupt for details.

46.4.6.2 Stepper motor driving without channel management

The following describes the details for motor driving *without* channel management are described.

The basic procedure is described in Section 46.4.12.3, Basic procedures: Motor driving without channel management.

(1) Provision of PWM signal information

Without channel management, the application needs to supply the following information for every step of the rotor:

- PWM signal information in ISMnCCMPm including the pulse width for the horizontal and vertical PWM signal and the quadrant information.
- Specification whether recirculation should be used for the horizontal and/or vertical PWM signals (ISMnCIOCm.ISMnCIHRm and ISMnCIOCm.ISMnCIVRm).

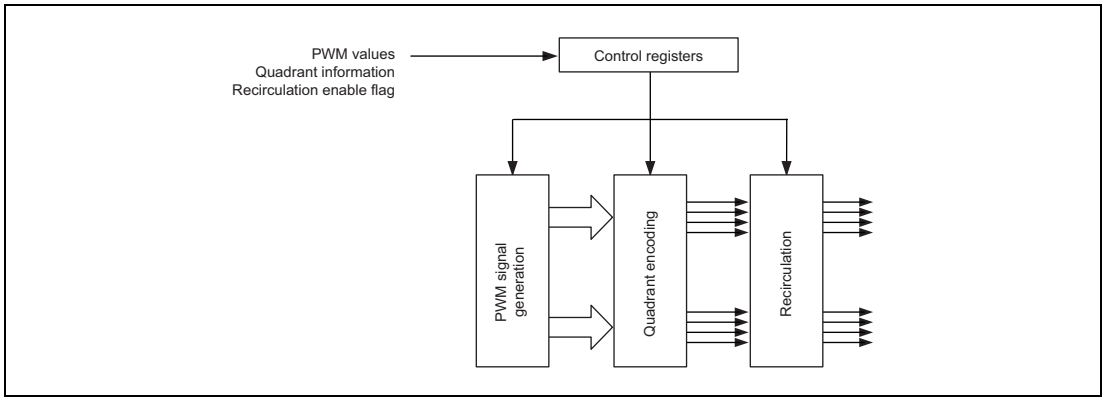


Figure 46.10 Provision of PWM signal information by the application

(2) Modification of motor channel data

- The PWM signal information can be modified at any time. The new values take effect at the beginning of the next PWM cycle.
- The output delays and the recirculation side (ISMnCOPT) can only be modified when channel management is switched off (ISMnGCTL.ISMnGGCE = 0).

46.4.7 Zero Point Detection (ZPD)

For zero point detection, the motor is driven while the voltage is measured on one selectable end of an un-driven coil at specified times.

With ISMn, up to 6 stepper motors can perform zero point detection at the same time. Furthermore, mixed operation of ZPD with regular motor driving is possible.

ZPD measurements require the zero point detection unit to be enabled (ISMnGZPDCTL.ISMnGGZE = 1). A channel is operated in ZPD operation mode, when ISMnCCMRm.ISMnCCZPm = 1.

46.4.7.1 Motor driving

A motor where ZPD operation mode is set can be driven by PWM signals (as in motor driving operation mode) and directly by using direct control.

(1) Motor driving by PWM signals

In ZPD operation mode, a motor can be driven by PWM signals as in motor driving operation mode. The PWM signal information must be applied by the application for every step of the motor.

A basic description of the generation and processing of PWM signals for motor driving can be found in Section 46.4.5, Generation and processing of PWM signals to drive a motor.

(2) Direct control

In ZPD operation mode, direct control can be used to directly define how the horizontal and vertical coils are applied with current.

Direct control can be enabled/disabled for the horizontal and vertical coils individually. When direct control is enabled for a coil, the PWM signal information for that coil is ignored (including recirculation settings). Instead the level of the signals TH[1:4]m and TV[1:4]m is set as defined in ISMnCIOCm.

- When ISMnCIOCm.ISMnCIHEm = 1, bits ISMnCIHDm[3:0] specify how the *horizontal* coil is to be activated.
- When ISMnCIOCm.ISMnCIVEm = 1, bits ISMnCIVDm[3:0] specify how the *vertical* coil is to be activated.

Table 46.16 Signal levels depending on direct control settings

ISMnCIOCm.ISMnCIHDm[3:0]	TH1m	TH2m	TH3m	TH4m
ISMnCIOCm.ISMnCIVDm[3:0]	TV1m	TV2m	TV3m	TV4m
0000_B	0	0	0	0
0001_B	0	0	0	1
0010_B	0	0	1	0
0100_B	0	1	0	0
0101_B	0	1	0	1
0110_B	0	1	1	0
1000_B	1	0	0	0
1001_B	1	0	0	1
1010_B	1	0	1	0

NOTE

To prevent the bridge circuit from being damaged, neither TH1m & TH2m, TV1m & TV2m, TH3m & TH4m nor TV3m & TV4m can be high at the same time. Therefore, direct control blocks all input conditions that are not given in this table: dangerous conditions are avoided in these cases by setting some of TH[1:4]m/TV[1:4]m to low.

When channel management is set, ISMnCIOCm.ISMnCIVEm, ISMnCIOCm.ISMnCIHEm, ISMn CIVDm[3:0], and ISMnCIHDm[3:0] are taken from the assigned ZPD table. Otherwise, they need to be supplied by the application.

The signals TH[1:4]m and TV[1:4]m are output to the bridges according to the tables in Section 46.4.5.6, Signal output.

46.4.7.2 Input selection for ZPD measurement

For zero point detection, one pin of a motor channel is used as input pin to measure the induced voltage. Both ends of the horizontal and vertical coils can be measured (plus side and minus side).

The valid input signal is specified in ISMnZPDCTL.ISMnCZISm[1:0].

When channel management is set, ISMnZPDCTL.ISMnCZISm[1:0] is taken from the assigned ZPD table. Otherwise, ISMnZPDCTL.ISMnCZISm[1:0] needs to be supplied by the application.

46.4.7.3 Set of measurement sequences

The motor channels are considered for ZPD measurement where the following conditions are true:

- Motor channel is operated in ZPD operation mode (ISMnCCMRm.ISMnCCZPm = 1)
- ZPD measurement trigger is set (ISMnZPDCTL.ISMnCZMTm = 1)

The input signals of all channels to be considered are measured concurrently.

For each channel, one measurement sequence is repeated endlessly until the ZPD measurement trigger ISMnZPDCTL.ISMnCZMTm is reset to 0.

When channel management is set, ISMnZPDCTL.ISMnCZMTm is taken from the assigned ZPD table. Otherwise, ISMnZPDCTL.ISMnCZMTm needs to be supplied by the application.

The following chapters describe how ISMn allows a very precise timing of ZPD measurements. To simplify matters a set of measurement sequences for one motor channel is described.

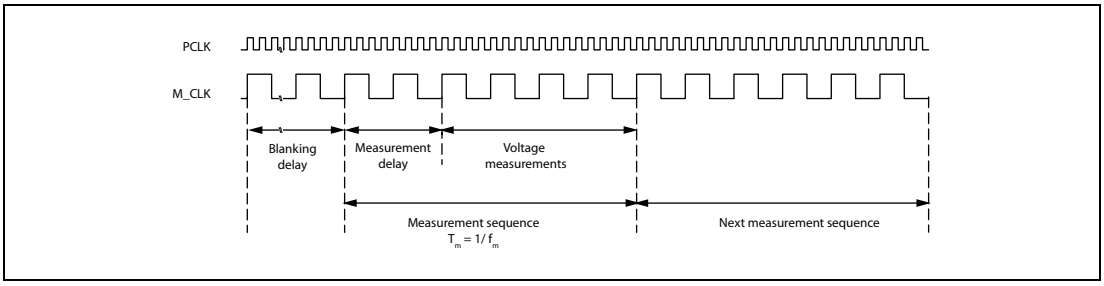


Figure 46.11 Set of measurement sequences

(1) ZPD measurement clock M_CLK

ZPD measurements are synchronized with the ZPD measurement clock M_CLK. M_CLK is a divided clock of PCLK and can therefore be used to slow down ZPD measurements.

$$M\_CLK = PCLK / N$$

The division factor N can be configured in ISMnGZPDCTL.ISMnGGZF[3:0].

(2) ZPD measurement frequency

The measurement frequency can be calculated using the following formula:

$$f_M = M\_CLK / (O + P)$$

O = Measurement delay specified in ISMnGZPDCTL.ISMnGGCS[2:0]

P = Number of times, the voltage is measured during one measurement sequence as specified in ISMnGZPDCTL.ISMnGGFD[3:0]

NOTE

ISMnGZPDCTL can only be modified when channel management is switched off (ISMnGCTL.ISMnGGCE = 0).

(3) Blanking delay

When the ZPD measurement is triggered by writing 1 to ISMnZPDCTL.ISMnCZMTm the first measurement sequence starts after the configured blanking delay (ISMnZPDCMPm.ISMnCZBTm[15:0]) which is defined as a multiple of measurement sequences T_m.

NOTE

ISMn can generate the interrupt request IRQ_ZPDAD at the beginning of the first measurement sequence.

(4) Measurement delay

A measurement sequence comprises a measurement delay and a period of time of actual voltage measurements.

- The measurement delay is defined in ISMnGZPDCTL.ISMnGGCS[2:0] as a number of clock cycles of M_CLK.
- The period of time for actual voltage measurements depends on the noise filter settings (see Section (6), Noise filtering).

ISMnGZPDCTL.ISMnGGCS[2:0] needs to be supplied by the application.

(5) Reference voltages

ISMn provides 3 different reference voltages:

- 2 internal reference voltage
- 1 external reference voltage is generated out of the voltage supplied by the ZPD external reference voltage port pin (ZPDVREF):

to be used by every channel: ISMnGZPDCTL.ISMnGGRV1[3:0] and ISMnGZPDCTL.ISMnGGRV2[3:0]. Possible reference voltages are between 100 mV and 850 mV.

The voltages measured for a channel are compared to the reference voltage selected for that channel in ISMnZPDOPT.ISMnCZRSm.

(6) Noise filtering

For reliable results, voltage measurements of a short pulse and spikes can be filtered out using digital noise filtering.

In the course of one measurement sequence, the voltage is measured during ISMnGZPDCTL.ISMnGGFD[3:0] clock cycles of M_CLK. When the voltage is above the channel's reference voltage for greater or equal ISMnGZPDCTL.ISMnGGFL[3:0] clock cycles of M_CLK, the motor is assumed to still not having reached its zero position. Subsequent measurement sequences can be appended to define the measurement time window.

Method

The voltage of the selected input channel is measured a number of consecutive times which is defined in ISMnGZPDCTL.ISMnGGFD[3:0].

Every voltage measured is compared against the channel's reference voltage:

- When a configured number of singular measurement results (ISMnGZPDCTL.ISMnGGFL[3:0]) is *above* the channel's reference voltage, ISMnZPDSTR.ISMnCZDRm is set to 1 (no zero point is detected).

- When less measurement results than configured are above the channel's reference voltage ISMnZPDSTR.ISMnCZDRm is set to 0 accordingly (zero point detection is yet unconfirmed).

(7) Noise filtering result

ISMnZPDSTR.ISMnCZDRm indicates whether – according to noise filtering – the level of the input signal is above or below the level of the reference voltage.

When ISMnZPDSTR.ISMnCZDRm = 1

The input signal level is *above* the reference voltage when ISMnZPDSTR.ISMnCZDRm = 1:

- The corresponding ZPD detection flag ISMnZPDIP.ISMnCZIPm is set to 1.
- The interrupt request IRQ_ZPD is generated only once per channel since ISMnZPDIP.ISMnCZIPm has last been cleared.

NOTES

1. IRQ_ZPD is a general interrupt that is issued for any motor channel that is operated in ZPD operation mode where the motor's zero position has not been reached yet. ISMnZPDSTR.ISMnCZDRm can be used to verify which channels motors did not reach their zero position so that appropriate steps can be taken.
2. ISMn automatically clears the ZPD detection flags (ISMnZPDSTR.ISMnCZDRm and ISMnZPDIP.ISMnCZIPm) when a motor channel is switched from motor driving to ZPD operation mode.
3. The application can clear the ZPD detection flag by writing 1 to ISMnZPDSTC.ISMnCCLZPm. For motor channels that do not use channel management, this should be done in regular intervals.

When ISMnZPDSTR.ISMnCZDRm = 0

The input signal level is *below* the reference voltage when ISMnZPDSTR.ISMnCZDRm = 0.

ISMnZPDIP.ISMnCZIPm is not altered thus indicating the ZPD measurement status since ISMnZPDIP.ISMnCZIPm has last been cleared.

(8) Example of set of measurement sequences

The following example illustrates one set of measurement sequences for motor channel m.

1. The ZPD measurement is triggered by writing 1 to ISMnZPDCTL.ISMnCZMTm.
2. The first measurement sequence starts after the blanking delay has passed. The blanking is a multiple of the measurement sequence T_m, defined by ISMnZPDCMPm.ISMnCZBTm[15:0].
3. The first actual voltage measurement is delayed by the number of clock cycles of M_CLK specified in ISMnGZPDCTL.ISMnGGCS[2:0] which is 01_H in this example.
4. The voltage level of the selected input signal then is measured 4 consecutive times (ISMnGZPDCTL.ISMnGGFD[3:0] = 3_H).

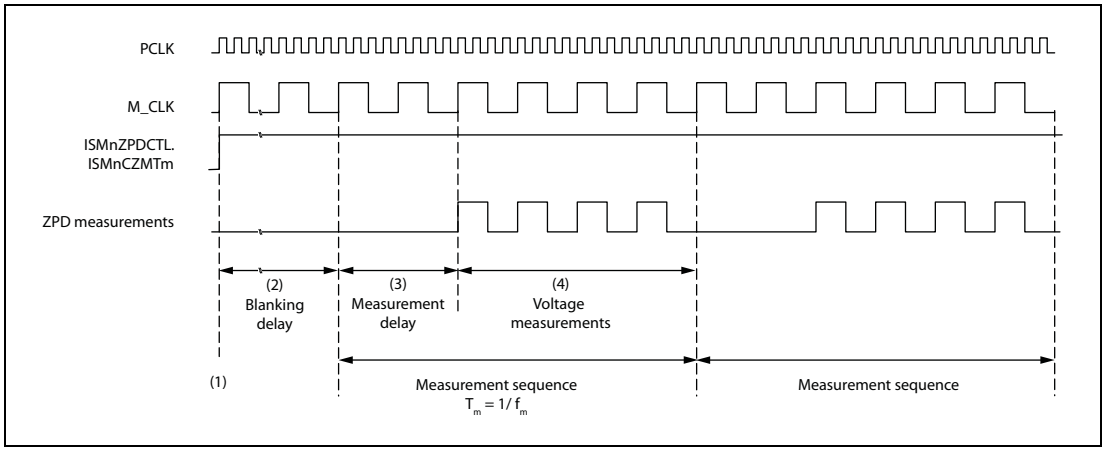


Figure 46.12 Example of set of measurement sequences

The measurement sequence is repeated until ISMnZPDCTL.ISMnCZMTm is reset to zero.

(9) Example for digital noise filtering

The following example illustrates digital noise filtering where the following settings are made:

- During one measurement sequence, 4 voltage measurements are performed (ISMnGZPDCTL.ISMnGGFD[3:0] = 3_H).
- If 3 or more measurements are *above* the reference voltage, ISMnZPDSTR.ISMnCZDRm is set to 1.
  If less than 3 measurements are *below* the reference voltage, ISMnZPDSTR.ISMnCZDRm is set to 0.
  (ISMnGZPDCTL.ISMnGGFL[3:0] = 2_H)

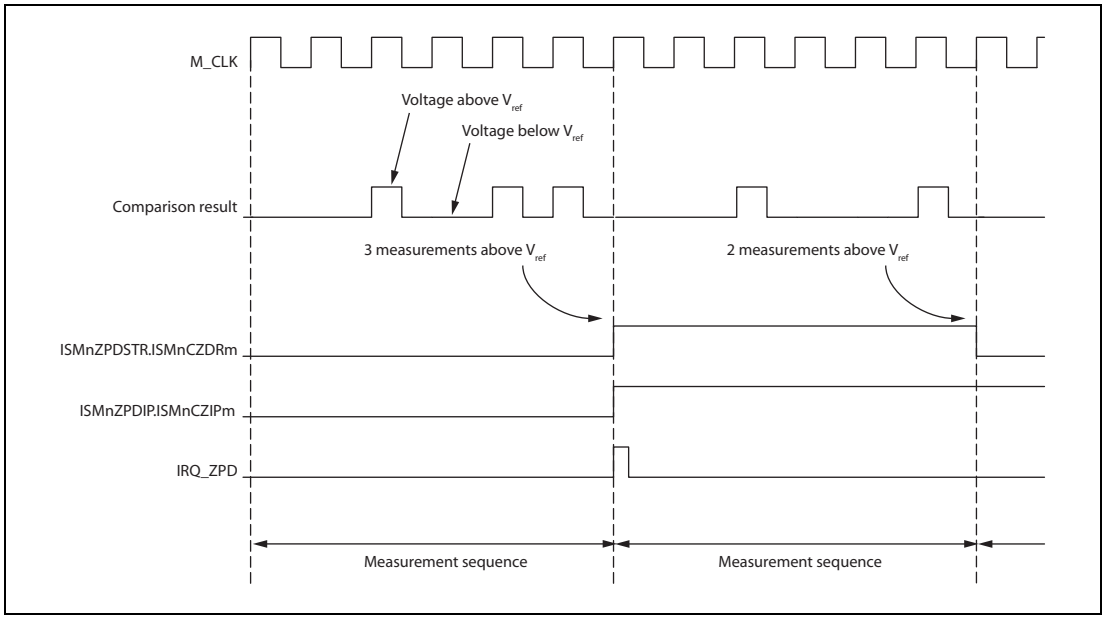


Figure 46.13 Example of noise filtering

During the first measurement sequence, 3 measurements exceed the reference voltage. This is more than the specified threshold of 2 measurements. Therefore, ISMnZPDSTR.ISMnCZDRm is set to 1.

When ISMnZPDSTR.ISMnCZDRm is set for the first time:

- The interrupt request IRQ_ZPD is generated.
- ISMnZPDIP.ISMnCZIPm is set and remains set until it is cleared (either by the application or by ISMn when the measurement start trigger ISMnZPDCTL.ISMnCZMTm changes from 0 to 1 to start a new set of measurement sequences).

During the second measurement sequence, 2 measurements exceed the reference voltage. This is less than the specified threshold. Therefore, ISMnZPDSTR.ISMnCZDRm is reset.

(10) ZPD vibration damping

If during a measurement sequence, it is detected that the motor has not reached its zero position yet (voltage has been above the reference voltage for ISMnGZPDCTL.ISMnGGFL[3:0] clock cycles of M_CLK), ISMn can short-circuit the coil which attenuates motor vibrations. The coils are shortcircuited until the ZPD measurement trigger is reset (ISMnZPDCTL.ISMnCZMTm = 0).

ISMnZPDCMPm.ISMnCZSSm can be used to enable/disable the ZPD vibration damping feature.

46.4.7.4 ZPD with channel management

The following describes the details for zero point detection *with* channel management.

The basic procedure is described in Section 46.4.12.4, Basic procedures: ZPD with channel management.

Configuration of ZPD unit

No matter whether or not channel management is used for ZPD, the ZPD unit needs to be set up in ISMnGZPDCTL. These general settings include the ZPD measurement clock M_CLK, the level of the reference voltages, the digital noise filter settings, the usage of the A/D converter etc. Most of them are described in Section 46.4.7.3, Set of measurement sequences.

NOTE

To perform ZPD measurements, the ZPD unit must be enabled (bit ISMnGGZE = 1) and power saving mode must be disabled (bit ISMnGGZP = 0).

(1) Processing of ZPD table

ZPD tables that are stored in the RAM contain a control sequence that shall bring the motors to their zero position (see Section 46.4.8.2, ZPD tables in RAM).

ISMnCCMRm.ISMnCCCTm[2:0] specifies the ZPD table to be used for channel m. The assigned ZPD table is processed row by row in recurring loops.

When a channel's operation mode changes from motor driving to ZPD operation mode, the ZPD table index counter (ISMnCCNT.ISMnCCZCm[6:0]) is reset and the values of the first row are fed into the corresponding registers. The values are applied at the beginning of the next PWM cycle.

From now on, every time channel management updates the channels' data (indicated by START events), ISMn checks whether the next row has to be fetched from the table or not:

- When the channel data update delay ISMnCZCFGm.ISMnCCZDm[4:0] > 0, the delay is

decreased by 1 and the old values stay effective.

This way, ISMnCZCFGm.ISMnCCZDm[4:0] allows the values of one row to be applied for multiple channel data update times, for example to expand ZPD measurement over a longer period of time.

- When ISMnCZCFGm.ISMnCCZDm[4:0] = 0, the table index counter is increased by 1 and the values of the next row are fed into the corresponding registers.
- When the last row to be considered (ISMnCCMRm.ISMnCCTLm[6:0]) has been processed, the ZPD table index counter is reset and the processing of the ZPD table restarts from the beginning.

The ZPD measurement procedure stops when any of the following conditions are true:

- ZPD measurement indicates that motor has reached its zero position
- Channel management is switched off
- The usage of channel management is disabled for that motor channel
- Operation mode is reset to motor driving operation mode

Example

In the following example, the table limit ISMnCCMRm.ISMnCCTLm[6:0] is set to 2 (3 rows). In the second table row of the ZPD table (table row number 1), a channel data update delay of 4 channel update times is configured (ISMnCZCFGm.ISMnCCZDm[4:0] = 3). The settings of this table row are applied at 4 consecutive START events.

Note that the table index counter ISMnCCNT.ISMnCCZCm[6:0] does not show the row number of the current table row but the number of the *next* row.

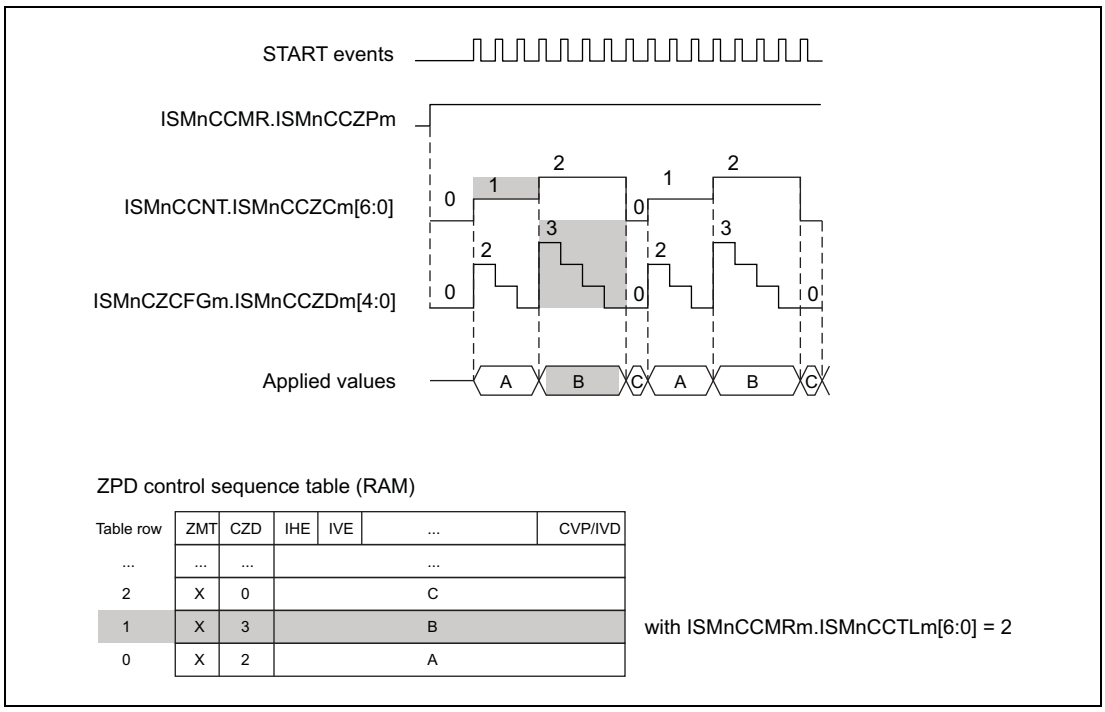


Figure 46.14 Processing of ZPD table

(2) Result evaluation when ZPD measurement is switched off

When the ZPD measurement trigger is reset (ISMnZPDCTL.ISMnCZMTm = 0), ISMn checks ISMnZPDIP.ISMnCZIPm to check whether or not the motor has reached its zero position during the last set of measurement sequences.

Motor has not reached zero position

ISMnZPDIP.ISMnCZIPm = 1 indicates that during at least one measurement sequence, the measured voltages were above the channel's reference voltage. Therefore the motor is assumed *not* to be at its zero position.

- The ZPD detection flag ISMnZPDIP.ISMnCZIPm is cleared.
- ISMn proceeds the ZPD measurement control sequence as defined by the assigned ZPD table.

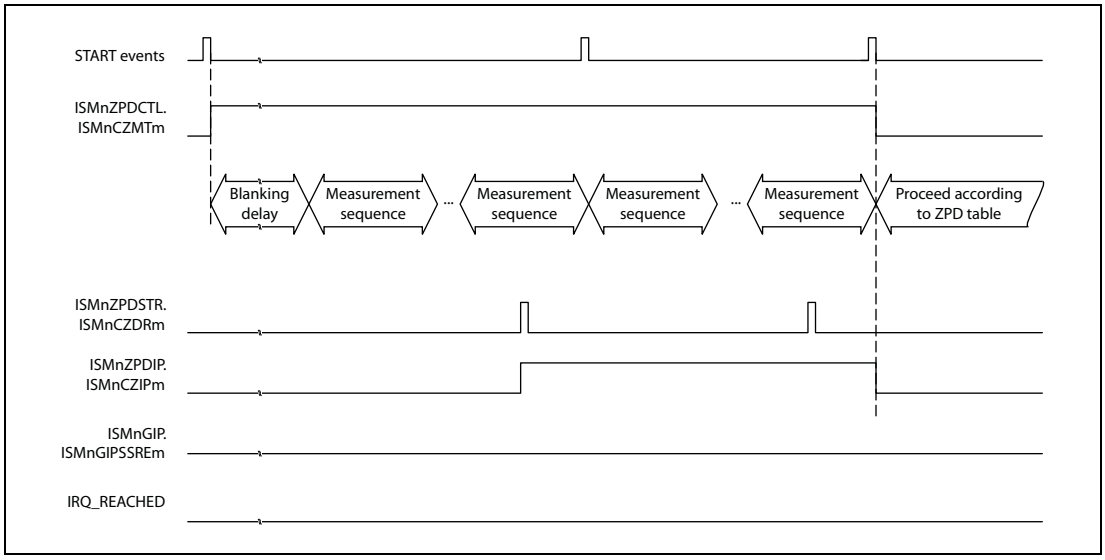


Figure 46.15 No zero point detected

Motor has reached zero position

ISMnZPDIP.ISMnCZIPm = 0 indicates that – according to digital noise filtering – the measured voltages were never above the channel's reference voltages. Therefore the motor is assumed to be at its zero position.

- The interrupt status flag of the respective channel ISMnGIP.ISMnGIPSSREm is set to 1.
- When ISMnGCTL.ISMnGIEREm = 1, ISMn additionally generates the interrupt request IRQ_REACHED as in motor driving mode.
  Refer to Section 46.4.10.1, IRQ_REACHED: Target/zero position reached for details.
- Channel management also resets the motor channel to motor driving operation mode so that the respective channel is no longer considered for ZPD.
- Also, the usage of channel management is disabled for that channel.
- ZPD table index counter ISMnCCNTm.ISMnCCZCm[6:0] and channel data update delay ISMnCZCFGm.ISMnCCZDm[4:0] are reset to zero.

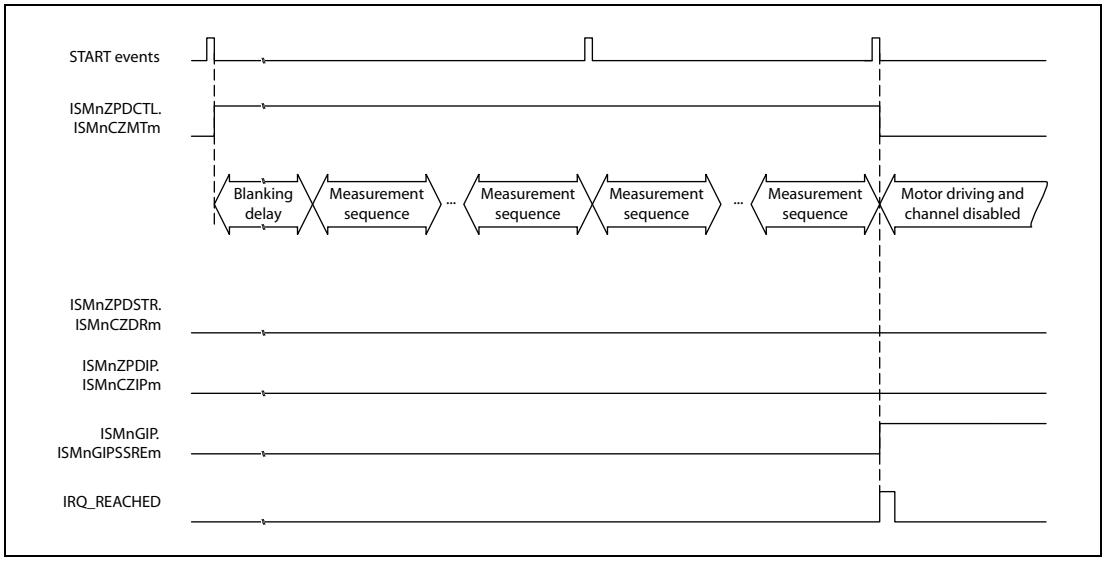


Figure 46.16 Zero position detected

46.4.7.5 ZPD without channel management

The following describes the details for zero point detection *without* channel management.

The basic procedure is described in Section 46.4.12.5, Basic procedures: ZPD without channel management.

Configuration of ZPD unit

No matter whether or not channel management is used for ZPD, the ZPD unit needs to be set up in ISMnGZPDCTL. These general settings include the ZPD measurement clock M_CLK, the level of the reference voltages, the digital noise filter settings, etc. Most of them are described in Section 46.4.7.3, Set of measurement sequences.

NOTE

To perform ZPD measurements, the ZPD unit must be enabled (bit ISMnGGZE = 1) and power saving mode must be disabled (bit ISMnGGZP = 0).

(1) Provision of channel data for motor driving

Without channel management, the application needs to supply the following information for every step of the rotor:

Generation of PWM signals

For motor driving by PWM signals:

- PWM signal information in ISMnCCMPm including the pulse width for the horizontal and vertical PWM signal and the quadrant information.
- Output delay for the horizontal and vertical PWM signal in ISMnCCMRm.ISMnCCDHm[3:0] and ISMnCCMRm.ISMnCCDVm[3:0].
- Specification whether recirculation should be used for the horizontal and/or vertical PWM signals (ISMnCIOCm.ISMnCIHRm and ISMnCIOCm.ISMnCIVRm).

Direct control

For motor driving by direct control:

- Specification whether direct control is used for the horizontal and/or vertical coil (ISMnCIOCm.ISMnCIHEm and/or ISMnCIOCm.ISMnCIVEm)
- Specification how the horizontaland/or vertical coil is to be activated (ISMnCIOCm.ISMn CIHDm[3:0] and/or ISMnCIOCm.ISMn CIVDm[3:0])

NOTE

New values take effect at the beginning of the next PWM cycle. When direct control is used to activate a coil, the registers that specify the PWM signal information for that coil are ignored.

(2) Provision of channel data for ZPD measurements

Without channel management, the application needs to supply the following information to perform ZPD measurements:

- Blanking time in ISMnZPDCMPm.ISMnCZBTm[15:0]
- Reference voltage to be used in ISMnZPDOPT.ISMnCZRSm
- Specification whether the ZPD vibration damping feature is used (ISMnZPDCMPm.ISMnCZSSm).
- Valid input signal in ISMnZPDCTL.ISMnCZISm[1:0]

NOTE

When the start trigger ISMnZPDCTL.ISMnCZMTm is set to 1, the values take effect at the beginning of the next PWM cycle.

(3) Evaluation of ZPD measurements

The measurement sequence is repeated until the ZPD measurement trigger ISMnZPDCTL.ISMnCZMTm is reset to 0.

ISMnZPDSTR.ISMnCZDRm stores the result of the last measurement sequence. It indicates whether – according to noise filtering – the level of the input signal is above or below the level of the reference voltage.

For a higher certainty whether or not the motor has reached its zero position, further measurement sequences can be evaluated.

The ZPD detection flag ISMnZPDIP.ISMnCZIPm stores the result of all measurement sequences since the detection flag has last been cleared. Therefore, it is recommended to verify ISMnZPDIP.ISMnCZIPm in regular intervals. For a certain result, the interval must allow enough measurement sequences to have passed.

- When ISMnZPDIP.ISMnCZIPm = 0, the measured voltage was not above the reference voltage during the last interval. When the zero position is assumed to be reached, ZPD measurements can be stopped by resetting the ZPD measurement trigger ISMnZPDCTL.ISMnCZMTm.
- When ISMnZPDIP.ISMnCZIPm = 1, at least one measured voltage was above the reference voltage during the last interval.

In this case, the motor has not reached its zero position and further steps must be taken.
To prepare for the next ZPD measurement phase, ISMnZPDIP.ISMnCZIPm needs to be cleared by writing 1 to ISMnZPDSTC.ISMnCCLZPm.

NOTE

The interrupt request IRQ_ZPD indicates the first time the measured voltage is above the reference voltage (see Section 46.4.10.4, IRQ_ZPD: Zero point not detected).

(4) Modification of motor channel data

- The direct control settings, the PWM signal information, and the input signal to be measured can be modified at any time. The new values take effect at the beginning of the next PWM cycle.
- The ZPD unit's general configuration can only be modified when the ZPD unit is disabled (ISMnGZPDCTL.ISMnGGZE = 0). This stops ZPD measurements.
- The output delays and the recirculation side (ISMnCOPT) for the PWM signals can only be modified when channel management is switched off.

46.4.7.6 Power save mode

To reduce power consumption, the comparators and reference voltage dividers can be disabled when the ZPD operation mode is not required by any motor channel.

The ZPD unit is disabled when ISMnGZPDCTL.ISMnGGZP = 1.

NOTE

When the ZPD operation mode is configured for a motor channel while the ZPD unit is disabled, ZPD measurement and motor driving operations may fail or result in unreliable outcomes.

46.4.8 PWM value look-up tables and ZPD tables in RAM

When channel management is used for motor driving or zero point detection, appropriate value tables need to be set up in the RAM.

NOTE

This has to be done after enabling/resetting the Intelligent Stepper Motor Driver but before switching on channel management.

When channel management is switched on (ISMnGCTL.ISMnGGCE = 1), the RAM tables are controlled by channel management. Access to the tables by the application is restricted: zero is read back and write accesses are ignored.

Required RAM space

The following RAM space is reserved for the tables for motor driving and zero point detection:

unused	<ISMn_base> + $7FFF_H$
	<ISMn_base> + $7C00_H$
ZPD	<ISMn_base> + $7BFF_H$
	<ISMn_base> + 7000_H
Motor driving	<ISMn_base> + $6FFF_H$
	<ISMn_base> + 4000_H

NOTE

Unused RAM space, and unused ZPD and motor driving table space can be used by the application for any other purpose.

Basic procedure

To set-up the tables in the RAM:

1. Ensure that channel management is switched off (ISMnGCTL.ISMnGGCE = 0).
2. Write the required data to the RAM according to the address and format requirements stated in the following chapters.

46.4.8.1 PWM value look-up tables for motor driving

PWM value look-up table(s) with a 32-bit PWM signal information for every position of the stepper motor need to be set up in the RAM.

- Stepper motors that should be driven with *high precision* (ISMnCCMRm.ISMnCCCPm = 1) require a PWM value look-up table that comprises PWM signal information for the 512 rotor positions possible.
- Stepper motors that should be driven with *standard precision* (ISMnCCMRm.ISMnCCCPm = 0) require the PWM signal information for 128 rotor positions – but for two different rotation directions and speeds. Therefore, a PWM value look-up table for motor driving with standard precision must comprise 4 sets of PWM signal information.

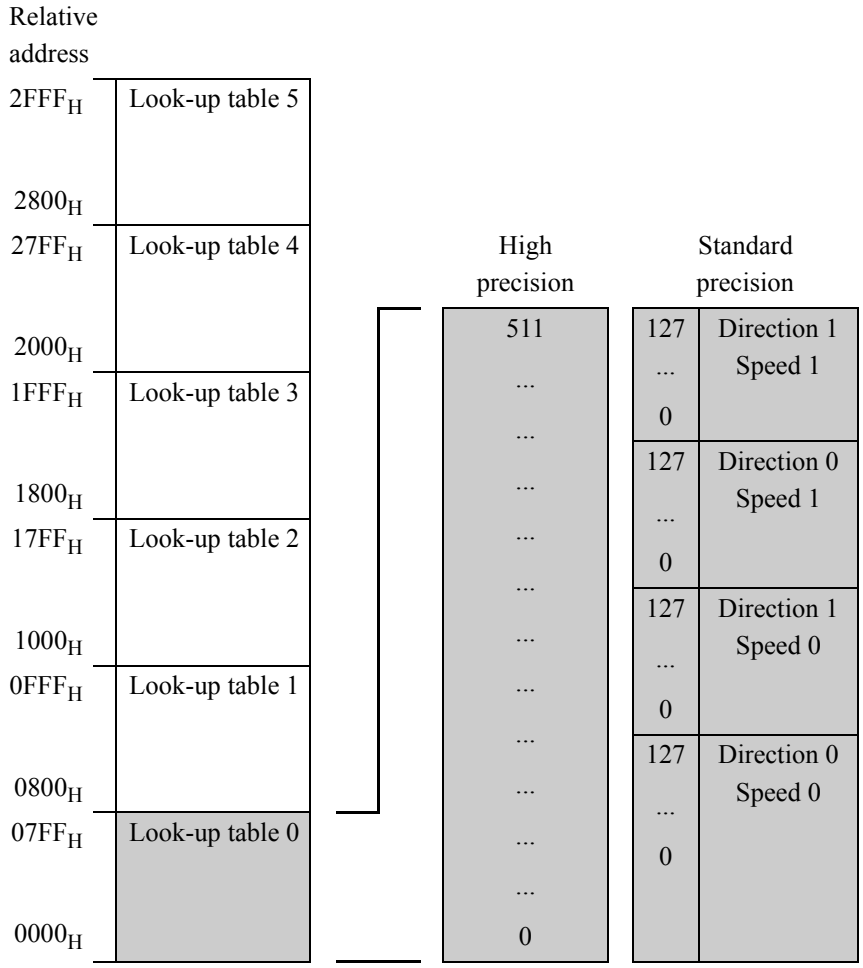
Up to 6 different PWM value look-up tables can be stored in the RAM.

ISMnCCMRm.ISMnCCCTm[2:0] specifies which look-up table is used for channel m. Multiple motors can share a look-up table provided that the same precision is configured.

RAM addresses

A 15-bit address is used to identify RAM locations:

- The base address for the PWM value look-up tables is <ISMn_base> + 4000_H.
- The relative addresses are given in the figure below.



Standard precision requires 4 sets of PWM values:

Direction:

0 = Movement to a higher position

1 = Movement to a lower position

Speed:

0 = Slow rotation

1 = Fast rotation

Storage order: bottom-up

Within a look-up table, the PWM signal information must be stored bottom-up. For example, to use look-up table 0 to drive a motor with high precision, the PWM signal information must be stored at the following relative addresses: step 0 at 0000_H and step 511 at $07FF_H$.

Table row format

For every rotor position, the PWM signal information must be stored in the following 32-bit format:

31 ... 24	23	22	21 20	19 ... 10	9 ... 0
Not used	RF-H	RF-V	CQI[1:0]	PWM-H[9:0]	PWM-V[9:0]

Table 46.17 Contents of PWM value look-up table row

Bit position	Name	Function	Applied to register
23	RF-H	Enables/disables recirculation for the *horizontal* PWM signal: 0: Recirculation disabled 1: Recirculation enabled	ISMnCIOCm. ISMnCIHRm
22	RF-V	Enables/disables recirculation for the *vertical* PWM signal: 0: Recirculation disabled 1: Recirculation enabled	ISMnCIOCm. ISMnCIVRm
21 to 20	CQI[1:0]	Specifies the quadrant: 00_B: Quadrant 1 (0° - 90° angle) 01_B: Quadrant 2 (90° - 180° angle) 10_B: Quadrant 3 (180° - 270° angle) 11_B: Quadrant 4 (270° - 360° angle)	ISMnCCMPm. ISMnCCQIm[1:0]
19 to 10	PWM-H[9:0]	Specifies the duty cycle time of the *horizontal* PWM signal as a number of cycles of TB_CLK The value range is from 0 (PWM signal switched off) to 2^{10} - 1 (duty cycle = 100%).	ISMnCCMPm. ISMnCCHPm[9:0]
9 to 0	PWM-V[9:0]	Specifies the duty cycle time of the *vertical* PWM signal	ISMnCCMPm. ISMnCCVPm[9:0]

46.4.8.2 ZPD tables in RAM

ISMn allows up to 6 different ZPD tables to be stored in the RAM. One ZPD table can be used by multiple motors.

As described in Section 46.4.7.4, ZPD with channel management, **(1) Processing of ZPD table**, ZPD tables are processed row by row in recurring loops. Each table can store up to 128 rows of PWM signals and ZPD measurement information. The end of a table is configurable in ISMnCCMRm.ISMnCCTLm[6:0] so that not all rows have to be filled.

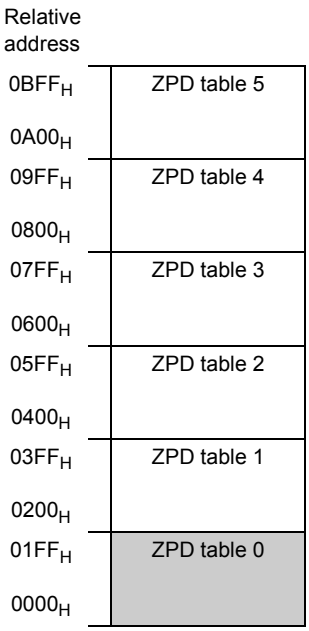
NOTE

The ZPD control sequence stored in one table must perform at least one full rotation of the motor. Otherwise it is not possible to ensure that the motor reaches its zero position from any start position.

RAM addresses

A 15-bit address is used to identify RAM locations:

- The base address for the ZPD tables is <ISMn_base> + 7000_H.
- The relative addresses are given in the figure below.



Storage order: bottom-up

Within a ZPD table, the ZPD control sequence must be stored bottom-up. For example, to use ZPD table 0, the values stored at the relative address 0000_H are used first and the values stored at the relative address $01FF_H$ are taken when the index counter (ISMnCCNTm.ISMnCCZCm[6:0]) and the table index limit (ISMnCCMRm.ISMnCCTLm[6:0]) are at their maximum value of $7F_H$.

Table row format

Each table row must be stored in the following 32-bit format. The bit allocation depends on the measurement trigger ZMT, and the direct control enable flags DC-H and DC-V.

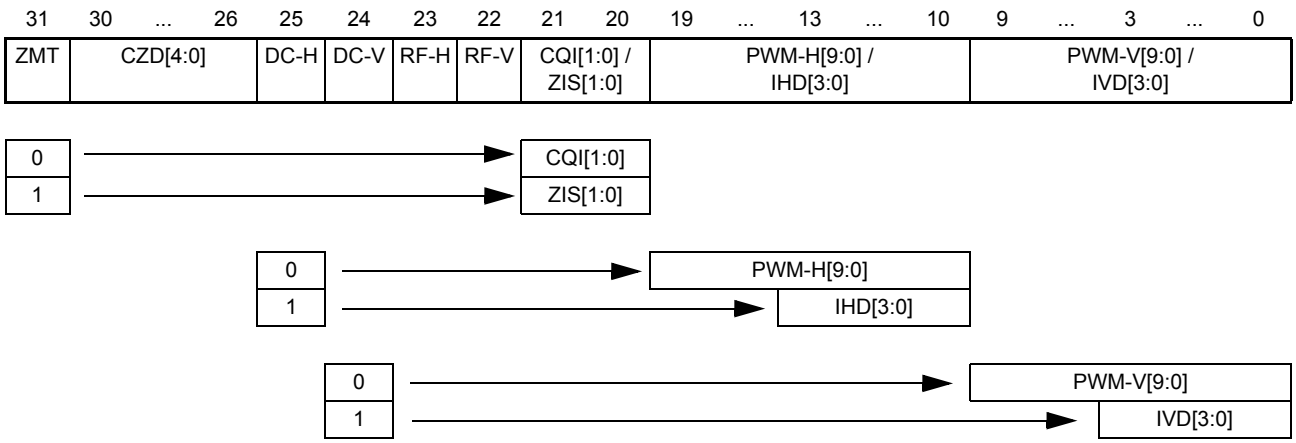


Table 46.18 Contents of ZPD value table row (1/2)

Bit position	Name	Function	Applied to register
31	ZMT	ZPD measurement trigger for channel m: 0: Do not perform ZPD measurements 1: Perform ZPD measurements	ISMnZPDCTL. ISMnCZMTm
30 to 26	CZD[4:0]	Number of START events that are waited for before the next table row is fetched (= delay). CZD therefore can be used to specify when new data is applied to the motor channel. The minimum value is 00_H and the maximum value is $1F_H$ = 31. An additional table row must be used when a longer delay is required. Refer to Section 46.4.7.4, ZPD with channel management, **(1) Processing of ZPD table** for details.	ISMnCZCFGm. ISMnCCZDm[4:0]
25	DC-H	Enables/disables direct control for the *horizontal* PWM signal: 0: Direct control disabled Bits 19 to 10 define the duty cycle time of the *horizontal* PWM signal. 1: Direct control enabled Bits 13 to 10 define which transistors will be activated to supply the *horizontal* coil with current. Refer to Section 46.4.2.1, Motor driving for details.	ISMnCIOCm. ISMnCIHEm
24	DC-V	Enables/disables direct control for the *vertical* PWM signal: 0: Direct control disabled Bits 9 to 0 define the duty cycle time of the *vertical* PWM signal. 1: Direct control enabled Bits 3 to 0 define which transistors will be activated to supply the *vertical* coil with current.	ISMnCIOCm. ISMnCIVEm
23	RF-H	Enables/disables recirculation for the *horizontal* PWM signal: 0: Recirculation disabled 1: Recirculation enabled Refer to Section 46.4.5.5, Recirculation for details.	ISMnCIOCm. ISMnCIHRm
22	RF-V	Enables/disables recirculation for the *vertical* PWM signal: 0: Recirculation disabled 1: Recirculation enabled	ISMnCIOCm. ISMnCIVRm
21 to 20	CQI[1:0]	The allocation of bits 21 and 20 depends on ZMT: ZMT = 0: Specifies the quadrant: 00_B: Quadrant 1 (0° - 90° angle) 01_B: Quadrant 2 (90° - 180° angle) 10_B: Quadrant 3 (180° - 270° angle) 11_B: Quadrant 4 (270° - 360° angle) Refer to Section 46.4.5.4, Encoding of PWM signals according to quadrant for details.	ISMnCCMPm. ISMnCCQIm[1:0]
	ZIS[1:0]	ZMT = 1: Specifies the input signal to be measured for ZPD: 00_B: SINPm 01_B: COSPm 10_B: SINMm 11_B: COSMm Refer to Section 46.4.7.2, Input selection for ZPD measurement for details.	ISMnZPDCTL. ISMnCZISm[1:0]

Table 46.18 Contents of ZPD value table row (2/2)

Bit position	Name	Function	Applied to register
19 to 10	PWM-H[9:0]	The allocation of bits 19 to 10 depends on DC-H: DC-H = 0: Bits 19 to 10 specify the duty cycle time of the *horizontal* PWM signal as a number of cycles of TB_CLK. The value range is from 0 (PWM signal switched off) to 2^{10} - 1 (duty factor = 100%). Refer to Section 46.4.5.2, Duty cycle time and duty factor of PWM signals for details.	ISMnCCMPm. ISMnCCHPm[9:0]
	IHD[3:0]	DC-H = 1: Bits 19 to 14 are ignored. Bits 13 to 10 define which transistors are activated to supply the *horizontal* coil with current. Refer to Section 46.4.7.1, Motor driving for details.	ISMnCIOCm. ISMnCIHDm[3:0]
9 to 0	PWM-V[9:0]	The allocation of bits 9 to 0 depend on DC-V: DC-V = 0: Bits 9 to 0 specify the duty cycle time of the *vertical* PWM signal as a number of cycles of TB_CLK The value range is from 0 (PWM signal switched off) to 2^{10} - 1 (duty factor = 100%).	ISMnCCMPm. ISMnCCVPm[9:0]
	IVD[3:0]	DC-H = 1: Bits 9 to 4 are ignored. Bits 3 to 0 define which transistors are activated to supply the *vertical* coil with current.	ISMnCIOCm. ISMnCIVDm[3:0]

NOTES

1. When ZMT = 1 and the PWM signal information is changed (while direct control is disabled), the last set quadrant information CQI[1:0] is used for the quadrant encoding of the PWM signals. Therefore, in this case the PWM signals may lack the correct quadrant information.
   However, for reliable results, it is not advisable to perform ZPD measurements while PWM signals are applied concurrently.
2. When direct control is enabled (DC-H and/or DC-V = 1), the quadrant information CQI[1:0] is ignored because IHD[3:0] and IVD[3:0] imply the quadrant information.
3. To perform ZPD measurements over a longer time, ZMT can be set to 1 in successive rows. ISMn evaluates the overall result at the time ZMT is reset. Refer to **Section (2), Result evaluation when ZPD measurement is switched off** for details.

CAUTION

The ZPD tables in the RAM must be defined by the application such that the discharge of any inductive load is performed before using direct control to change to a high-impedance state (at least three transistors off).
Otherwise, severe damage to the output stages of the I/O ports may result.

PWM table example

Typically, a PWM table contains sine/cosine value pattern sets as PWM duty cycle definitions, in conjunction with the associated quadrants. Mathematically, the PWM duty cycle values are the sine and cosine values of the angle of motor position, and the quadrants are those in the mathematical definiton within the X-Y coordinate system.

For each “microstep” of the motor (i.e. one of 128 or 512 entries), an angle is assigned, and for this angle, a horizontal and vertical PWM shall be applied, whose duty cycles are according to the sine and cosine values of the angle.

The sequence of quadrants within the table depends on the desired rotation direction of the motor. For a rotation in positive direction in the mathematical sense, the quadrant sequence must be 3, 2, 1, 0; in negative direction, the sequence would be 0, 1, 2, 3. In any case, the number of the first quadrant and the offset of the sine and cosine values depends on the initial position of the motor at position zero.

Like this, a PWM table of normal precision (128 steps), positive (right) turning when moving to higher positions, motor zero position at 0°, will look like this (excerpt):

```
/*  CVP     CHP     CQI   IVR IHR */

{                           // 4th quadrant dir. 0, speed 0

{{{ 0x000L, 0x3FFL, 0x3L, 1L, 1L, 0L },
  { 0x032L, 0x3FDL, 0x3L, 1L, 1L, 0L },
  { 0x064L, 0x3FAL, 0x3L, 1L, 1L, 0L },
  { 0x096L, 0x3F3L, 0x3L, 1L, 1L, 0L },
  { 0x0C7L, 0x3EBL, 0x3L, 1L, 1L, 0L },
  { 0x0F8L, 0x3E0L, 0x3L, 1L, 1L, 0L },
  { 0x128L, 0x3D2L, 0x3L, 1L, 1L, 0L },
  { 0x158L, 0x3C3L, 0x3L, 1L, 1L, 0L },
  { 0x187L, 0x3B1L, 0x3L, 1L, 1L, 0L },
  { 0x1B5L, 0x39CL, 0x3L, 1L, 1L, 0L },
  { 0x1E2L, 0x386L, 0x3L, 1L, 1L, 0L },
  { 0x20DL, 0x36DL, 0x3L, 1L, 1L, 0L },
...
  { 0x3C3L, 0x158L, 0x3L, 1L, 1L, 0L },
  { 0x3D2L, 0x128L, 0x3L, 1L, 1L, 0L },
  { 0x3E0L, 0x0F8L, 0x3L, 1L, 1L, 0L },
  { 0x3EBL, 0x0C7L, 0x3L, 1L, 1L, 0L },
  { 0x3F3L, 0x096L, 0x3L, 1L, 1L, 0L },
  { 0x3FAL, 0x064L, 0x3L, 1L, 1L, 0L },
  { 0x3FDL, 0x032L, 0x3L, 1L, 1L, 0L },

                             // 3rd quadrant dir. 0, speed 0
  { 0x3FFL, 0x000L, 0x2L, 1L, 1L, 0L },
  { 0x3FDL, 0x032L, 0x2L, 1L, 1L, 0L },
  { 0x3FAL, 0x064L, 0x2L, 1L, 1L, 0L },
  { 0x3F3L, 0x096L, 0x2L, 1L, 1L, 0L },
  { 0x3EBL, 0x0C7L, 0x2L, 1L, 1L, 0L },
  { 0x3E0L, 0x0F8L, 0x2L, 1L, 1L, 0L },
  { 0x3D2L, 0x128L, 0x2L, 1L, 1L, 0L },
  { 0x3C3L, 0x158L, 0x2L, 1L, 1L, 0L },
  { 0x3B1L, 0x187L, 0x2L, 1L, 1L, 0L },
  { 0x39CL, 0x1B5L, 0x2L, 1L, 1L, 0L },
  { 0x386L, 0x1E2L, 0x2L, 1L, 1L, 0L },

...                       // end of the table, 1st quadrant

  { 0x187L, 0x3B1L, 0x0L, 1L, 1L, 0L },
  { 0x158L, 0x3C3L, 0x0L, 1L, 1L, 0L },
  { 0x128L, 0x3D2L, 0x0L, 1L, 1L, 0L },
  { 0x0F8L, 0x3E0L, 0x0L, 1L, 1L, 0L },
  { 0x0C7L, 0x3EBL, 0x0L, 1L, 1L, 0L },
  { 0x096L, 0x3F3L, 0x0L, 1L, 1L, 0L },
  { 0x064L, 0x3FAL, 0x0L, 1L, 1L, 0L },
  { 0x032L, 0x3FDL, 0x0L, 1L, 1L, 0L } }

(to be continued for other directions and speeds)
```

Due to manufactural tolerances of the motor, dedicated values may diverge from the sine or cosine values, so that an optimized (more smooth) rotation of the motor can be achieved.

In an overview, the table can have the following content as a graphical representation:

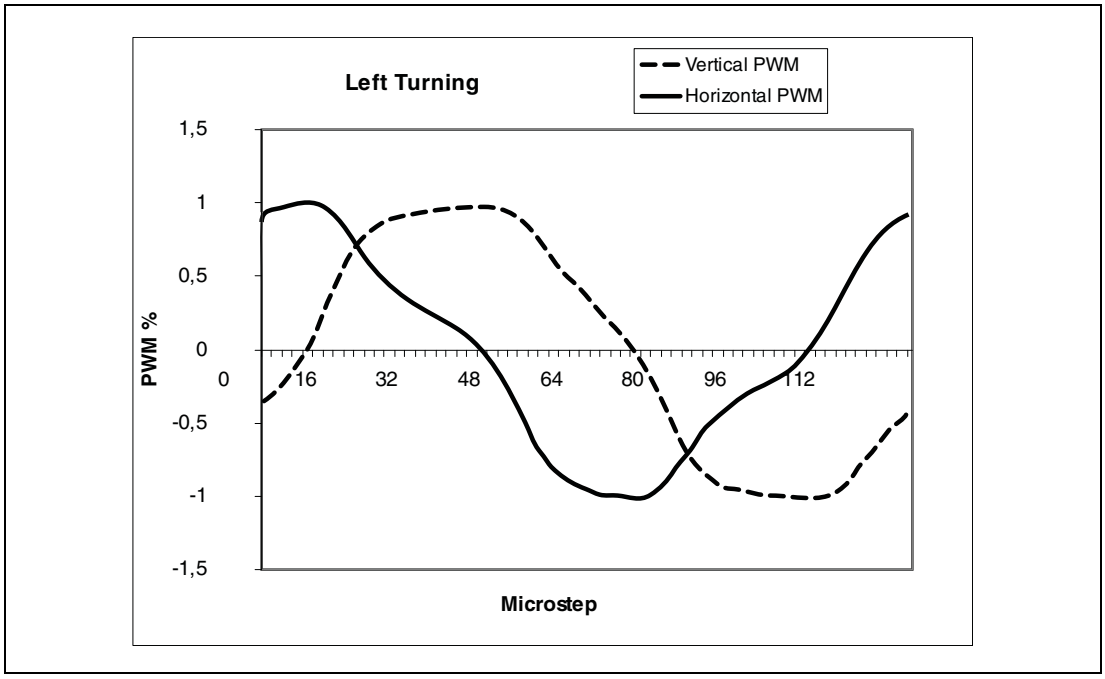


Figure 46.17 PWM table graphical representation example

46.4.9 Timing

46.4.9.1 Channel data update timing

All channels are updated within one PWM cycle. The channel data is buffered ensuring that all channels are updated synchronously.

Application supplies new data

The data of motor channels where channel management is not set can be modified at any time. The new register values take effect immediately and are applied at the beginning of the next PWM cycle.

Channel management supplies new data

The data of motor channels where channel management is set is automatically updated in regular intervals which are a multiple of PWM cycles (indicated by a START event; see **Section 46.4.4.2, Channel data update interval**).

- The ISMn sequencer feeds the required PWM signal or ZPD information from the assigned RAM table into the corresponding registers.
- The registers values are applied at the beginning of the next PWM cycle.

Example

The following example illustrates the value buffering and channel data update timing.

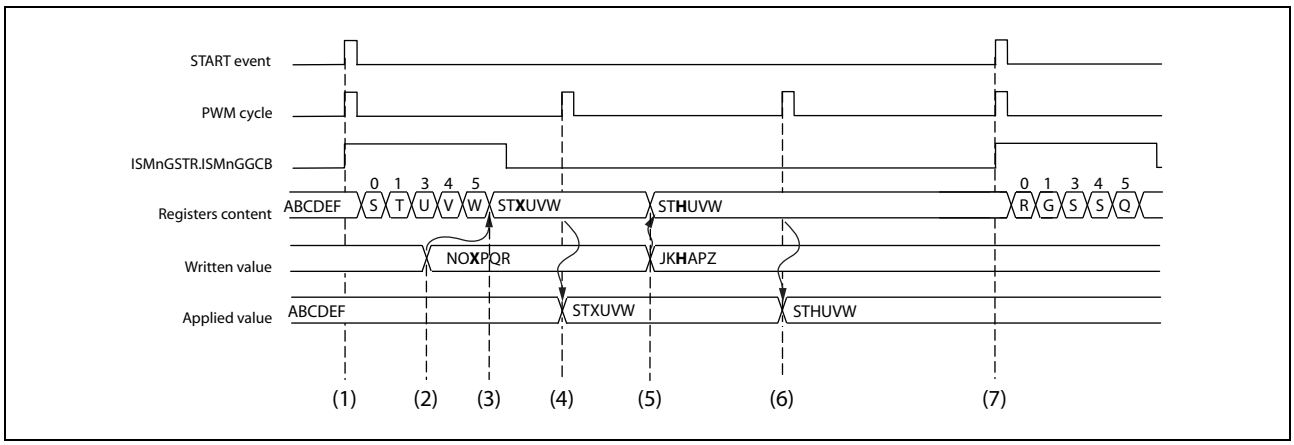


Figure 46.18 Channel data update timing

Channel management is set for all motor channels except for motor channel 2:

ISMnGSTR.ISMnCCENm = 1 for m = 0, 1, 3, 4, 5, and

ISMnGSTR.ISMnCCEN2 = 0

1. The sequencer is enabled at the beginning of a START event (ISMnGSTR.ISMnGGCB = 1) and feeds the values from the assigned RAM table into the corresponding registers (except for channel 2).
   – In motor driving mode this is the PWM signal information of the calculated rotor position.
   – In ZPD operation mode this is the data of the ZPD table.

Channel	0	1	2	3	4	5
Register values	S	T	–	U	V	W

   Afterwards, the sequencer returns into idle state (ISMnGSTR.ISMnGGCB = 0) and stays idle until the next START event.

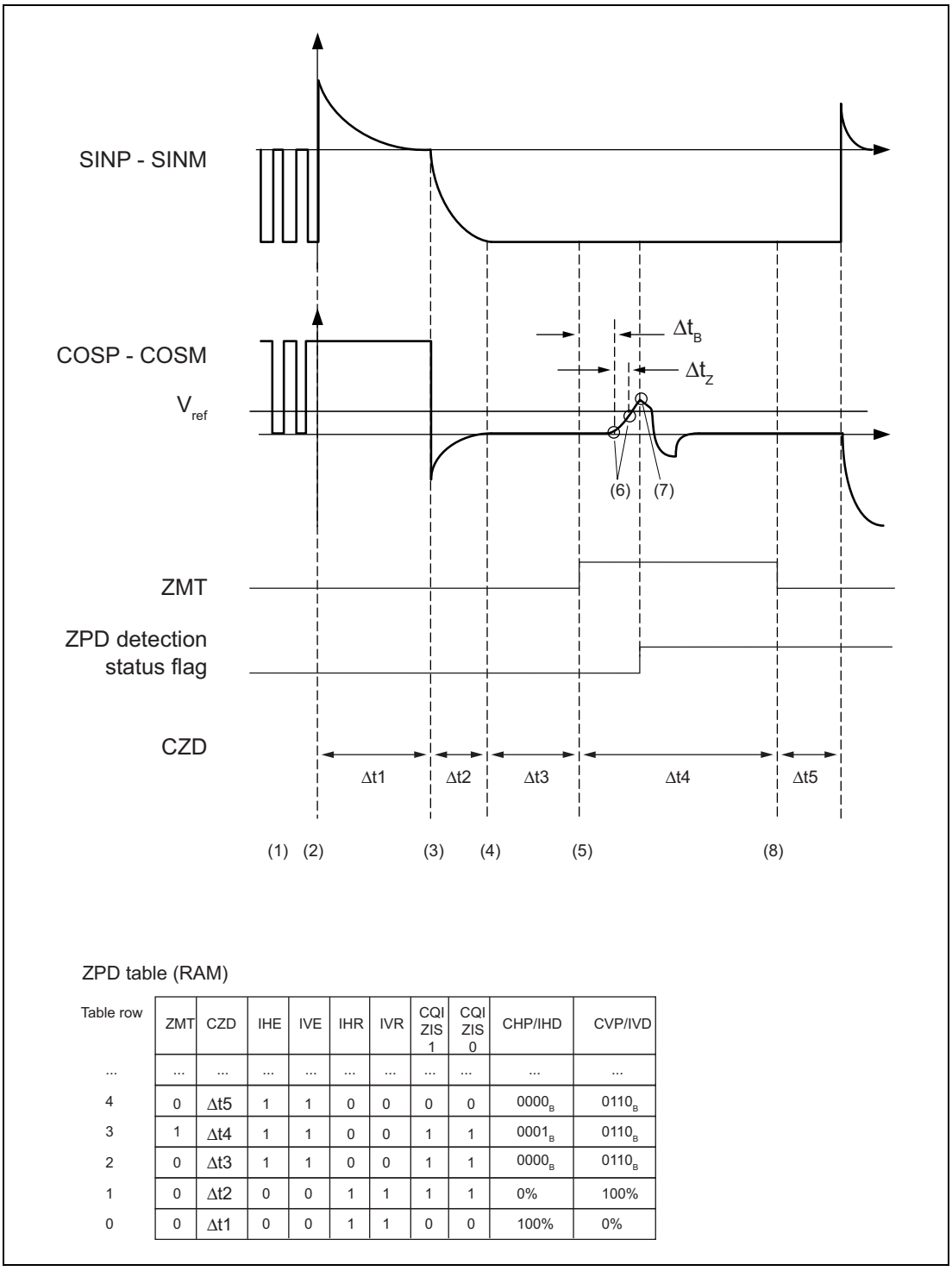
2. Meanwhile the following values are written by the application:

Channel	0	1	2	3	4	5
Written values	N	O	X	P	Q	R

3. Only the register contents of motor channel 2 are updated, because channel 2 is the only channel where channel management is not set.
4. The values take effect at the beginning of the next PWM cycle.
5. The application can update the values of motor channel 2 at any time.
6. New values take effect the next PWM cycle.
7. The values for channels 0, 1, 3, 4, and 5 are recalculated and updated when the next START event occurs.

46.4.9.2 Timing of ZPD measurements with channel management

The following example illustrates the timing of ZPD measurements with channel management.



ZPD table (RAM)

Table row	ZMT	CZD	IHE	IVE	IHR	IVR	CQI ZIS 1	CQI ZIS 0	CHP/IHD	CVP/IVD
...	...	...	...	...	...	...	...	...	...	...
4	0	Δt5	1	1	0	0	0	0	0000_B	0110_B
3	1	Δt4	1	1	0	0	1	1	0001_B	0110_B
2	0	Δt3	1	1	0	0	1	1	0000_B	0110_B
1	0	Δt2	0	0	1	1	1	1	0%	100%
0	0	Δt1	0	0	1	1	0	0	100%	0%

Figure 46.19 Timing of ZPD measurements with channel management

ZMT	CZD	IHE	IVE	IHR	IVR	CQI ZIS 1	CQI ZIS 0	CHP/IHD	CVP/IVP	Function
–										1. The channel is operated in motor driving mode. PWM signals are output.
0	Δt1	0	0	1	1	0	0	100%	0%	2. When ZPD operation mode is set: The table index counter is reset and the values of table row 0 are fed into the corresponding registers. These values are applied at the beginning of the next PWM cycle. Direct control is not set. Therefore, according to the set PWM signal information: – The horizontal coil is supplied with current. – The vertical coil is disconnected from the current. Induced voltages are reduced using recirculation.
0	Δt2	0	0	1	1	1	1	0%	100%	3. When Δt1 has elapsed, the registers are updated with the values of table row 1. According to the set PWM signal information, the coils are supplied with currents in turn: – The horizontal coil is disconnected from the current. Induced voltages are reduced using recirculation. – The vertical coil is supplied with current.
0	Δt3	1	1	0	0	1	1	0000_B	0110_B	4. When Δt2 has elapsed, the registers are updated with the values of table row 2. Direct control is set for both coils. The coils are supplied with currents according to the settings made: – The horizontal coil is disconnected from the current. – The vertical coil is supplied with current.

ZMT	CZD	IHE	IVE	IHR	IVR	CQI ZIS 1	CQI ZIS 0	CHP/IHD	CVP/IVP	Function
1	Δt4	1	1	0	0	1	1	0001_B	0110_B	5. When Δt3 has elapsed, the registers are updated with the values of table row 3. As before, direct control is used to supply the coils with current. The start measurement trigger ZMT is set. Therefore, after the configured blanking delay Δt_B, ISMn starts measuring the induced voltage at input pin COSPm. The voltages measured are compared with the reference voltage V_{ref}.
										6. After each measurement sequence (Δt_Z) the ZPD detection status flag is updated: When – according to digital noise filtering – the measured voltage is lower than the reference voltage, the ZPD detection status flag remains 0. Otherwise it is set to 1.
										7. If it is detected that the motor has not reached its zero position yet (number of measurements above threshold), ISMn short-circuits the coil until the ZPD measurement trigger ISMnZPDCTL.ISMnCZMTm is reset to 0 (ZPD vibration damping).
0	Δt5	1	1	0	0	0	0	0000_B	0110_B	8. When Δt4 has elapsed, the registers are updated with the values of table row 4. As before, direct control is used to supply the coils with current. ZPD measurements are stopped because the start measurement trigger ZMT has been reset. ISMn checks the ZPD detection status flag, whether the motor has reached its zero position. This is not the case in this example. Therefore ISMn proceeds processing the ZPD table when Δt5 has elapsed. For example, ZPD measurements might be started again after one turn of the motor.

46.4.10 Interrupt requests and status flags

Interrupts and status flags indicate certain states:

- Depending on the operation mode, a rotor has reached its target or zero position.
- The sequencer is in idle state.
- The ZPD measurements have started after the configured blanking delay.
- A motor did not reach its zero position.

Interrupt requests

The generation of interrupt requests can be enabled/disabled in the global control register ISMnGCTL (except for IRQ_ZPD).

By default, the generation of interrupt requests is disabled.

Status flags

In the ISMnGIP register, status flags are set according to interrupt conditions – no matter whether the generation of the interrupt request is disabled.
The setting of interrupt status flags can not be masked. Status flags need to be cleared by writing 1 to the corresponding bits in ISMnGSTC.

46.4.10.1 IRQ_REACHED: Target/zero position reached

IRQ_REACHED indicates that a motor has reached its target/zero position.

- In motor driving operation mode, this is the case when the motor's actual position (ISMnVAR4CFGm) and the configured target position (ISMnPAR0CFGm) do match and the actual speed (ISMnVAR1CFGm) is zero.
- In ZPD operation mode, this is the case when the ZPD measurement status flag ISMnZPDIP.ISMnCZIPm = 0 at the end of a set of measurement sequences (measurement start trigger ISMnZPDCTL.ISMnCZMTm = 0).

As result, the interrupt status flag of the respective channel ISMnGIP.ISMnGIPSSREm is set to 1. When ISMnGCTL.ISMnGIEREm = 1, ISMn additionally generates the interrupt request IRQ_REACHED.

NOTES

1. IRQ_REACHED is generated only once per channel since the last time the status flag ISMnGIPSSREm was cleared by writing 1 to ISMnGSTC.ISMnGIPCLREm.
2. IRQ_REACHED is a general interrupt that is issued for any motor channel where ISMnGCTL.ISMnGIEREm = 1 that has reached its target/zero position.

46.4.10.2 IRQ_DONE: Sequencer idle interrupt

The usage of channel management can only be enabled/disabled (ISMnGSTR.ISMnCCENm), when the sequencer is idle.

IRQ_DONE indicates that the sequencer has returned to the idle state (until the next START event) after all channels' data has been updated.

The status flag ISMnGIP.ISMnGIPSSDO is set to 1 when the sequencer is in idle state and can be cleared by writing 1 to ISMnGSTC.ISMnGIPCLDO.

The generation of the respective interrupt request can be enabled/disabled in ISMnGCTL.ISMnGIEDO.

Refer to Section 46.4.4.2, Channel data update interval for details.

46.4.10.3 IRQ_ZPDAD: Start of ZPD measurements

IRQ_ZPDAD indicates when the actual ZPD measurements start after the configured blanking delay:

- The status flag ISMnGIP.ISMnGIPSSAD is set to 1.
- The respective interrupt request is issued when ISMnGCTL.ISMnGIEAD = 1.

The status flag ISMnGIP.ISMnGIPSSAD can be cleared by writing 1 to ISMnGSTC.ISMnGIPCLAD.

46.4.10.4 IRQ_ZPD: Zero point not detected

In ZPD operation mode, IRQ_ZPD indicates that a motor has not reached its zero position yet (refer to Section 46.4.7.3, Set of measurement sequences for details):

- The channel's interrupt status flag ISMnZPDIP.ISMnCZIPm is set to 1.
- The respective interrupt request is additionally issued.

NOTES

1. IRQ_ZPD is generated only once per channel since the last time, the status flag ISMnZPDIP.ISMnCZIPm has been cleared.
   When channel management is not set, the status flag ISMnZPDIP.ISMnCZIPm can be cleared by writing 1 to ISMnZPDSTC.ISMnCCLZPm. This is helpful, for example, to verify the ZPD detection status in a regular interval.
   When channel management is set, ISMnZPDIP.ISMnCZIPm is automatically cleared when a new set of measurement sequences is started (measurement start trigger ISMnZPDCTL.ISMnCZMTm changes from 0 to 1).
2. IRQ_ZPD is a general interrupt that is issued for any motor channel that is operated in ZPD operation mode where the motor's zero position has not been reached yet.

46.4.11 Soft reset

ISMn can be reset by software. A soft reset has the following effects:

- PWM signal output is stopped and all output signals are forced to reset level
- Internal state machines are stopped
- The following data is *not* initialized, it can be modified while the soft reset is active:
  - Register contents
  - Movement calculator variables
  - Movement parameters and measurement parameters
  - RAM data (PWM value look-up tables and ZPD tables)

The soft reset is carried out by setting ISMnGCTL.ISMnGGEN = 0.

46.4.12 Procedures

46.4.12.1 Initialization of Intelligent Stepper Motor Driver

The Intelligent Stepper Motor Driver must be initialized before it can be used for motor driving or zero point detection:

1. Specify the time base clock TB_CLK and the channel data update interval by setting the prescalers in ISMnGCFG.
   Refer to Section 46.4.3, Clocking and Section 46.4.4.2, Channel data update interval for details.

NOTE

The channel data update interval is relevant only when channel management is used.

2. Specify the recirculation side in ISMnCOPT.
3. When using channel management, set up the PWM value look-up tables and the ZPD tables according to the address and format specifications described in Section 46.4.8, PWM value look-up tables and ZPD tables in RAM.
4. When using ZPD operation mode, the following general features must be set up in the ZPD control register ISMnGZPDCTL:
   - ZPD measurement clock M_CLK
   - The reference voltages available
   - The digital noise filter settings
   - The usage of the ZPD unit
   - The usage of the ZPD measurement start interrupt IRQ_ZPDAD (ISMnGCTL.ISMnGIEAD = 1).

Now the motor channels can be configured as required. Follow the basic procedures contained in the following chapters:

- Section 46.4.12.2, Basic procedures: Motor driving with channel management
- Section 46.4.12.3, Basic procedures: Motor driving without channel management
- Section 46.4.12.4, Basic procedures: ZPD with channel management
- Section 46.4.12.5, Basic procedures: ZPD without channel management

46.4.12.2 Basic procedures: Motor driving with channel management

To use channel management for driving a motor, initialize ISMn if not done yet (see Section 46.4.12.1, Initialization of Intelligent Stepper Motor Driver), then configure the channel specific settings, and finally supply the desired target position:

Motor channel configuration

Each motor channel is configured individually:

1. Write 0 to ISMnCCMRm.ISMnCCZPm to specify the operation mode “motor driving” for motor channel m.
2. Configure the motor’s movement parameters in ISMnPAR[9:0]CFGm and initialize the variable parameters ISMnVAR[6:0]CFGm.
3. In ISMnCCMRm, specify the following:
   – The look-up table to be used for PWM signal generation (bits ISMnCCCTm[2:0]).
   – The number of microsteps within one 360° turn (defined by the movement precision in bit ISMnCCCPm).
   – The output delays for the PWM signals (bits ISMnCCDHm[3:0] and ISMnCCDVm[3:0]).
4. In ISMnGCTL.ISMnGIEREm, specify whether or not the interrupt IRQ_REACHED is generated when the target position is reached.
5. Write 1 to ISMnGSTR.ISMnCCENm to specify that motor channel m is controlled by channel management.
6. When all motor channels are set up as required, write 1 to ISMnGCTL.ISMnGGCE to switch on channel management.
7. Write the target position for the rotor to ISMnPAR0CFGm.

The settings take effect after the next START event (see Section 46.4.4.2, Channel data update interval).

ISMn automatically moves the rotor step by step to its target position. The target position is indicated by the interrupt IRQ_REACHED (when configured) and the status flag ISMnGIP.ISMnGIPSSREm.

NOTES

1. The parameters that are used to calculate intermediate rotor positions (ISMnPAR[9:1]CFGm and ISMnVAR[6:1]CFGm) can be read at any time. Note that old values may be read while the sequencer is busy updating the channels’ data (ISMnGSTR.ISMnGGCB = 1). When ISMnGCTL.ISMnGIEDO = 1, the interrupt IRQ_DONE is generated whenever the sequencer returns to its idle state and valid data can be read.
2. Only positive or zero values for target positions are allowed. Negative target positions will cause the algorithm of the channel management to stop immediately, if the current position becomes negative.

Update of target position

To move the rotor to a new target position:

1. Clear the IRQ_REACHED status flag of channel m by writing 1 to ISMnGSTC.ISMnGIPCLREm.
2. Update the target position of the respective rotor in ISMnPAR0CFGm.

The new value takes effect after the next START event.

Update of configuration

Except for the target position, a motor channel's settings and the PWM value look-up tables can only be modified when channel management is not enabled (either for that channel and/or for all channels).

Refer to Section 46.4.12.6, Disabling/re-enabling channel management for details.

Example

A senseful set of parameters is given by the following:

PMP	=	0x00000000 (initial position assumed to be zero)
PDF	=	0x06
PAL	=	0x20
PDL	=	0x20
PMS	=	0x000009C0
PHC	=	0x00000700
PS1	=	0x0000010E
PS2	=	0x00000620
PS3	=	0x000001C3
PS4	=	0x0000056B

All variable registers shall be cleared to zero content before starting the channel management.

46.4.12.3 Basic procedures: Motor driving without channel management

To drive a motor without using channel management, initialize ISMn if not done yet (see Section 46.4.12.1, Initialization of Intelligent Stepper Motor Driver), then configure the channel specific settings, and finally apply the PWM signal information whenever the rotor is required to be moved.

Motor channel configuration

Each motor channel is configured individually:

1. Write 0 to ISMnGSTR.ISMnCCENm to specify that all parameters for motor channel m are input by the application (no channel management).
2. Write 0 to ISMnCCMRm.ISMnCCZPm to specify the operation mode "motor driving" for motor channel m.
3. In ISMnCCMRm, specify the output delays for the PWM signals (bits ISMnCCDHm[3:0] and ISMnCCDVm[3:0]).
4. In ISMnCCMPm, specify the quadrant information and the PWM signal pulse widths for the sine and cosine side of the desired position.
5. In ISMnCIOCm, enable/disable recirculation for the horizontal and/or vertical PWM signal.

The settings take effect at the beginning of the next PWM cycle.

Update of rotor position

To move a rotor one step further:

1. Update the quadrant information and the PWM signal pulse widths in ISMnCCMPm.
2. If required, alter the recirculation enable flags in ISMnCIOCm.

The settings take effect at the beginning of the next PWM cycle.

Update of configuration

PWM signal information and the recirculation enable flags can be modified anytime, for a channel which is not under channel management control.

Refer to Section 46.4.12.6, Disabling/re-enabling channel management for details.

46.4.12.4 Basic procedures: ZPD with channel management

For ZPD using channel management, initialize ISMn if not done yet (see **Section 46.4.12.1, Initialization of Intelligent Stepper Motor Driver**), then configure the motor channel's specific ZPD settings, and finally enable channel management to start the ZPD procedure.

Motor channel configuration

Each motor channel is configured individually:

1. Write 1 to ISMnCCMRm.ISMnCCZPm to specify the operation mode "ZPD" for motor channel m.
2. In ISMnCCMRm, also specify the following:
   - The ZPD table to be used (ISMnCCCTm[2:0])
   - Last table row to be considered (ISMnCCTLm[6:0])
   - When using PWM signals to drive the motor:
     Output delays for the horizontal and/or vertical PWM signals (ISMnCCDHm[3:0] and ISMnCCDVm[3:0]).
     The number of microsteps within one 360° turn (defined by the movement precision in bit ISMnCCCPm)
3. In ISMnGCTL.ISMnGIEREm, specify whether the interrupt request IRQ_REACHED is generated when the motor reaches its zero position.
4. In ISMnZPDCMPm.ISMnCZBTm[15:0], specify the blanking time.
5. In ISMnZPDCMPm.ISMnCZSSm, specify whether the ZPD vibration damping feature is used.
6. In ISMnZPDOPT.ISMnCZRSm, specify the reference voltage.
7. Write 1 to ISMnGSTR.ISMnCCENm to specify that motor channel m is controlled by channel management.
8. When all motor channels are set up as required, write 1 to ISMnGCTL.ISMnGGCE to switch on channel management.

The settings take effect after the next START event (see Section 46.4.4.2, Channel data update interval).

ISMn automatically processes the control sequence stored in the assigned ZPD table. The zero position is indicated by the interrupt IRQ_REACHED (when configured) and the status flag ISMnGIP.ISMnGIPSSREm.

When the zero position is reached, channel management and ZPD operation mode are automatically disabled for this channel.

NOTE

The registers that are filled from the ZPD table can be read at any time. Note that old values may be read while the sequencer is busy updating the channels' data (ISMnGSTR.ISMnGGCB = 1). When ISMnGCTL.ISMnGIEDO = 1, the interrupt IRQ_DONE is generated whenever the sequencer returns to its idle state.

Update of configuration

A motor channel's settings can only be modified when channel management is disabled for this channel. ZPD tables can only be modified, when channel management is switched-off.

Refer to Section 46.4.12.6, Disabling/re-enabling channel management for details.

46.4.12.5 Basic procedures: ZPD without channel management

Without channel management, zero point detection can be controlled using the ZPD registers. The motor can be driven by PWM signals or direct control concurrently.

NOTE

Zero point detection without channel management requires a detailed knowledge of the relationships between the various settings.

Due to the large number of possible combinations, zero point detection without channel management is not covered in this documentation.

46.4.12.6 Disabling/re-enabling channel management

Channel management can be switched off for all channels or disabled for individual motor channels.

In both operation modes, the rotor holds its position – unless the pulse width, quadrant information etc. are updated by the application (see Section (1), Provision of PWM signal information).

Switching off channel management

All settings can be modified when channel management is switched off. This includes the general settings, a channel's specific settings, and the RAM tables:

1. Write 0 to ISMnGCTL.ISMnGGCE.
   Channel management is switched off no matter whether ISMnGSTR.ISMnCCENm = 1.
   When the sequencer is currently busy updating channels' data (ISMnGSTR.ISMnGGCB = 1), channel management stops as soon as the sequencer returns into idle state (ISMnGSTR.ISMnGGCB = 0).
2. If required, modify the settings and/or the RAM tables as desired.

When all motor channels are configured as required, write 1 to ISMnGCTL.ISMnGGCE to switch on channel management again.

Disabling channel management for one motor channel

To disable the usage of channel management for a certain channel it is not mandatory to switch off channel management for all channels:

- Wait until the sequencer is in idle state (ISMnGSTR.ISMnGGCB = 0), and then set ISMnGSTR.ISMnCCENm = 0.

When ISMnGCTL.ISMnGIEDO = 1, the interrupt request IRQ_DONE indicates when the sequencer has finished updating the channels and has returned to its idle state until the next START event.

When the motor channel is set up as required, wait until the sequencer is in idle state again and then write 1 to ISMnGSTR.ISMnCCENm to switch on channel management for that channel again.

46.5 Intelligent Stepper Motor Driver Registers

This section contains a description of all the registers of the Intelligent Stepper Motor.

46.5.1 Intelligent Stepper Motor Driver registers overview

The Intelligent Stepper Motor is controlled and operated by the following registers.

In the table “CHm” is an abbreviation for “motor channel m”.

Table 46.19 Intelligent Stepper Motor Driver registers overview (1/2)

Register name	Symbol	Address
Common control registers		
Global control	ISMnGCTL	<ISMn_base> + 0000_H
Global status	ISMnGSTR	<ISMn_base> + 0004_H
Global timebase control	ISMnGCFG	<ISMn_base> + 0008_H
Channel timebase reading	ISMnGCNT	<ISMn_base> + $000C_H$
Global interrupt pending	ISMnGIP	<ISMn_base> + 0014_H
Global interrupt pending clear	ISMnGSTC	<ISMn_base> + 0018_H
Channel control	ISMnCCMRm	<ISMn_base> + $001C_H$ + m x 4_H
Channel PWM setting	ISMnCCMPm	<ISMn_base> + $004C_H$ + m x 4_H
Channel I/O control setting	ISMnCIOCm	<ISMn_base> + $007C_H$ + m x 4_H
Channels recirculation side setting	ISMnCOPT	<ISMn_base> + 0094_H
ISMn movement calculator parameters		
Target motor position CHm	ISMnPAR0CFGm	<ISMn_base> + $00C8_H$ + m x 28_H
Damping factor CHm	ISMnPAR1CFGm	<ISMn_base> + $00CC_H$ + m x 28_H
Acceleration limit CHm	ISMnPAR2CFGm	<ISMn_base> + $00D0_H$ + m x 28_H
Deceleration limit CHm	ISMnPAR3CFGm	<ISMn_base> + $00D4_H$ + m x 28_H
Maximum speed CHm	ISMnPAR4CFGm	<ISMn_base> + $00D8_H$ + m x 28_H
Hysteresis correction CHm	ISMnPAR5CFGm	<ISMn_base> + $00DC_H$ + m x 28_H
Speed threshold 1 CHm	ISMnPAR6CFGm	<ISMn_base> + $00E0_H$ + m x 28_H
Speed threshold 2 CHm	ISMnPAR7CFGm	<ISMn_base> + $00E4_H$ + m x 28_H
Speed threshold 3 CHm	ISMnPAR8CFGm	<ISMn_base> + $00E8_H$ + m x 28_H
Speed threshold 4 CHm	ISMnPAR9CFGm	<ISMn_base> + $00EC_H$ + m x 28_H
ISMn movement calculator variables		
Actual acceleration/deceleration CHm	ISMnVAR0CFGm	<ISMn_base> + $01B8_H$ + m x $1C_H$
Actual speed CHm	ISMnVAR1CFGm	<ISMn_base> + $01BC_H$ + m x $1C_H$
Previous iteration speed CHm	ISMnVAR2CFGm	<ISMn_base> + $01C0_H$ + m x $1C_H$
Temporary calculator register PT1 CHm	ISMnVAR3CFGm	<ISMn_base> + $01C4_H$ + m x $1C_H$
Actual position, current result CHm	ISMnVAR4CFGm	<ISMn_base> + $01C8_H$ + m x $1C_H$
Virtually displayed position CHm	ISMnVAR5CFGm	<ISMn_base> + $01CC_H$ + m x $1C_H$
Direction/speed flag CHm	ISMnVAR6CFGm	<ISMn_base> + $01D0_H$ + m x $1C_H$
ZPD registers		
Global ZPD control	ISMnGZPDCTL	<ISMn_base> + 0010_H
ZPD counter	ISMnCCNTm	<ISMn_base> + 0034_H + m x 4_H
ZPD configuration	ISMnCZCFGm	<ISMn_base> + 0064_H + m x 4_H
ZPD info bus settings	ISMnZPDCTL	<ISMn_base> + 0098_H
ZPD options setting	ISMnZPDOPT	<ISMn_base> + $009C_H$

Table 46.19 Intelligent Stepper Motor Driver registers overview (2/2)

Register name	Symbol	Address
ZPD status	ISMnZPDSTR	<ISMn_base> + $00A0_H$
ZPD detection flag	ISMnZPDIP	<ISMn_base> + $00A4_H$
ZPD detection flag clear	ISMnZPDSTC	<ISMn_base> + $00A8_H$
ZPD blanking and vibration damping setting	ISMnZPDCMPm	<ISMn_base> + $00AC_H$ + m x 4_H
ZPD measurement activity status	ISMnZPDCSTR	<ISMn_base> + $00C4_H$
Emulation mode register		
Emulation register	ISMnEMU	<ISMn_base> + 0260_H

<ISMn_base>

The base addresses <ISMn_base> of the ISMn is defined in the first section of this chapter under the key word “Register addresses”.

46.5.2 Common control registers

46.5.2.1 ISMnGCTL – ISM global control register

This register is used to switch channel management on/off and to enable/disable the generation of interrupt requests. Additionally, it allows ISMn to be reset.

For details, refer to the following sections:

- Section 46.4.10, Interrupt requests and status flags
- Section 46.4.12.6, Disabling/re-enabling channel management
- Section 46.4.11, Soft reset

Access: This register can be read/written in 32-bit units.

Address: <ISMn_base> + 0000_H

Initial Value: 0001 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISMn GIEAD	ISMn GIEDO	ISMnGIERE[5:0]						0	0	0	0	0	0	0	ISMn GGEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ISMn GGCE	0	0	0	0	0	0	0	ISMn ZPDU*1
R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Note 1. Bit 0 has to be changed to “1” when the ZPD function is used.

CAUTION

The default value of bit 0 has to be changed to “1” when the ZPD function is used.

Table 46.20 ISMnGCTL register contents (1/2)

Bit position	Bit name	Function
31	ISMnGIEAD	Enables/disables the generation of the interrupt request IRQ_ZPDAD that indicates the actual beginning of ZPD measurements: 0: Do not generate IRQ_ZPDAD 1: Generate IRQ_ZPDAD The status flag ISMnGIP.ISMnGIPSSAD is set no matter whether the interrupt request is masked by this register.
30	ISMnGIEDO	Enables/disables the generation of the interrupt request IRQ_DONE that indicates that the sequencer has returned to idle state for the current PWM cycle: 0: Do not generate IRQ_DONE 1: Generate IRQ_DONE The status flag ISMnGIP.ISMnGIPSSDO is set no matter whether the interrupt request is masked by this register.
29 to 24	ISMnGIEREm	Enables/disables the generation of the interrupt request IRQ_REACHED that indicates that a motor has reached its target position. IRQ_REACHED can be enabled/disabled for every channel individually: 0: Do not generate IRQ_REACHED 1: Generate IRQ_REACHED The corresponding channel specific status flag in ISMnGIP.ISMnGIPSSREm is set no matter whether the interrupt request is masked by this register.
23 to 17	Reserved	When read, the value after reset is returned. When written, write the value after reset.

Table 46.20 ISMnGCTL register contents (2/2)

Bit position	Bit name	Function
16	ISMnGGEN	Enables/disables the soft reset: 0: Resets ISMn 1: Releases the soft reset **CAUTION** Before release of the soft reset by setting ISMnGGEN = 1, the ZPD measurement for all channels m have to be disabled by setting ISMnZPDCTL.ISMnCZMTm = 0.
15 to 9	Reserved	When read, the value after reset is returned. When written, write the value after reset.
8	ISMnGGCE	Specifies whether channel management is switched on/off: 0: Switched off (disabled for all channels no matter whether ISMnGSTR.ISMnCCENm = 1) 1: Switched on Channel management supplies motor channels where ISMnGSTR.ISMnCCENm = 1.
7 to 1	Reserved	When read, the value after reset is returned. When written, write the value after reset.
0	ISMnZPDU	**CAUTION** The default value of bit 0 has to be changed to “1” when the ZPD function is used.

46.5.2.2 ISMnGSTR – ISM global status register

This register is used to specify whether channel management is used for motor channel m or not. It also indicates the status of the ISMn sequencer because the usage of channel management can only be enabled/disabled when the sequencer is idle (or when channel management is switched off).

Refer to Section 46.4.9.1, Channel data update timing and Section 46.4.12.6, Disabling/re-enabling channel management for details.

Access: This register can be read/written in 32-bit units.

Address: <ISMn_base> + 0004_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	ISMn GGCB	0	ISMnCCEN[5:0]					
R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.21 ISMnGSTR register contents

Bit position	Bit name	Function
31 to 8	Reserved	When read, the value after reset is returned. When written, write the value after reset.
7	ISMnGGCB	Indicates the status of the sequencer: 0: Sequencer is idle 1: Sequencer is updating channels' data This bit is read-only.
6	Reserved	When read, the value after reset is returned. When written, write the value after reset.
5 to 0	ISMnCCENm	Enables/disables the usage of channel management for motor channel m: 0: Do not use channel management 1: Use channel management

NOTE

When reading parameters that are used to calculate intermediate rotor positions (ISMnPAR[9:1]CFGm and ISMnVAR[6:1]CFGm) while the sequencer is busy (ISMnGSTR.ISMnGGCB = 1), old values may be read.

46.5.2.3 ISMnGCFG – ISM global timebase control register

This register specifies the timebase clock TB_CLK and the channel data update interval.

CAUTION

If the values in this register are changed during PWM output, the PWM output may become disturbed intermediately. Thus make sure to suppress any PWM output when changing this register by, for instance,

- **keeping the ISMn module in software reset (ISMnGCTL.ISMnGGEN = 0) or**
- **setting the PWM duty cycle to 0% or 100 %**

or any other appropriate means.

Access: This register can be read/written in 32-bit units when channel management is switched off (ISMnGCTL.ISMnGGCE = 0).
This register is read-only when channel management is switched on (ISMnGCTL.ISMnGGCE = 1).

Address: <ISMn_base> + 0008_H

Initial Value: $000F\ 00FF_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	ISMnGGTB[3:0]			
R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	ISMnGGUD[7:0]							
R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.22 ISMnGCFG register contents (1/2)

Bit position	Bit name	Function
31 to 20	Reserved	When read, the value after reset is returned. When written, write the value after reset.
19 to 16	ISMn GGTB[3:0]	Specifies the timebase clock TB_CLK which is PCLK / (ISMnGGTB[3:0] + 1)
15 to 8	Reserved	When read, the value after reset is returned. When written, write the value after reset.

ISMn GGTB[3:0]	Timebase clock
0000	PCLK / 1
0001	PCLK / 2
...	...
1111	PCLK / 16

Table 46.22 ISMnGCFG register contents (2/2)

Bit position	Bit name	Function
7 to 0	ISMn GGUD[7:0]	Specifies the channel data update interval which is (ISMnGGUD[7:0] + 1) / f_{PWM}:

ISMn GGUD[7:0]	Channel data update interval
0000 0000	1 / f_{PWM}
0000 0001	2 / f_{PWM}
...	...
1111 1111	256 / f_{PWM}

Refer to Section 46.4.4.2, Channel data update interval for details.

46.5.2.4 ISMnGCNT – ISM channel timebase reading register

This register can be used to read the current value of the TB_CLK counter.

Refer to Section 46.4.5.2, Duty cycle time and duty factor of PWM signals for details.

Access: This register can be read in 32-bit units.

Address: <ISMn_base> + $000C_H$

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	ISMnGGTR[9:0]									
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 46.23 ISMnGCNT register contents

Bit position	Bit name	Function
31 to 10	Reserved	When read, the value after reset is returned.
9 to 0	ISMn GGTR[9:0]	10-bit counter value

46.5.2.5 ISMnGIP – ISM global interrupt pending register

This register indicates the interrupt status. The status flags are set no matter whether the generation of the interrupt requests are masked in ISMnGCTL or not.

Refer to Section 46.4.10, Interrupt requests and status flags for details.

Access: This register can be read in 32-bit units.

Address: <ISMn_base> + 0014_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	ISMn GIPS SAD	ISMn GIPS SDO	ISMnGIPSSRE[5:0]					
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 46.24 ISMnGIP register contents

Bit position	Bit name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	ISMn GIPSSAD	Indicates whether ZPD measurements have actually begun: 0: ZPD measurements have not begun yet 1: ZPD measurements have been started When configured the interrupt request IRQ_ZPDAD has additionally been issued. This bit is cleared when ISMnGSTC.ISMnGIPCLAD is set to 1.
6	ISMn GIPSSDO	Indicates that the sequencer has returned to idle state for the current channel data update interval: 0: Sequencer busy 1: Sequencer returned to idle state When configured, the interrupt request IRQ_DONE has additionally been issued. This bit is cleared when ISMnGSTC.ISMnGIPCLDO is set to 1.
5 to 0	ISMn GIPSSREm	Indicates whether the motor of channel m has reached its target/zero position: 0: Target/zero position has not been reached 1: Target/zero position has been reached When configured, the interrupt request IRQ_REACHED has additionally been issued. IRQ_REACHED is generated for any motor that has reached its target/zero position. This bit is cleared when ISMnGSTC.ISMnGIPCLREm is set to 1.

46.5.2.6 ISMnGSTC – ISM global interrupt pending clear register

This register is the clear control register of ISMnGIP.

Refer to Section 46.4.10, Interrupt requests and status flags and Section 46.5.2.5, ISMnGIP – ISM global interrupt pending register for details.

Access: This register can be written in 32-bit units.
Reading this register returns 0000 0000_H.

Address: <ISMn_base> + 0018_H

Initial Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	ISMn GIP CLAD	ISMn GIP CLDO	ISMnGIPCLRE[5:0]					
R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 46.25 ISMnGSTC register contents

Bit position	Bit name	Function
31 to 8	Reserved	When read, the value after reset is returned. When written, write the value after reset.
7	ISMn GIPCLAD	0: No function 1: Clears the status flag ISMnGIP.ISMnGIPSSAD
6	ISMn GIPCLDO	0: No function 1: Clears the status flag ISMnGIP.ISMnGIPSSDO
5 to 0	ISMn GIPCLREm	0: No function 1: Clears the status flag ISMnGIP.ISMnGIPSSREm

46.5.2.7 ISMnCCMRm – ISM channel control register

This register specifies the operation mode for motor channel m.

- In motor driving operation mode, it is used to specify the delay for the PWM signal output. When channel management is set for motor channel m, this register is also used to specify the PWM value look-up table to be used and the motor channel's precision.
  Refer to Section 46.4.5.3, Output delay and Section 46.4.6.1, Stepper motor driving with channel management for details
- In ZPD operation mode, it is used to specify the following:
  - The delay for the PWM signal output when direct control is not used
  - The ZPD table to be used and the maximum table row number to be considered when channel management is set for motor channel m.

  Refer to Section (1), Processing of ZPD table for details.

Access: This register can be
- read/written in 32-bit units when channel management is switched off (ISMnGCTL.ISMnGGCE = 0).
- read only in 32-bit units when channel management is switched on (ISMnGCTL.ISMnGGCE = 1).

Address: <ISMn_base> + $001C_H$ + m x 4_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	ISMnCCTLm[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISMnCCDVm[3:0]				ISMnCCDHm[3:0]				0	ISMn CCZPm	ISMn CCCPm	0	0	ISMnCCCTm[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R	R	R/W	R/W	R/W

Table 46.26 ISMnCCMRm register contents (1/2)

Bit position	Bit name	Function
31 to 23	Reserved	When read, the value after reset is returned. When written, write the value after reset.
22 to 16	ISMn CCTLm[6:0]	Specifies the number of rows of the assigned ZPD table to be considered:

ISMnCCTL[6:0]	Number of valid rows in ZPD table
0	1
1	2
...	...
127	128

Table 46.26 ISMnCCMRm register contents (2/2)

Bit position	Bit name	Function
15 to 12	ISMn CCDVm[3:0]	Specifies the output delay for the *vertical* PWM signal of channel m: ISMn CCDV[3:0] / Output delay 0 / No output delay 1 / 1 cycle of TB_CLK ... / ... 15 / 15 cycles of TB_CLK
11 to 8	ISMn CCDHm[3:0]	Specifies the output delay for the *horizontal* PWM signal of channel m: ISMn CCDH[3:0] / Output delay 0 / No output delay 1 / 1 cycle of TB_CLK ... / ... 15 / 15 cycles of TB_CLK
7	Reserved	When read, the value after reset is returned. When written, write the value after reset.
6	ISMn CCZPm	Specifies the channel's operation mode: 0: Motor driving mode (generation of PWM signal) 1: ZPD mode
5	ISMnCCCPm	Specifies the motor channel's precision: 0: Standard precision (128 microsteps within a 360° turn) 1: High precision (512 microsteps within a 360° turn)
4 to 3	Reserved	When read, the value after reset is returned. When written, write the value after reset.
2 to 0	ISMn CCCTm[2:0]	Specifies which ZPD/PWM value look-up table is used for channel m: ISMn CCCTm[2:0] / RAM table 000 / 0 001 / 1 ... / ... 101 / 5 Other values than 0 to 5 are not allowed.

46.5.2.8 ISMnCCMPm – ISM channel PWM setting register

This register specifies the quadrant information and the PWM signal pulse widths for the sine and cosine side of the desired rotor position.

Refer to Section 46.4.5.2, Duty cycle time and duty factor of PWM signals and Section 46.4.5.4, Encoding of PWM signals according to quadrant for details.

Access: This register can be read/written in 32-bit units.
When channel management is set for a motor channel and channel management is switched on (ISMnGSTR.ISMnCCENm = 1 and ISMnGCTL.ISMnGGCE = 1), the values written by the application are ignored. Channel management supplies the channel's values from the assigned RAM table (ISMnCCMRm.ISMnCCCTm[2:0]).

Address: <ISMn_base> + $004C_H$ + m x 4_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	ISMn CCQIm[1:0]		ISMnCCHPm[9:6]			
R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISMnCCHPm[5:0]						ISMnCCVPm[9:0]									
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.27 ISMnCCMPm register contents

Bit position	Bit name	Function
31 to 22	Reserved	When read, the value after reset is returned. When written, write the value after reset.
21 to 20	ISMn CCQIm[1:0]	Specifies the quadrant 00_B: Quadrant 1 (0° - 90° angle) 01_B: Quadrant 2 (90° - 180° angle) 10_B: Quadrant 3 (180° - 270° angle) 11_B: Quadrant 4 (270° - 360° angle)
19 to 10	ISMn CCHPm[9:0]	Specifies the duty cycle time of the channel's *horizontal* PWM signal as a number of cycles of TB_CLK. The value range is from 0 (PWM signal switched off) to 2^{10} - 1 (duty factor = 100%).
9 to 0	ISMn CCVPm[9:0]	Specifies the duty cycle time of the channel's *vertical* PWM signal as a number of cycles of TB_CLK. The value range is from 0 (PWM signal switched off) to 2^{10} - 1 (duty factor = 100%).

46.5.2.9 ISMnCIOCm – ISM channel I/O control setting register

This register is used to enable/disable the recirculation of PWM signals and to configure direct control (which is only possible in ZPD operation mode).

Refer to Section 46.4.5.5, Recirculation and Section (2), Direct control for details.

Access: This register can be read/written in 32-bit units.
When channel management is set for a motor channel and channel management is switched on (ISMnGSTR.ISMnCCENm = 1 and ISMnGCTL.ISMnGGCE = 1), the values written by the application are ignored. Channel management supplies the channel's values from the assigned RAM table (ISMnCCMRm.ISMnCCCTm[2:0]).

Address: <ISMn_base> + $007C_H$ + m x 4_H

Initial Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	ISMn CIHEm	ISMn CIVEm	ISMn CIHRm	ISMn CIVRm	0	0	0	0	0	0
R	R	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	ISMnCIHDm[3:0]				0	0	0	0	0	0	ISMnCIVDm[3:0]			
R	R	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 46.28 ISMnCIOCm register contents

Bit position	Bit name	Function
31 to 26	Reserved	When read, the value after reset is returned. When written, write the value after reset.
25	ISMnCIHEm	Enables/disables direct control for the *horizontal* PWM signal: 0: Direct control disabled 1: Direct control enabled ISMnCIHDm[3:0] defines which transistors will be activated to supply the *horizontal* coil with current.
24	ISMnCIVEm	Enables/disables direct control for the *vertical* PWM signal: 0: Direct control disabled 1: Direct control enabled ISMnCIVDm[3:0] defines which transistors will be activated to supply the *vertical* coil with current.
23	ISMnCIHRm	Enables/disables recirculation for the *horizontal* PWM signal: 0: Recirculation disabled 1: Recirculation enabled
22	ISMnCIVRm	Enables/disables recirculation for the *vertical* PWM signal: 0: Recirculation disabled 1: Recirculation enabled
21 to 14	Reserved	When read, the value after reset is returned. When written, write the value after reset.
13 to 10	ISMn CIHDm[3:0]	Defines which transistors will be activated to supply the *horizontal* coil with current when direct control is enabled (ISMnCIHEm = 1).
9 to 4	Reserved	When read, the value after reset is returned. When written, write the value after reset.
3 to 0	ISMn CIVDm[3:0]	Defines which transistors will be activated to supply the *vertical* coil with current when direct control is enabled (ISMnCIVEm = 1).

NOTE

Direct control is bound to ZPD operation mode (ISMnCCMR.ISMnCCZPm = 1).

CAUTION

Using the ISMn with inductive loads (such as the coils of the stepper motors) the following must be considered:

- **Recirculation must be enabled (ISMnCIVRm and ISMnCIHRm set). Otherwise, severe damage to the output stages of the I/O ports may result.**
- **The ZPD tables in the RAM must be defined by the application such that the discharge of any inductive load is performed, before using direct control to change to a high-impedance state (at least three transistors off). Otherwise, severe damage to the output stages of the I/O ports may result.**

46.5.2.10 ISMnCOPT – ISM channels recirculation side setting register

This register specifies the connection to which the current is directed for recirculation.

Refer to Section 46.4.5.5, Recirculation for details.

Access: This register can be read/written in 32-bit units when channel management is switched off (ISMnGCTL.ISMnGGCE = 0).

Address: <ISMn_base> + 0094_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	ISMn CIRS5	ISMn CIRS4	ISMn CIRS3	ISMn CIRS2	ISMn CIRS1	ISMn CIRS0
R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.29 ISMnCOPT register contents

Bit position	Bit name	Function
31 to 6	Reserved	When read, the value after reset is returned. When written, write the value after reset.
5 to 0	ISMnCIRSm	Specifies the connection to which the current is directed for recirculation: 0: Redirection to the ground connection $ISMV_{SS}$ 1: Redirection to the power supply $ISMV_{CC}$

46.5.3 ISMn movement calculator parameters

The movement calculator parameters are defined via a set of 10 registers: ISMnPARiCFGm with i = 0 to 9.

The bit names of these registers include an indicator “Pxx”, that points to the purpose of the parameter.

46.5.3.1 ISMnPAR0CFGm – Target motor position CHm

This register specifies the target motor position when motor channel m is operated in motor driving operation mode and all values for intermediate motor steps are supplied by channel management (ISMnGSTR.ISMnCCENm = 1 and ISMnGCTL.ISMnGGCE = 1).

Refer to Section 46.4.6.1, Stepper motor driving with channel management for details.

Bit name indicator

Pxx = PMP

Access: This register can be read/written in 32-bit units.

Address: <ISMn_base> + $00C8_H$ +m x 28_H

Initial Value: 0000 0000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	ISMn SPMPm	ISMnIPMPm[15:8]							
R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISMnIPMPm[7:0]								ISMnFPMPm[7:0]*1							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. ISMFPMPm[7:0] must be set to FF_H.

Table 46.30 ISMnPAR0CFGm register contents

Bit position	Bit name	Function
31 to 25	Reserved	When read, the value after reset is returned. When written, write the value after reset.
24	ISMn SPMPm	Sign of parameter
23 to 8	ISMn IPMPm[15:0]	Integer part of parameter
7 to 0	ISMn FPMPm[7:0]	Fractional part of parameter **CAUTION** The fractional part must be set to FF_H.

46.5.3.2 ISMnPAR1CFGm – Damping factor CHm

This register specifies the damping factor of the movement calculation algorithm.

Bit name indicator

Pxx = PDF

Access: This register can be read/written in 32-bit units when channel management is disabled for that channel (ISMnGSTR.ISMnCCENm = 0).
This register is read-only when channel management is enabled (ISMnGSTR.ISMnCCENm = 1 and ISMnGCTL.ISMnGGCE = 1).

Address: <ISMn_base> + $00CC_H$ +m x 28_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	ISMnFPDFm[2:0]		
R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Table 46.31 ISMnPAR1CFGm register contents

Bit position	Bit name	Function
31 to 3	Reserved	When read, the value after reset is returned. When written, write the value after reset.
2 to 0	ISMn FPDFm[2:0]	Represents the inertia PDF of the algorithm. 2^{PDF} divides the theoretical maximum speed required to reach the target position within only one algorithm pass.

46.5.3.3 ISMnPARiCFGm for i = 2 to 9 – Movement parameters CHm

ISMnPARiCFGm specify the movement parameters for the automatic calculation of rotor positions for channel m.

The values in this register are required when motor channel m is operated in motor driving operation mode and all values for intermediate motor steps are supplied by channel management (ISMnGSTR.ISMnCCENm = 1 and ISMnGCTL.ISMnGGCE = 1).

All registers have the same binary-fixed-point storage format and are therefore described together in this section.

Table 46.32 ISMnPARiCFGm parameter registers overview

Register	Parameter	Indicator in bit name (Pxx)	Address
ISMnPAR2CFGm	Acceleration limit CHm	PAL	<ISMn_base> + $00D0_H$ + m x 28_H
ISMnPAR3CFGm	Deceleration limit CHm	PDL	<ISMn_base> + $00D4_H$ + m x 28_H
ISMnPAR4CFGm	Maximum speed CHm	PMS	<ISMn_base> + $00D8_H$ + m x 28_H
ISMnPAR5CFGm	Hysteresis correction CHm	PHC	<ISMn_base> + $00DC_H$ + m x 28_H
ISMnPAR6CFGm	Speed threshold 1 CHm	PS1	<ISMn_base> + $00E0_H$ + m x 28_H
ISMnPAR7CFGm	Speed threshold 2 CHm	PS2	<ISMn_base> + $00E4_H$ + m x 28_H
ISMnPAR8CFGm	Speed threshold 3 CHm	PS3	<ISMn_base> + $00E8_H$ + m x 28_H
ISMnPAR9CFGm	Speed threshold 4 CHm	PS4	<ISMn_base> + $00EC_H$ + m x 28_H

Bit name indicator

See Table 46.32, ISMnPARiCFGm parameter registers overview.

Access: These registers can be read/written in 32-bit units when channel management is disabled for that channel (ISMnGSTR.ISMnCCENm = 0).
These register are read-only when channel management is enabled (ISMnGSTR.ISMnCCENm = 1 and ISMnGCTL.ISMnGGCE = 1).

Address: See Table 46.32, ISMnPARiCFGm parameter registers overview.

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	ISMn SPxxm	ISMnIPxxm[15:8]							
R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISMnIPxxm[7:0]								ISMnFPxxm[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.33 ISMnPARiCFGm register contents

Bit position	Bit name	Function
31 to 25	Reserved	When read, the value after reset is returned. When written, write the value after reset.
24	ISMn SPxxm	Sign of parameter
23 to 8	ISMn IPxxm[15:0]	Integer part of parameter
7 to 0	ISMn FPxxm[7:0]	Fractional part of parameter

46.5.4 ISMn movement calculator variables

46.5.4.1 ISMnVARjCFGm for j = 0 to 5 – Movement variables CHm

ISMn stores the movement variables for the automatic calculation of rotor positions for channel m (when ISMnGSTR.ISMnCCENm = 1 and ISMnGCTL.ISMnGGCE = 1) in ISMnVARjCFGm.

These registers are used when motor channel m is operated in motor driving operation mode and all values for intermediate motor steps are supplied by channel management (ISMnGSTR.ISMnCCENm = 1 and ISMnGCTL.ISMnGGCE = 1).

All registers have the same binary-fixed-point storage format and are therefore described together in this section.

NOTE

These registers should not be modified during normal operation of ISMn. The values can be modified for test purposes in debug mode (ISMnEMU.ISMnSVDIS = 1).

Table 46.34 ISMnVARjCFGm variable registers overview

Register	Variable	Indicator in bit name (Vxx)	Address
ISMnVAR0CFGm	Actual acceleration/deceleration CHm	VAX	<ISMn_base> + $01B8_H$ + m x $1C_H$
ISMnVAR1CFGm	Actual speed CHm	VAS	<ISMn_base> + $01BC_H$ + m x $1C_H$
ISMnVAR2CFGm	Previous iteration speed CHm	VPS	<ISMn_base> + $01C0_H$ + m x $1C_H$
ISMnVAR3CFGm	Temporary calculator register PT1	VPT	<ISMn_base> + $01C4_H$ + m x $1C_H$
ISMnVAR4CFGm	Actual position, current result CHm	VAP	<ISMn_base> + $01C8_H$ + m x $1C_H$
ISMnVAR5CFGm	Virtually displayed position CHm	VVP	<ISMn_base> + $01CC_H$ + m x $1C_H$

Access: ISMnVAR[2:5]CFGm registers can be read/written in 32-bit units when channel management is disabled for that channel (ISMnGSTR.ISMnCCENm = 0).
ISMnVAR[2:5]CFGm are read-only when channel management is enabled (ISMnGSTR.ISMnCCENm = 1 and ISMnGCTL.ISMnGGCE = 1).
ISMnVAR[0:1]CFGm are read-only.
Note that old values may be read while the sequencer is busy updating channels' data (ISMnGSTR.ISMnGGCB = 1).

Address: See Table 46.34, ISMnVARjCFGm variable registers overview.

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	ISMnSVxxm	ISMnIVxxm[15:8]							
R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISMnIVxxm[7:0]								ISMnFVxxm[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.35 ISMnVARjCFGm register contents (1/2)

Bit position	Bit name	Function
31 to 25	Reserved	When read, the value after reset is returned. When written, write the value after reset.

Table 46.35 ISMnVARjCFGm register contents (2/2)

Bit position	Bit name	Function
24	ISMn SVxxm	Sign of variable
23 to 8	ISMn IVxxm[15:0]	Integer part of variable
7 to 0	ISMn FVxxm[7:0]	Fractional part of variable

46.5.4.2 ISMnVAR6CFGm – Flag variables CHm

ISMn stores speed and direction flags in ISMnVAR6CFGm when the rotor's movements are controlled by channel management (ISMnGSTR.ISMnCCENm = 1 and ISMnGCTL.ISMnGGCE = 1) and standard precision is configured (ISMnCCMRm.ISMnCCCPm = 0).

Refer to Section 46.4.8.1, PWM value look-up tables for motor driving for details.

NOTE

This register should not be modified during normal operation of ISMn. The values can be modified for test purposes in debug mode (ISMnEMU.ISMnSVDIS = 1).

Access: This register can be read/written in 32-bit units when channel management is disabled for that channel (ISMnGSTR.ISMnCCENm = 0).
This register is read-only when channel management is enabled (ISMnGSTR.ISMnCCENm = 1 and ISMnGCTL.ISMnGGCE = 1). Note that old values may be read while the sequencer is busy updating channels' data (ISMnGSTR.ISMnGGCB = 1).

Address: <ISMn_base> + $01D0_H$ + m x $1C_H$

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	ISMn VVDRm	ISMn VVSPm
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 46.36 ISMnVAR6CFGm register contents

Bit position	Bit name	Function
31 to 2	Reserved	When read, the value after reset is returned. When written, write the value after reset.
1	ISMnVVDRm	0: Channel management is using part for speed 0 on the PWM value look-up table. 1: Channel management is using part for speed 1 on the PWM value look-up table.
0	ISMnVVSPm	0: Channel management is using part for direction 0 on the PWM value look-up table. 1: Channel management is using part for direction 1 on the PWM value look-up table.

46.5.5 ZPD registers

46.5.5.1 ISMnGZPDCTL – ISM global ZPD control register

This register is used to enable zero point detection and to specify the general settings for ZPD (such as measurement clock M_CLK, reference voltages, digital noise filtering, etc.).

Access: This register can be read/written in 32-bit units when channel management is disabled (ISMnGSTR.ISMnCCENm = 0 or ISMnGCTL.ISMnGGCE = 0).
This register is read-only when channel management is enabled (ISMnGSTR.ISMnCCENm = 1 and ISMnGCTL.ISMnGGCE = 1).

Address: <ISMn_base> + 0010_H

Initial Value: $4033\ FF7F_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISMn GGZE	ISMn GGZP	0	0	0	0	0	0	ISMnGGFL[3:0]				ISMnGGFD[3:0]			
R/W	R/W	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISMnGGRV2[3:0]				ISMnGGRV1[3:0]				0	ISMnGGZF[3:0]				ISMnGGCS[2:0]		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.37 ISMnGZPDCTL register contents (1/4)

Bit position	Bit name	Function
31	ISMnGGZE	Enables/disables the ZPD unit: 0: Disabled 1: Enabled ZPD measurements are only possible when the ZPD unit is enabled. **CAUTION** When ZPD is enabled, the default value of bit 0 of ISMnGCTL has to changed to “1”.
30	ISMnGGZP	Enables/disables power saving mode: 0: power save mode disabled 1: power save mode enabled The power saving mode has to be disabled during ZPD measurements. Otherwise the ZPD measurement results are invalid. Refer to Section 46.4.7.6, Power save mode for details.
29 to 24	Reserved	When read, the value after reset is returned. When written, write the value after reset.
23 to 20	ISMnGGFL[3:0]	Specifies the minimum time the voltage of the selected input has to be above the channel’s reference voltage during one measurement sequence at which the motor is assumed to be not at its zero position. The value is specified as a number of clock cycles of M_CLK. **ISMn GGFL[3:0] / Minimum no. of clock cycles of M_CLK** 0 / 1 1 / 2 ... / ... 15 / 16 Refer to Section (6), Noise filtering for details.

Table 46.37 ISMnGZPDCTL register contents (2/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>19 to 16</td><td>ISMn GGFD[3:0]</td><td>Specifies how often the voltage is measured during one measurement sequence.
The value is specified as a number of clock cycles of M_CLK.
<table>
<tr><th>ISMn GGFD[3:0]</th><th>No. of clock cycles of M_CLK</th></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>2</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>15</td><td>16</td></tr>
</table>
Refer to Section (6), Noise filtering for details.</td></tr>
<tr><td>15 to 12</td><td>ISMn GGRV2[3:0]</td><td>Specifies the level of the reference voltage the level of the selected input signal can be compared with:
<table>
<tr><th>ISMnGGRV2[3:0]</th><th>Voltage</th></tr>
<tr><td>0</td><td>215 mV</td></tr>
<tr><td>1</td><td>225 mV</td></tr>
<tr><td>2</td><td>235 mV</td></tr>
<tr><td>3</td><td>245 mV</td></tr>
<tr><td>4</td><td>470 mV</td></tr>
<tr><td>5</td><td>480 mV</td></tr>
<tr><td>6</td><td>490 mV</td></tr>
<tr><td>7</td><td>500 mV</td></tr>
<tr><td>8</td><td>150 mV</td></tr>
<tr><td>9</td><td>350 mV</td></tr>
<tr><td>10</td><td>450 mV</td></tr>
<tr><td>11</td><td>550 mV</td></tr>
<tr><td>12</td><td>650 mV</td></tr>
<tr><td>13</td><td>750 mV</td></tr>
<tr><td>14</td><td>850 mV</td></tr>
<tr><td>15</td><td>External VREF x 0.2</td></tr>
</table>
The reference voltage is used that is configured in ISMnZPDOPT.ISMnCZRSm.
Refer to Section (5), Reference voltages for details.</td></tr>
</table>

Table 46.37 ISMnGZPDCTL register contents (3/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>11 to 8</td><td>ISMn GGRV1[3:0]</td><td>Specifies the level of the reference voltage the level of the selected input signal can be compared with:
<table>
<tr><th>ISMnGGRV1[3:0]</th><th>Voltage</th></tr>
<tr><td>0</td><td>215 mV</td></tr>
<tr><td>1</td><td>230 mV</td></tr>
<tr><td>2</td><td>235 mV</td></tr>
<tr><td>3</td><td>245 mV</td></tr>
<tr><td>4</td><td>450 mV</td></tr>
<tr><td>5</td><td>480 mV</td></tr>
<tr><td>6</td><td>500 mV</td></tr>
<tr><td>7</td><td>550 mV</td></tr>
<tr><td>8</td><td>100 mV</td></tr>
<tr><td>9</td><td>150 mV</td></tr>
<tr><td>10</td><td>250 mV</td></tr>
<tr><td>11</td><td>350 mV</td></tr>
<tr><td>12</td><td>650 mV</td></tr>
<tr><td>13</td><td>750 mV</td></tr>
<tr><td>14</td><td>850 mV</td></tr>
<tr><td>15</td><td>External VREF x 0.2</td></tr>
</table>
The reference voltage is used that is configured in ISMnZPDOPT.ISMnCZRSm. Refer to Section (5), Reference voltages for details.</td></tr>
</table>

Table 46.37 ISMnGZPDCTL register contents (4/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>6 to 3</td><td>ISMn GGZF[3:0]</td><td>Specifies the PCLK prescaler for defining M_CLK:
<table>
<tr><th>ISMn GGZF[3:0]</th><th>Prescaler</th></tr>
<tr><td>0</td><td>PCLK / 4</td></tr>
<tr><td>1</td><td>PCLK / 5</td></tr>
<tr><td>2</td><td>PCLK / 6</td></tr>
<tr><td>3</td><td>PCLK / 7</td></tr>
<tr><td>4</td><td>PCLK / 8</td></tr>
<tr><td>5</td><td>PCLK / 9</td></tr>
<tr><td>6</td><td>PCLK / 10</td></tr>
<tr><td>7</td><td>PCLK / 11</td></tr>
<tr><td>8</td><td>PCLK / 12</td></tr>
<tr><td>9</td><td>PCLK / 16</td></tr>
<tr><td>10</td><td>PCLK / 32</td></tr>
<tr><td>11</td><td>PCLK / 64</td></tr>
<tr><td>12</td><td>PCLK / 128</td></tr>
<tr><td>13</td><td>PCLK / 256</td></tr>
<tr><td>14</td><td>PCLK / 512</td></tr>
<tr><td>15</td><td>PCLK / 1024</td></tr>
</table>
Refer to Section (1), ZPD measurement clock M_CLK for details.</td></tr>
<tr><td>2 to 0</td><td>ISMn GGCS[2:0]</td><td>Specifies the measurement delay within one measurement sequence. The value is specified as a number of clock cycles of the measurement clock M_CLK.
<table>
<tr><th>ISMn GGCS[2:0]</th><th>Clock cycles</th></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>2</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>7</td><td>8</td></tr>
</table>
Refer to Section (4), Measurement delay for details.</td></tr>
</table>

46.5.5.2 ISMnCCNTm – ISM channel ZPD counter register

This register can be used to read the ZPD table index counters for all motor channels.

Refer to Section (1), Processing of ZPD table for details.

Access: This register can be read in 32-bit units.

Address: <ISMn_base> + 0034_H + m x 4_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	ISMnCCZCm[6:0]						
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 46.38 ISMnCCNTm register contents

Bit position	Bit name	Function
31 to 7	Reserved	When read, the value after reset is returned.
6 to 0	ISMn CCZCm[6:0]	ZPD table index counter 0_H: Beginning of ZPD table $7F_H$: End of ZPD table The table index counter does not show the row number of the current table row but the number of the *next* row. Counting restarts from 0 when the counter reaches the limit specified in ISMnCCMR.ISMnCCTLm[6:0].

46.5.5.3 ISMnCZCFGm – ISM channel ZPD configuration register

This register can be used to read when the next table row will be fetched from the ZPD table that is assigned to channel m.

Refer to Section (4), Measurement delay for details.

Access: This register can be read in 32-bit units.

Address: <ISMn_base> + 0064_H + m x 4_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	ISMnCCZDm[4:0]				
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 46.39 ISMnCZCFGm register contents

Bit position	Bit name	Function
31 to 5	Reserved	When read, the value after reset is returned.
4 to 0	ISMn CCZDm[4:0]	Number of START events that occur before the next table row is fetched from the assigned ZPD table. The value is decremented periodically when a START event occurs. The minimum value is 0 and the maximum value is 31.

46.5.5.4 ISMnZPDCTL – ISM channels ZPD info bus settings register

This register is used to start ZPD measurements. It also specifies the input signal to be measured.

Access: This register can be read/written in 32-bit units when channel management is switched off (ISMnGCTL.ISMnGGCE = 0).
This register is read-only when channel management is switched on (ISMnGCTL.ISMnGGCE = 1).

Address: <ISMn_base> + 0098_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	ISMn CZIS5[1:0]		0	0	ISMn CZIS4[1:0]		0	0	ISMn CZIS3[1:0]		0	0	ISMn CZIS2[1:0]	
R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	ISMn CZIS1[1:0]		0	0	ISMn CZIS0[1:0]		0	0	ISMn CZMT5	ISMn CZMT4	ISMn CZMT3	ISMn CZMT2	ISMn CZMT1	ISMn CZMT0
R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.40 ISMnZPDCTL register contents

Bit position	Bit name	Function
29 to 28	ISMnCZIS5[1:0]	Specifies the input signal to be measured for ZPD:
25 to 24	ISMnCZIS4[1:0]	
21 to 20	ISMnCZIS3[1:0]	
17 to 16	ISMnCZIS2[1:0]	
13 to 12	ISMnCZIS1[1:0]	
9 to 8	ISMnCZIS0[1:0]	

ISMnCZISm	Signal
00_B	SINPm
01_B	COSPm
10_B	SINMm
11_B	COSMm

5 to 0	ISMnCZMTm	ZPD measurement trigger for channel m: 0: Do not perform ZPD measurement 1: Perform ZPD measurement First measurement starts after the blanking delay defined in ISMnZPDCMP.ISMnCZBTm[15:0].

NOTE

When channel management is set, ISMnZPDCTL.ISMnCZISm[1:0] and ISMnCZMTm are taken from the assigned ZPD table. Otherwise, ISMnZPDCTL.ISMnCZISm[1:0] and ISMnCZMTm need to be supplied by the application.

46.5.5.5 ISMnZPDOPT – ISM channels ZPD options setting register

This register specifies which of the two reference voltages the level of a channel's input signal is compared with. For each channel, one of the two reference voltages can be selected.

Refer to Section (5), Reference voltages for details.

Access: This register can be read/written in 32-bit units when channel management is switched off (ISMnGCTL.ISMnGGCE = 0).
This register is read-only when channel management is switched on (ISMnGCTL.ISMnGGCE = 1).

Address: <ISMn_base> + $009C_H$

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	ISMn CZRS5	ISMn CZRS4	ISMn CZRS3	ISMn CZRS2	ISMn CZRS1	ISMn CZRS0
R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.41 ISMnZPDOPT register contents

Bit position	Bit name	Function
31 to 6	Reserved	When read, the value after reset is returned. When written, write the value after reset.
5 to 0	ISMnCZRSm	Specifies which of the two reference voltages is used for channel m: 0: Use reference voltage defined by ISMnGZPDCTL.ISMnGGRV1[3:0] 1: Use reference voltage defined by ISMnGZPDCTL.ISMnGGRV2[3:0]

46.5.5.6 ISMnZPDSTR – ISM channels ZPD status register

This register can be used to verify the last noise filtering result for all channels.

Refer to Section (6), Noise filtering and Section (7), Noise filtering result for details.

Access: This register can be read in 32-bit units.

Address: <ISMn_base> + $00A0_H$

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	ISMn CZDR5	ISMn CZDR4	ISMn CZDR3	ISMn CZDR2	ISMn CZDR1	ISMn CZDR0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 46.42 ISMnZPDSTR register contents

Bit position	Bit name	Function
31 to 6	Reserved	When read, the value after reset is returned.
5 to 0	ISMnCZDRm	Indicates whether – according to noise filtering – the level of the input signal is above or below the level of the reference voltage: 0: Voltage is *below* reference voltage (yet unconfirmed zero position) 1: Voltage is *above* reference voltage (no zero point detected so far)

46.5.5.7 ISMnZPDIP – ISM channels ZPD detection flag register

This register indicates for all channels whether their motors have reached their zero positions since the detection flag in this register was last cleared.

Refer to Section (7), Noise filtering result for details.

Access: This register can be read in 32-bit units.

Address: <ISMn_base> + $00A4_H$

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	ISMn CZIP5	ISMn CZIP4	ISMn CZIP3	ISMn CZIP2	ISMn CZIP1	ISMn CZIP0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 46.43 ISMnZPDIP register contents

Bit position	Bit name	Function
31 to 6	Reserved	When read, the value after reset is returned.
5 to 0	ISMnCZIPm	Indicates the status of the interrupt IRQ_ZPD that indicates whether a motor has reached its zero position or not: 0: No IRQ_ZPD pending (voltage has not been above the configured reference voltage since this bit was last cleared) 1: IRQ_ZPD has been generated (voltage was at least once above the configured reference voltage) This bit is automatically cleared, when a new set of ZPD measurements is started (ZPD measurement trigger ISMnZPDCTL.ISMnCZMTm changes from 0 to 1. This bit can be cleared by the application by writing 1 to ISMnZPDSTC.ISMnCCLZPm.

46.5.5.8 ISMnZPDSTC – ISM channels ZPD detection flag clear register

This register is the clear control register of ISMnZPDIP.

Access: This register can be written in 32-bit units.

Address: <ISMn_base> + $00A8_H$

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	ISMn CCLZP5	ISMn CCLZP4	ISMn CCLZP3	ISMn CCLZP2	ISMn CCLZP1	ISMn CCLZP0
R	R	R	R	R	R	R	R	R	R	W	W	W	W	W	W

Table 46.44 ISMnZPDSTC register contents

Bit position	Bit name	Function
31 to 6	Reserved	When read, the value after reset is returned.
5 to 0	ISMnCCLZPm	0: No function 1: Clears the status flag ISMnZPDIP.ISMnCZIPm that indicates a pending IRQ_ZPD interrupt.

46.5.5.9 ISMnZPDCMPm – ISM channel ZPD blanking and vibration damping setting register

This register specifies the blanking delay of ZPD measurements and whether the ZPD vibration damping feature is used.

Refer to Section (3), Blanking delay and Section (10), ZPD vibration damping for details.

Access: This register can be read/written in 32-bit units when channel management is switched off (ISMnGCTL.ISMnGGCE = 0).
This register is read-only when channel management is switched on (ISMnGCTL.ISMnGGCE = 1).

Address: <ISMn_base> + $00AC_H$ + m x 4_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISMn CZSSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISMnCZBTm[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.45 ISMnZPDCMPm register contents

Bit position	Bit name	Function
31	ISMn CZSSm	Specifies whether the ZPD vibration damping feature is used or not. 0: ZPD vibration damping disabled 1: ZPD vibration damping enabled If during a measurement sequence, it is detected that the motor has not reached its zero position yet, the coil is short-circuited for the rest of the measurement sequence.
30 to 16	Reserved	When read, the value after reset is returned. When written, write the value after reset.
15 to 0	ISMn CZBTm[15:0]	Specifies the blanking delay as a number of measurement sequences T_m. The minimum value is 0 and the maximum value is $FFFF_H$.

46.5.5.10 ISMnZPDCSTR – ISM channels ZPD measurement activity status register

This register contains the ZPD current measurement activity status.

Access: This register can be read in 32-bit units.

Address: <ISMn_base> + $00C4_H$

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	ISMnCZAF5	0ISMnCZAF4	ISMnCZAF3	ISMnCZAF2	ISMnCZAF1	ISMnCZAF0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 46.46 ISMnZPDCSTR register contents

Bit position	Bit name	Function
31 to 6	Reserved	When read, the value after reset is returned.
5	ISMnCZAF5	ZPD current measurement activity status for channel 5. 0: ZPD measurement inactive. 1: ZPD measurement active.
4	ISMnCZAF4	ZPD current measurement activity status for channel 4. 0: ZPD measurement inactive. 1: ZPD measurement active.
3	ISMnCZAF3	ZPD current measurement activity status for channel 3. 0: ZPD measurement inactive. 1: ZPD measurement active.
2	ISMnCZAF2	ZPD current measurement activity status for channel 2. 0: ZPD measurement inactive. 1: ZPD measurement active.
1	ISMnCZAF1	ZPD current measurement activity status for channel 1. 0: ZPD measurement inactive. 1: ZPD measurement active.
0	ISMnCZAF0	ZPD current measurement activity status for channel 0. 0: ZPD measurement inactive. 1: ZPD measurement active.

46.5.6 ISMn emulation mode register

46.5.6.1 ISMnEMU – Emulation register

This register controls whether the ISMn module can be stopped during emulation, for instance when a breakpoint is reached.

Access: This register can be read/written in 32-bit units.

Address: <ISMn_base> + 0260_H

Initial Value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	ISMn SVDIS	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R

Table 46.47 ISMnEMU register contents

Bit position	Bit name	Function
31 to 8	Reserved	When read, the value after reset is returned.
7	ISMnSVDIS	Emulation control: 0: ISMn module can be stopped during emulation When the EPC.SVSTOP bit is set to 1: – The ISM sequencer continues with an already started processing of all enabled channels. A new trigger (START) is ignored. – PWM generation continues (no stop of PWM generation). – Changing of channel parameters by APB interface is still possible. When the EPC.SVSTOP bit is set to 0: – The ISM continues execution. 1: ISMn module operates continuously during emulation The ISM continues execution on a debug break. (EPC.SVSTOP is ignored).
6 to 0	Reserved	When read, the value after reset is returned.

46.6 Detection and Correction of Errors in ISM RAM

46.6.1 ECC for the ISM RAM

Table 46.48 gives an outline of the ECC functions for the ISM RAM.

Table 46.48 List of the ECC Functions for the ISM RAM

Item	Outline of Functions
ECC error detection/correction	The RAM is checked for ECC errors. The following options are selectable. • 2-bit error detection and 1-bit error detection/correction • 2-bit error detection and 1-bit error detection The ECC error detection/correction can be disabled by using through mode. With the initial settings, error detection/correction is enabled. The 7-bit ECC value is calculated for each 32-bit transfer.
Error notification	When an ECC 2-bit error is generated, the error is notified. • Error notification can be enabled or disabled when an ECC 2-bit error is detected. In the initial setting, 2-bit error notification is enabled.
Error status	Monitoring for the detection of two-bit ECC errors and for the detection of one-bit ECC errors is available. A register for clearing the error status is provided.
Address capture	• Only one address at which an ECC error has occurred can be captured. • A signal is generated upon detection of a 2-bit or ECC 1-bit error, and the signal is used as a trigger to capture the error-causing address (when the first (1-bit or 2-bit) error is detected after the flag is cleared).

CAUTION

When ECC error detection/correction is performed, initialize the ISM RAM by the ISM module before it is used.

46.6.2 Interrupt Request

Table 46.49 lists the ECC interrupt request of ISM RAM.

Table 46.49 Interrupt Requests

Unit Interrupt Signal	Outline	Connected to
—	ISM RAM ECC 2 bit error interrupt	Error Control Module INTECCDPERIRAM
—	ISM RAM ECC 1 bit error interrupt	Error Control Module INTECCSPERIRAM

46.6.3 ECCISMnCTL — ISMn RAM ECC Control Register

The ECCISMnCTL register controls the mode of the ECC and the status for ISMn.

Bits 7, 5 and 4 should be set (written) while the ISMn operation is stopped.

In addition, when writing to bit 7, EMCA1 and EMCA0 need to be 01_B.

Access: This register can be read/written in 16-bit units.

Address: FFC7 8000_H

Value after reset: Undefined

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA1	EMCA0	—	—	—	EC ER2C	EC ER1C	—	EC THM	ECERV F	EC 1ECP	EC 2EDIC	EC1EDI C	EC ER2F	EC ER1F	EC EMF
Value after reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	Undefined
R/W	R/W*1	R/W*1	R	R	R	R/W*1	R/W*1	R	R/W	R/W	R/W	R/W	R/W	R	R	R

Note 1. These bits are always read as 0.

Table 46.50 ECCISMnCTL Register Contents (1/2)

Bit position	Bit Name	Function
15	EMCA1	Access control bits 1 and 0 to ECC mode selection bit These bits specify whether modifying the ECTHM bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 01_B, writing to bit 7 is enabled.
14	EMCA0	
13 to 11	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
10	ECER2C	2-bit ECC error detection flag clear bit This bit is used to clear the 2 bit error detection flag of ECER2F (bit 2). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER2F bit is set to clear the ECER2F bit. When a conflict between this bit writing and ECER2F bit setting occurs, writing to this bit has a priority.
9	ECER1C	1-bit ECC error correction accumulation flag clear bit This bit is used to clear the 1 bit error detection/correction flag of ECER1F (bit 1). This bit is always read as 0. Writing 0 is ignored. Write 1 to this bit while the ECER1F bit is set to clear the ECER1F bit. When a conflict between this bit writing and ECER1F bit setting occurs, writing to this bit has a priority.
8	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
7	ECTHM	ECC function through mode selection bit This bit is used to set enabling and disabling of ECC. Setting this bit to 1 disables ECC function. When writing to this bit, (0, 1) must be written to (EMCA1, EMCA0) at the same time. 0: Through mode is disabled (normal operation mode). 1: Through mode is enabled. (ECC function disable)
6	ECERVF	ECC error judgment enable 0: error judgement disabled 1: error judgement enabled Error judgement is only effective in normal operation mode, i.e. if ECCISMnCTL.ECTHM = 0. If error judgement is disabled no interrupts are asserted in case of a single or double bit error detection. However detected single bit errors are corrected. Modification of this bit is only possible, if EMCA[1:0] = 01_B. Otherwise any write to this bit is ignored.

Table 46.50 ECCISMnCTL Register Contents (2/2)

Bit position	Bit Name	Function
5	EC1ECP	1-bit error correction enable bit This bit specifies whether to enable or disable 1-bit error correction when the ECC error detection/correction is enabled. 0: When 1-bit error is detected, the error will be corrected. 1: When 1-bit error is detected, the error will not be corrected.
4	EC2EDIC	2-bit error detection interrupt control bit This bit controls whether to generate an interrupt when 2-bit error is detected. 0: When 2-bit error is detected, a INTECCDPERIRAM interrupt will not be generated. 1: When 2-bit error is detected, a INTECCDPERIRAM interrupt will be generated. (initial value)
3	EC1EDIC	1-bit error detection interrupt control bit This bit controls whether to generate an interrupt when 1-bit error is detected. 0: When 1-bit error is detected, a INTECCSPERIRAM interrupt will not be generated. 1: When 1-bit error is detected, a INTECCSPERIRAM interrupt will be generated.
2	ECER2F	2-bit error detection flag bit This flag indicates whether 2-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). When 2-bit error interrupt is enabled (EC2EDIC = 1) and this flag is set, an ECC 2-bit error interrupt (INTECCDPERIRAM) is output. Write 1 to the ECER2C bit (bit 10) to clear the flag. This bit is cleared at the time of through mode enable selection (ECTHM = 1). If 2-bit error is detected again while this bit is set, an interrupt signal will not be generated. 0: 2-bit error has not occurred since this bit was cleared. 1: 2-bit error has occurred.
1	ECER1F	1-bit error detection/correction flag bit This flag indicates whether 1-bit error is detected during read access to the RAM when error determination is enabled (ECTHM = 0). Write 1 to the ECER1C bit (bit 9) to clear the flag. This bit is cleared at the time of through mode enable selection (ECTHM = 1). 0: 1-bit error has not occurred since this bit was cleared. 1: 1-bit error has occurred.
0	ECEMF	ECC error message flag This flag indicates whether an error exists in the current read data. This bit is updated whenever the RAM reads data. This bit might be set if it is read before initialization of the RAM. This bit is cleared at the time of through mode enable selection (ECTHM = 1) and when there is no 1 bit error in the decode circuit input data. 0: The currently-read RAM data does not have bit errors. 1: The currently-read RAM data have bit errors.

CAUTION

Bits 2 and 1 should be cleared when the ECC error message flag (ECEMF) is not set.

We recommend initializing the RAM before clearing bits 2 and 1.

46.6.4 ECCISMnTMC — ISM RAM ECC Test Mode Control Register

The ECCISMnTMC register switches to and controls the test mode.

This register can be used when ISM is not accessed to RAM.

Access: This register can be read/written in 16-bit units.

Address: FFC7 8004_H

Value after reset: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA1	ETMA0	—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W*1	R/W*1	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Note 1. These bits are always read as 0.

Table 46.51 ECCISMnTMC Register Contents (1/2)

Bit position	Bit Name	Function
15	ETMA1	Access control bits 1 and 0 to ECC test mode bit These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 10_B, writing to bit 7 is enabled.
14	ETMA0	
13 to 8	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
7	ECTMCE	ECC test mode enable bit This bit specifies whether to enable access to the test registers and test control bits. When writing to this bit, (1, 0) should be written to (ETMA1, ETMA0) at the same time. 0: Access to the test mode registers and bits is disabled. 1: Access to the test mode registers and bits is enabled. Test registers: ECCISMnTED, ECCISMnTRC, ECCISMnSYND, ECCISMnHORD, ECCISMnECRD, ECCISMnERDB Register test control bit: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6, 5	Reserved	When read, the value after reset is returned. Writing to these bits, write the value after reset.
4	ECTRRS	ECC RAM read test mode selection bit Select reading destination when reading ECCISMnTED register and reading destination when reading ECCISMnERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0:The read value of the ECCISMnTED register is the write value of the ECCISMnTED register. The read value of the ECCISMnERDB register is the write value of the ECCISMnERDB register. 1: The read value of the ECCISMnTED register can read RAM data. The read value of the ECCISMnERDB register is the ECC data to be written to RAM.
3	ECREOS	ECC redundant bit output data selection bit This bit is used to identify and select the ECC data to be stored in RAM as ECC data generated for write data or the value of the ECCISMnERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Store ECC data generated for write data to RAM. 1: Store the value of ECCISMnERDB register to RAM.

Table 46.51 ECCISMnTMC Register Contents (2/2)

Bit position	Bit Name	Function
2	ECENS	ECC encoder input selection bit This bit is used to identify the targeted data for ECC data generation as write data to RAM or the value of the ECCISMnTED register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate ECC data from the write data to RAM. 1: Generate ECC data from the value of the ECCISMnTED register.
1	ECDCS	ECC decoder input selection bit This bit is used to indentify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the ECCISMnTED register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate syndrome code from RAM data and detect errors. 1: Generate syndrome code from the value of the ECCISMnTED register and detect errors.
0	ECREIS	ECC redundant bit input data selection bit This bit is used to indetify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the ECCISMnERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate syndrome code from ECC data stored in RAM and detect errors. 1: Generate syndrome code from the value of the ECCISMnERDB register and detect errors.

46.6.5 ECCISMnTED — ISMn RAM ECC Encode/Decode Input/Output Replacement Test Register

In ECC test mode, this register handles test data.

The value of the register can be used to generate ECC data or syndrome code.

When ECC test mode is enabled (ECCISMnTMC.ECTMCE = 1), it is accessible. When ECCISMnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when ISM is not accessed to RAM.

Access: This register can be read/written in 32-bit units.

Address: FFC7 $800C_H$

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.52 ECCISMnTED Register Contents

Bit position	Bit Name	Function
31 to 0	ECEDB[31:0]	When ECCISMnTMC.ECENS = 1, the value of this register is used to generate ECC data and the value this register to RAM is stored to RAM. When ECCISMnTMC.ECDCS = 1, the value of this register is used to generate syndrome code and the value of this register is stored in ECC decode syndrome data register (ECCISMnSYND). In addition, when ECCISMnTMC.ECTRRS = 1, RAM data [31:0], instead of written data, is read for the value of this register.

46.6.6 ECCISMnTRC — ISMn RAM ECC Redundant Bit Data Control Test Register

In ECC test mode, this test register, for ECC data, consists of four 8-bit registers, ECCISMnSYND, ECCISMnHORD, ECCISMnECRD, and ECCISMnERDB.

When ECC test mode is enabled (ECCISMnTMC.ECTMCE = 1), this register can be accessed. When ECCISMnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when ISM is not accessed to RAM.

Access: This register can be read/written in 32-bit units.

Address: FFC7 8008_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECCISMnSYND (See **Section 46.6.7**)								ECCISMnHORD (See **Section 46.6.8**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECCISMnECRD (See **Section 46.6.9**)								ECCISMnERDB (See **Section 46.6.10**)							
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

46.6.7 ECCISMnSYND — ISMn RAM ECC Decode Syndrome Data Register

In ECC test mode, this is a read-only register for storing generated syndrome code.

Writing to this register is ignored.

When ECC test mode is enabled (ECCISMnTMC.ECTMCE = 1), this register can be accessed. When ECC test mode is disabled (ECCISMnTMC.ECTMCE = 0), 00_H is read.

Access: This register can be read in 8-bit units.

Address: FFC7 $800B_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SYND6	SYND5	SYND4	SYND3	SYND2	SYND1	SYND0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 46.53 ECCISMnSYND Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	SYND[6:0]	The generated syndrome code is stored as needed.

46.6.8 ECCISMnHORD — ISMn RAM ECC 7-Bit Redundant Bit Data Hold Test Register

In ECC test mode, this register is used to store ECC data for read RAM data.

Writing to this register is ignored.

When ECC test mode is enabled (ECCISMnTMC.ECTMCE = 1), this register can be accessed. When ECC test mode is disabled (ECCISMnTMC.ECTMCE = 0), 00_H is read.

Access: This register can be read in 8-bit units.

Address: FFC7 $800A_H$

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	HORD6	HORD5	HORD4	HORD3	HORD2	HORD1	HORD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 46.54 ECCISMnHORD Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	HORD[6:0]	ECC code for read RAM data is stored as needed. When ECCISMnTMC.ECTRRS = 1 and ECCISMnTED register is read, ECC code is stored.

46.6.9 ECCISMnECRD — ISMn RAM ECC Encode Test Register

In ECC test mode, this is a read-only register for storing ECC data generated for write RAM data.

Writing to this register is ignored.

When ECC test mode is enabled (ECCISMnTMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (ECCISMnTMC.ECTMCE = 0), 00_H is read.

Access: This register can be read in 8-bit units.

Address: FFC7 8009_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ECRD6	ECRD5	ECRD4	ECRD3	ECRD2	ECRD1	ECRD0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 46.55 ECCISMnECRD Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned.
6 to 0	ECRD[6:0]	These bits can read ECC data generated at the time of RAM data writing, and can read ECC data for the data written to the ECCISMnTED register when ECCISMnTMC.ECENS = 1.

46.6.10 ECCISMnERDB — ISMn RAM ECC Redundant Bit Input/Output Replacement Register

In ECC test mode, this register handles ECC data.

The value of this register can be used as ECC data generated at the time of RAM writing or ECC data read at the time of RAM data reading.

When ECC test mode is enabled (ECCISMnTMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (ECCISMnTMC.ECTMCE = 0), 00_H is read.

Access: This register can be read/written in 8-bit units.

Address: FFC7 8008_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	ERDB6	ERDB5	ERDB4	ERDB3	ERDB2	ERDB1	ERDB0
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.56 ECCISMnERDB Register Contents

Bit position	Bit Name	Function
7	Reserved	When read, the value after reset is returned. When writing to this bit, write the value after reset.
6 to 0	ERDB[6:0]	When ECCISMnTMC.ECREOS = 1, these bits store the value of this register as ECC data to RAM. When ECCISMnTMC.ECREIS = 1, the value of this register is read as ECC data read from RAM. When ECCISMnTMC.ECTRRS = 1, ECC data to be stored in RAM is read instead of the write data for the read value of this register.

46.6.11 ECCISMnEAD0 — ISMn ECC Error Address Register

ECCISMnEAD0 is a read-only register to hold the address at which an ECC error has occurred.

Access: This register can be read/written in 32-bit units.

Address: FFC7 8010$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEAD[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEAD[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 46.57 ECCISMnEAD0 Register Contents

Bit position	Bit Name	Function
31 to 0	ECEAD[31:0]	ECCISMnEAD0 is a read-only register to hold the address at which an ECC error has occurred. If an ECC error is detected while ECC error detection is enabled, the RAM address is latched using the detection signal as a trigger, and the address is stored in ECCISMnEAD0 as the address at which the ECC error has occurred. The address is stored upon detection of the first ECC error while no error status is set. However, if a 1-bit error is followed by a 2-bit error, the address of the latter is stored. Only one address can be held in ECCISMnEAD0.

Section 47 Functional Safety

The RH850/D1L/D1M series microcontrollers are designed under the premiss that correct functionality of all safety relevant components can be proven. Therefore it features several modules, which ensure respectively support continuous verification of the microcontroller's functionality.

All devices of the RH850/D1L/D1M series are developed as Safety Element out of Context (SEooC) in accordance to ISO26262 targeting ASIL B.
For details about safety work products, "RH850/D1L/D1M Group Safety Application Note" and etc., please contact our sales representatives.

The following table summarizes which microcontroller components are considered as safety relevant and how their functionality can be proven.

Table 47.1 Safety features overview (1/2)

Hardware function	Test concept
Memory Protection Unit (MPU)	Cyclic test by application software. MPU functionality can be tested by provoking protection violations.
Interrupt Controller (INTC)	Cyclic test by application software INTC functionality can be tested by asserting interrupt requests by software. Refer to Section 7, Interrupt for details.
Bus access protections	Bus guards for single- and multi-master busses to prevent prohibited accesses to certain resources. Refer to Section 14.6, Bus Guards for details.
Flash memories and RAM	• Memory -DMA-CRC check Checksum calculation to secure content of data blocks. This test is supported by the DMA Controller and Data CRC (DCRA) function. The target data can be copied from the memory to DCRA by the DMA Controller (Memory-DMA-CRC check). Refer to Section 49, Data CRC (DCRA) for details. • Local RAM and flash memories are protected by Error Correction Coding (ECC) circuits ECC circuits can also be tested provoking memory errors. Refer to the description of the Error Correction Coding circuits in the respective sections for details. • Video RAM can be ECC protected by a special access mode of the RAM wrapper. Refer to Section 54, Video RAM and Video RAM Wrapper (VRAM) for details.
Functional modules	Functionality of most modules can be verified by software controlled tests on application level. RAMs of the modules are protected by Error Correction Coding (ECC) circuits.
Window Watchdog (WDTA)	Watchdog modules to escape from a system deadlock or program runaway. Refer to Section 26, Window Watchdog Timer (WDTA) the for details.
Clock generators	Supervision of the oscillators and PLL circuits by Clock Monitors (CLMA). Clock Monitors can be tested by provoking clock generators fails. Refer to Section 12.7, Clock Monitor A (CLMA) for details.
Temperature control	Temperature measurement circuit warns about out-of-range temperature. Refer to Section 11, Temperature Measurement for details.
Supply voltage	Power-On-Clear (POC) circuit ensures microcontroller reset if voltage supply below minimum. Refer to Section 10, Power Supply and Power Domains and Section 9, Reset Controller for details. For external voltage supervision the A/D Converter (ADCE) with built-in comparator function can be utilized. Refer to Section 45, A/D Converter (ADCE) for details.
Reset	Redundant Reset Controller ensures reset functionality. Refer to Section 9, Reset Controller for details.
Clocked serial interfaces (CSIG, CSIH), CAN interface (RS-CAN), CANFD Interface (RS-CANFD)	These communication interfaces feature loop back mode function for self tests. Refer to Section 18, Clocked Serial Interface G (CSIG), Section 19, Clocked Serial Interface H (CSIH) and Section 22, CAN Interface (RSCAN), Section 23, CANFD Interface (RS-CANFD) for details.
A/D Converter (ADCE)	The ADCE features a self diagnosis function to test correct functionality. Refer to Section 45, A/D Converter (ADCE) for details.

Table 47.1 Safety features overview (2/2)

Hardware function	Test concept
PWM diagnosis and port check functions	Verification of correct operation of safety relevant notification signals, e.g. warning lights and sound output. Refer to Section 32, PWM Generators and Diagnostic (PWM-Diag) and Section 2.4, Output ports check functions for details.
Video input pixel clock supervision	Verification of live view from external video sources. Refer to Section 37.5.8, Video Input Pixel Clock Monitoring for details.
Video output data check	Verification of correct video output content. Refer to Section 40, Video Output Checker A (VOCA) and Section 41, Display Output Comparator (DISCOM) for details.
2D Graphics Processing Unit (GPU2D)	Checksum verification of drawing command lists to ensure correct graphic drawings. Refer to Section 42, 2D Graphics Processing Unit (GPU2D) for details.
Capture-compare timers, Zero Point Detection	Verification of correct position of dialers and gauge needles.
Error Control Module (ECM)	Collection of all exception that are asserted from various modules in one entity. Depending on the safety use-case it can be configured whether a reset, non-maskable or maskable interrupt is asserted. Refer to Section 48, Error Control Module (ECM) for details.
Data CRC (DCRA)	Hardware CRC module that can be utilized e.g. to ensure end-to-end protection of communication interfaces or regular y check memory areas.

Section 48 Error Control Module (ECM)

48.0.1 ECM Master and Checkers

Following table summarizes which devices feature an ECM Master and ECM Checker module.

Table 48.1 ECM Master and Checkers ("√": module available, –": module not available)

ECM module	D1L1	D1L2(H)	D1M1(H)	D1M1-V2 D1M1A	D1M2(H)
Master	√	√	√	√	√
Checker	√	√	√	–	√

Refer to Section 48.1.11, ECM Master - Checker operation for details about ECM Master and Checker.

48.1 Overview

48.1.1 Specification Overview

ECM (Error Control Module) collects error signals coming from different error sources and monitoring circuits. It also outputs error signals from the error pin ($\overline{\text{ERROROUT}}$) and generates interrupts and internal reset signals. **Table 48.2** shows the specification overview of ECM.

Table 48.2 Specification Overview (1/2)

Item	Description
Safety processing	ECM can handle the following processing in response to error signal inputs from individual modules. • Error flag set • FE level maskable interrupt generation FE level maskable interrupt generation can be controlled (enabled/disabled) for individual errors. • FE level non-maskable interrupt generation FE level non-maskable interrupt generation can be controlled (enabled/disabled) for individual errors. • Internal reset generation Internal reset generation can be controlled (enabled/disabled) for individual errors. • Error pin output Pin output mask can be controlled (enabled/disabled) for individual errors. Output can be toggled in response to a timer input or made at a fixed level.
Error status	ECM incorporates the error status register, which can be used to confirm the error status from the error flag. The error flags are only cleared by the ECMSTATRES reset. Refer to **Section 9, Reset Controller** for details about the ECMSTATRES.
Debug, self-diagnosis	• Pseudo errors can be generated for debug and self-diagnosis. The operation during injection of pseudo errors is identical to that for the occurrence of real errors. All configurations for the mask to the error pin output, interrupt, or internal reset apply in the same way. • ECM incorporates a loop-back function of the error pin output that is used to diagnose the path to the error output pin. The status of the error output pin is reflected to an internal register and can be confirmed by reading the register.
Timeout function	ECM incorporates a function that generates an error signal output or internal reset when the count value of the delay timer matches with the delay timer compare register because the delay timer was not stopped during the interrupt processing after being started simultaneously with the occurrence of an interrupt request.

Table 48.2 Specification Overview (2/2)

Item	Description
Register protection	A write-protection with a special sequence is incorporated to protect registers from inadvertent write access.
Others	ECM is duplexed. ECM incorporates the error output pin.

48.1.2 Error Input

Table 48.3 shows the error input to ECM of RH850/D1L/D1M.

Table 48.3 List of Error Inputs (1/3)

No	Error Source	Unit	Initial Settings					
			FE level maskable Interrupt		FE level non-maskable Interrupt		Internal Reset	Error Mask
				Delay Timer Start		Delay Timer Start		
0	Watchdog 0 NMI output signal INTWDTA0NMI	WDTA0	OFF	OFF	OFF	OFF	**ON**	OFF
1	Watchdog 1 NMI output signal INTWDTA1NMI	WDTA1	OFF	OFF	OFF	OFF	**ON**	OFF
2	External non-maskable interrupt NMI	Port	OFF	OFF	OFF*1	OFF*1	**ON**	OFF
3	Reserved	—	—	—	—	—	—	—
4	Logical OR combination of Code Flash ECC 2-bit error signals INTECCDFLI: 2-bit error detection during access by the CPU 2-bit error detection during access by the GVCI master	CPU Subsystem	OFF	OFF	OFF	OFF	OFF	OFF
5	Logical OR combination of Code Flash ECC 1-bit error signals INTECCSFLI: 1-bit error detection during access by the CPU 1-bit error detection during access by the GVCI master	CPU Subsystem	OFF	OFF	OFF	OFF	OFF	OFF
6	Data Flash ECC 2-bit error interrupt INTECCDEDEEP	Data Flash	OFF	OFF	OFF	OFF	OFF	OFF
7	Data Flash ECC 1-bit error interrupt INTECCSEDEEP	Data Flash	OFF	OFF	OFF	OFF	OFF	OFF
8	Logical OR combination of Instruction Cache RAM ECC error signals INTECCIC: Instruction cache data RAM ECC 2-bit error detection Instruction cache data RAM ECC 1-bit error detection Instruction cache tag RAM ECC 2-bit error detection Instruction cache tag RAM ECC 1-bit error detection	CPU Subsystem	OFF	OFF	OFF	OFF	OFF	OFF
9	Logical OR combination of XC Cache RAM ECC error signals INTECCXCC XC Cache Data RAM ECC 2-bit error detection XC Cache Data RAM ECC 1-bit error detection XC Cache Tag RAM ECC 2-bit error detection XC Cache Tag RAM ECC 1-bit error detection	CPU Subsystem	OFF	OFF	OFF	OFF	OFF	OFF

Table 48.3 List of Error Inputs (2/3)

No	Error Source	Unit	Initial Settings					
			FE level maskable Interrupt		FE level non-maskable Interrupt		Internal Reset	Error Mask
				Delay Timer Start		Delay Timer Start		
10	Logical OR combination of various video output display error signals INTVOCAERR*3:		OFF	OFF	OFF	OFF	OFF	OFF
	Video Output Monitor error	VOCA0						
	Activity Monitor error							
	Display output compare error detect interrupt	DISCOM0						
	Display output compare error detect interrupt	DISCOM1						
	Display output compare error detect interrupt	DISCOM2						
	Display output compare error detect interrupt	DISCOM3						
11	Logical OR combination of various bus guard error signals INTPERIGRD:		OFF	OFF	OFF	OFF	OFF	OFF
	PBUS Guards errors	PBGn						
	Cross-connect Guards errors	XCGn						
12	Logical OR combination of PE Guard error signals INTPEGRD:	PEG	OFF	OFF	OFF	OFF	OFF	OFF
	PEGuard read error							
	PEGuard write error							
13	Logical OR combination of VDCE0 error interrupt INTVDCE0ERR	VDCE0	OFF	OFF	OFF	OFF	OFF	OFF
14	Logical OR combination of VDCE1 error interrupt INTVDCE1ERR	VDCE1	OFF	OFF	OFF	OFF	OFF	OFF
15	Local CPU RAM ECC 2-bit error detection INTECCDLRAM	LRAM	OFF	OFF	OFF	OFF	OFF	OFF
16	Local CPU RAM ECC 1-bit error detection INTECCSLRAM		OFF	OFF	OFF	OFF	OFF	OFF
17	FACI Reset transfer error (FRTERR)	FACI	OFF	OFF	OFF	OFF	OFF	OFF
18	Flash sequencer error (FLERR)	FACI	OFF	OFF	OFF	OFF	OFF	OFF
19	Retention RAM ECC 2-bit error interrupt INTECCDEDRRAM	RRAM	OFF	OFF	OFF	OFF	OFF	OFF
20	Retention RAM ECC 1-bit error interrupt INTECCSEDRRAM		OFF	OFF	OFF	OFF	OFF	OFF
21	Reserved	—	—	—	—	—	—	—
22	Logical OR combination RSCAN/ISM ECC 2-bit error signals INTECCDPERIRAM:		OFF	OFF	OFF	OFF	OFF	OFF
	RS-CAN RAM ECC 2-bit error interrupt	RSCAN0						
	Stepper Motor Driver RAM ECC 2-bit error interrupt	ISM0						
23	Logical OR combination RSCAN/ISM ECC 1-bit error signals INTECCSPERIRAM:		OFF	OFF	OFF	OFF	OFF	OFF
	RS-CAN RAM ECC 1bit-error interrupt	RSCAN0						
	Stepper Motor Driver RAM ECC 1-bit error interrupt	ISM0						
24	Video RAM wrapper ECC 2-bit error interrupt INTECCDEDVRAM	VRAM	OFF	OFF	OFF	OFF	OFF	OFF
25	Video RAM wrapper ECC 1-bit error interrupt INTECCSEDVRAM	VRAM	OFF	OFF	OFF	OFF	OFF	OFF
26	Interrupt INTOSTM1	OSTM1	OFF	OFF	OFF	OFF	OFF	OFF
27	Reserved	—	—	—	—	—	—	—

Table 48.3 List of Error Inputs (3/3)

No	Error Source	Unit	Initial Settings					
			FE level maskable Interrupt		FE level non-maskable Interrupt		Internal Reset	Error Mask
				Delay Timer Start		Delay Timer Start		
28	CLMA1 interrupt INTCLMATI1*4	CLMA1	OFF	OFF	OFF	OFF	OFF	OFF
29	CLMA2 interrupt INTCLMATI2*4	CLMA2	OFF	OFF	OFF	OFF	OFF	OFF
30	CLMA3 interrupt INTCLMATI3*4	CLMA3	OFF*2	OFF*2	OFF*2	OFF*2	OFF	OFF
31	CLMA4 interrupt INTCLMATI4*4	CLMA4	OFF*2	OFF*2	OFF*2	OFF*2	OFF	OFF
32	CLMA5 interrupt INTCLMATI5*4	CLMA5	OFF	OFF	OFF	OFF	OFF	OFF
33	CLMA6 interrupt INTCLMATI6*4	CLMA6	OFF	OFF	OFF	OFF	OFF	OFF
34	Logical OR combination of SG0 ports error signals INTSG0DIAG:	SG0 ports XOR Compare Unit	OFF	OFF	OFF	OFF	OFF	OFF
35	Logical OR combination of PCMP0 ports error signals INTPCMP0DIAG:	PCMP0 ports XOR Compare Unit	OFF	OFF	OFF	OFF	OFF	OFF
36	Logical OR combination of TAUB0/TAUB1 ports error signals INTTAUBDIAG:	TAUB ports XOR Compare Unit	OFF	OFF	OFF	OFF	OFF	OFF
37	ECM compare error INTECM	ECM	OFF	OFF	OFF	OFF	OFF	OFF
38	Reserved	—	—	—	—	—	—	—
39	Reserved	—	—	—	—	—	—	—
40	Error Interrupt INTADCEERR	ADCE0	OFF	OFF	OFF	OFF	OFF	OFF
41	Stepper Motor Driver Zero Point Detection interrupt INTISM0ZPD	ISM0	OFF	OFF	OFF	OFF	OFF	OFF
42	Reserved	—	—	—	—	—	—	—
43	Reserved	—	—	—	—	—	—	—
44	Reserved	—	—	—	—	—	—	—
45	Reserved	—	—	—	—	—	—	—
46	Reserved	—	—	—	—	—	—	—
47	Reserved	—	—	—	—	—	—	—

Note 1. If the port filter of the external non-maskable interrupt NMI is configured for level detection, the NMI must not be used to generate an FE level non-maskable interrupt.

Note 2. The error output signal of the Clock Monitor, that monitors the source clock of the CPU Subsystem clock C_ISO_CPUCLK must not be used to generate an FE level maskable or non-maskable interrupt, but an internal reset.
That means:
If PLL0CLK is the source clock of C_ISO_CPUCLK, the CLMA3 error output can only be used to generate an internal reset.
If PLL1CLK is the source clock of C_ISO_CPUCLK, the CLMA4 error output can only be used to generate an internal reset.

Note 3. If the error mask of INTVOCAERR is released an unintended error signal may be generated after DEEPSTOP mode. Refer to Section 40.5.1, Procedure for entering and resuming DEEPSTOP for detail.

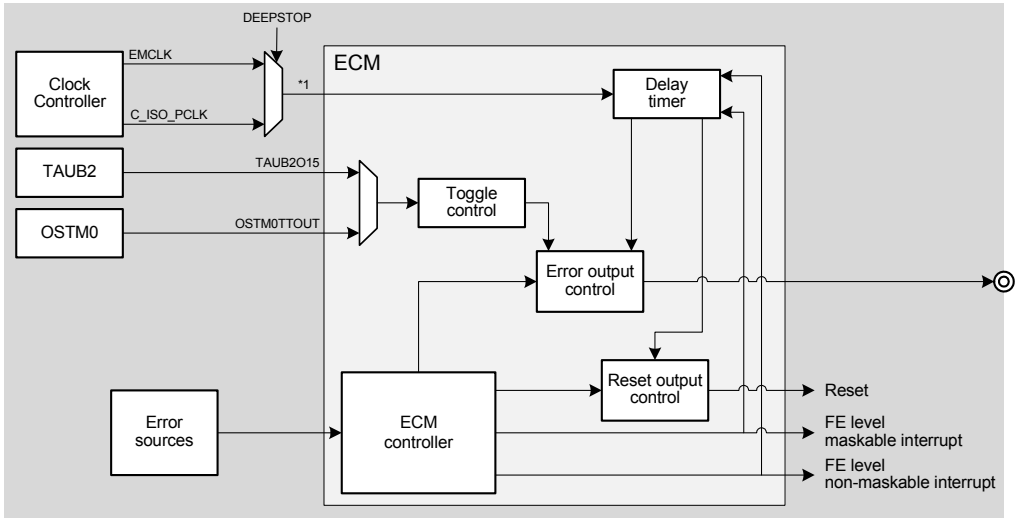
Note 4. If the error mask of INTCLMATIn (n = 1 to 6) is released an unintended error signal may be generated after DEEPSTOP mode. Refer to Section 12.7.7.1, ECM detects false error signals from CLMA at resuming DEEPSTOP for detail.

NOTE

OFF: This indicates that when an error occurs, an FE level maskable interrupt, FE level non-maskable interrupt, internal reset, or delay timer start is not generated, and the error mask is disabled.

ON: This indicates that when an error occurs, an FE level maskable interrupt, FE level non-maskable interrupt, internal reset, or delay timer start is generated, and the error mask is enabled.

48.1.3 Error Control Module Block Diagram



Note 1. In normal operation mode the delay time is supplied with the C_ISO_PCLK clock. In DEEPSTOP mode, the delay timer clock changes automatically to EMCLK.

Figure 48.1 Error Control Module block diagram

48.1.4 Operations for Error Output

All error sources can be configured to assert the external error output signal $\overline{\text{ERROROUT}}$, which is used to alert externally connected device.

The behaviour of the external error signal upon an error occurrence can be selected:

- Non-dynamic mode
  The error signal is asserted and keeps this level.
- Dynamic mode
  The error signal level toggles, i.e. a pulse signal is generated. The toggle frequency is determined by the Timer Array Unit B TAUB2 channel 15 or the OS Timer OSTM0 output.
- Error output signal set/reset by software
  The error output signal can be set active/inactive by the software.

After reset release, the $\overline{\text{ERROROUT}}$ pin outputs the low (error) level. Follow the procedure described in Section 48.2.3, ECM Master/Checker Error Clear Trigger Register (ECMmECLR, m = M/C), to clear the error before using ECM.

The error output can be configured for two different modes of operation, non-dynamic or dynamic.

Error Status ECMmSSE031 to ECMmSSE000 ECMmSSE115 to ECMmSSE100	Operating Mode ECMSL0 Bit	Error Output Operating Mode	Error Output Level	Error Status
0	0	Non-dynamic	H	No error
	1	Dynamic	Toggles (according to timer input)	No error
1	0	Non-dynamic	L	Error
	1	Dynamic	L	Error

48.1.4.1 Dynamic Mode Enable

1. Select the timer output by the ECMEPCTL register (see Section 48.2.29, ECM Error Pulse Control Register (ECMEPCTL).
2. Initialize OSTM0/TAUB2 channel 15.
3. Set the error output to high level by setting the ECMmECT (m = M/C) bit in the ECM master/checker error clear trigger register to 1.
4. Set the ECMSL0 bit in the ECM error pulse configuration register to 1 to specify dynamic mode.
5. Start up OSTM0/TAUB2 channel 15.

48.1.4.2 Dynamic Mode Disable

1. Set the error output to low level by setting the ECMmEST bit (m = M/C) in the ECM master/checker error clear trigger register to 1.
2. Stop OSTM0/TAUB2 channel 15.
3. Clear the ECMSL0 bit in the ECM error pulse configuration register to 0 to specify non-dynamic mode.

48.1.5 Delay timer

By use of a delay timer the assertion of the error output and reset signals can be delayed after the occurrence of an error event has generated a maskable or non-maskable interrupt.

After overflow of the programmable delay time the error output and the reset signals are asserted and stays active until it is de-asserted by the software.

This way the software has some time to recover from the error without notifying the error to external.

48.1.6 Loop-Back Function

ECM incorporates a loop-back function that is used to check the path to the error output pin. The output level of the error output pin can be checked with the ECMmSSE131 bit (m = M/C) in the ECM master/checker error source status register 1.

48.1.7 Pseudo Error Generation

ECM incorporates a function that can generate pseudo errors for test or debug purposes. The operation of the ECM during injection of pseudo errors is identical to that for the occurrence of real errors. All configurations for error masks, interrupt, internal reset, or delay timer apply in the same way.

48.1.8 Error Status

The error status is indicated by ECM master/checker error source status register 0 and ECM master/checker error source status register 1.
The error status is only cleared by the application program (write to ECMESSTC0 and ECMESSTC1 register) or by the ECMSTATRES signal (see Section 9, Reset Controller).
In case of any other reset, the error status is kept and the error of the reset source can be confirmed by reading the ECM master/checker error source status register 0 and ECM master/checker error source status register 1 after reset release.

48.1.9 Writing to Protected Registers

Write protected registers are protected from inadvertent write access due to erroneous program execution, etc.

48.1.9.1 Protection Unlock Sequence

Write access to a write protected register is only possible within the following sequence.

1. Write the fixed value 0000 $00A5_H$ to the ECM protection command register or ECM master/checker protection command register.
2. Write the desired value to the ECM register.
3. Write the inversion of desired value to the ECM register.
4. Write the desired value to the ECM register.
5. Check successful write of the desired value to the protected register by checking that the ECMPRERR bit of the ECM protection status register is 0.

In case of any access to another register between step 1 to step 4 of the above sequence, the protection mechanism behaves as follows.

- If that register belongs to ECM, the write to the protected register fails (the ECMPRERR bit of the ECM protection status register becomes 1). The sequence has to be reexecuted from step 1.
- If that register does not belong to ECM, the sequence is not disrupted and the write to the protected register is conducted successfully.

For details, refer to Section 4, Write-Protected Registers.

48.1.10 Timeout Function for Interrupt Processing

The delay timer incorporated to ECM can be started simultaneously with the occurrence of an interrupt request. ECM incorporates a function that generates an error signal output or internal reset when the count value of the delay timer matches with the value of the delay timer compare register because the delay timer was not stopped during the interrupt processing. The timer counting is not stopped when a break occurs.

The counting of the delay timer always starts from 0. Configure the duration until an internal reset or error output is generated with the settings of the delay timer compare register.

The clock supply of the delay timer depends on the operation mode:

- in normal operation mode: C_ISO_PCLK
- in DEEPSTOP mode: EMCLK

48.1.11 ECM Master - Checker operation

The figure below shows, how the separate output signals of the ECM Master and Checker are generating the ECM output signals.

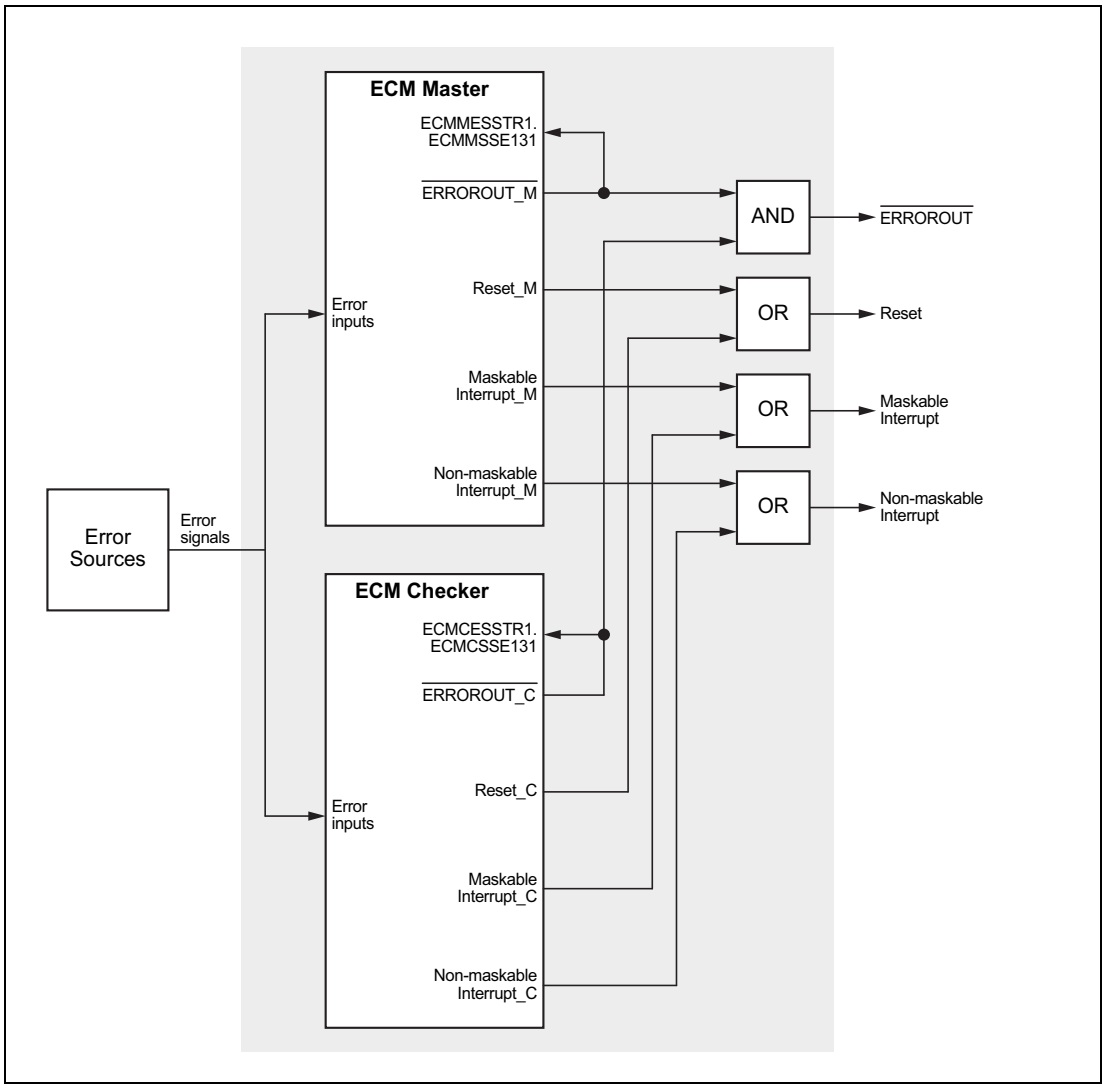


Figure 48.2 ECM Master - Checker operation

48.2 Register Specification

NOTE

The bits in registers, which refer to certain error input signals, may not all be available.

Refer to the “No” column in Table 48.3, List of Error Inputs to check which bits in those registers are available.

48.2.1 List of Registers

ECM consists of three address areas: common part, ECM master, and ECM checker.

The following shows the register map of the ECM master registers.

Table 48.4 Address List of ECM Master Registers

<ECMM_base: FFCB 0000$_H$>

Register Symbol	Register Name	R/W	Initial Value	Protection by Sequence	Address
ECMMESET	ECM master error set trigger register	W	00_H	Protected	<ECMM_base>
ECMMECLR	ECM master error clear trigger register	W	00_H	Protected	<ECMM_base> + 04_H
ECMMESSTR0	ECM master error source status register 0	R	0000 0000_H	Not protected	<ECMM_base> + 08_H
ECMMESSTR1	ECM master error source status register 1	R	0000 0000_H	Not protected	<ECMM_base> + $0C_H$
ECMMPCMD0	ECM master protection command register	W	Undefined	Not protected	<ECMM_base> + 10_H

The following shows the register map of the ECM checker registers.

Table 48.5 Address List of ECM Checker Registers

<ECMC_base: FFCB 1000$_H$>

Register Symbol	Register Name	R/W	Initial Value	Protection by Sequence	Address
ECMCESET	ECM checker error set trigger register	W	00_H	Protected	<ECMC_base>
ECMCECLR	ECM checker error clear trigger register	W	00_H	Protected	<ECMC_base> + 04_H
ECMCESSTR0	ECM checker error source status register 0	R	0000 0000_H	Not protected	<ECMC_base> + 08_H
ECMCESSTR1	ECM checker error source status register 1	R	0000 0000_H	Not protected	<ECMC_base> + $0C_H$
ECMCPCMD0	ECM checker protection command register	W	Undefined	Not protected	<ECMC_base> + 10_H

The following shows the register map of the ECM common part.

Table 48.6 Address List of ECM Registers

<ECM_base: FFCB 2000$_H$>

Register Symbol	Register Name	R/W	Initial Value	Protection by Sequence	Address
ECMEPCFG	ECM error pulse configuration register	R/W	00_H	Protected	<ECM_base>
ECMMICFG0	ECM maskable interrupt configuration register 0	R/W	$0000\ 0000_H$	Protected	<ECM_base> + 04_H
ECMMICFG1	ECM maskable interrupt configuration register 1	R/W	$0000\ 0000_H$	Protected	<ECM_base> + 08_H
ECMNMICFG0	ECM FE level non-maskable interrupt configuration register 0	R/W	$0000\ 0000_H$	Protected	<ECM_base> + $0C_H$
ECMNMICFG1	ECM FE level non-maskable interrupt configuration register 1	R/W	$0000\ 0000_H$	Protected	<ECM_base> + 10_H
ECMIRCFG0	ECM internal reset configuration register 0	R/W	$0000\ 000F_H$	Protected	<ECM_base> + 14_H
ECMIRCFG1	ECM internal reset configuration register 1	R/W	$0000\ 0000_H$	Protected	<ECM_base> + 18_H
ECMEMK0	ECM error mask register 0	R/W	$0000\ 0000_H$	Protected	<ECM_base> + $1C_H$
ECMEMK1	ECM error mask register 1	R/W	$0000\ 0000_H$	Protected	<ECM_base> + 20_H
ECMESSTC0	ECM error source status clear register 0	W	$0000\ 0000_H$	Protected	<ECM_base> + 24_H
ECMESSTC1	ECM error source status clear register 1	W	$0000\ 0000_H$	Protected	<ECM_base> + 28_H
ECMPCMD1	ECM protection command register	W	Undefined	Not protected	<ECM_base> + $2C_H$
ECMPS	ECM protection status register	R	00_H	Not protected	<ECM_base> + 30_H
ECMPE0	ECM pseudo error trigger register 0	W	$0000\ 0000_H$	Protected	<ECM_base> + 34_H
ECMPE1	ECM pseudo error trigger register 1	W	$0000\ 0000_H$	Protected	<ECM_base> + 38_H
ECMDTMCTL	ECM delay timer control register	R/W	00_H	Protected	<ECM_base> + $3C_H$
ECMDTMR	ECM delay timer register	R	0000_H	Not protected	<ECM_base> + 40_H
ECMDTMCMP	ECM delay timer compare register	R/W	0000_H	Protected	<ECM_base> + 44_H
ECMDTMCFG0	ECM delay timer configuration register 0	R/W	$0000\ 0000_H$	Protected	<ECM_base> + 48_H
ECMDTMCFG1	ECM delay timer configuration register 1	R/W	$0000\ 0000_H$	Protected	<ECM_base> + $4C_H$
ECMDTMCFG2	ECM delay timer configuration register 2	R/W	$0000\ 0000_H$	Protected	<ECM_base> + 50_H
ECMDTMCFG3	ECM delay timer configuration register 3	R/W	$0000\ 0000_H$	Protected	<ECM_base> + 54_H
ECMEPCTL	ECM error pulse control register	R/W	00_H	Not protected	FFC0 $600C_H$

The ECM registers are the register areas common to the redundancy area to be implemented. Writes to the common register areas are conducted simultaneously. The common area for ECM master is read by reading access to the common area. The ECM master register and the ECM checker register represent the address areas which can be written separately.

48.2.2 ECM Master/Checker Error Set Trigger Register (ECMmESET, m = M/C)

The ECM master/checker error set trigger register is for setting the error signal from the error pin to the low level. When the ECMmEST bit is set to 1, the error pin immediately outputs the low level. The output cannot be masked. You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers, for the details of the write protection sequence. This register is always read as 00_H.

Access: This register can be written in 8-bit units.

Address: <ECMM_base>
<ECMC_base>

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ECMmEST
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 48.7 ECMmESET register contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When written, write the value after reset.
0	ECMmEST	Error set trigger bit 0: Writing 0 is invalid 1: Set the output level from the error pin to the active (low) level.

CAUTIONS

Setting or clearing the error output from the error pin via the ECMmESET or ECMmECLR register will set the ECMmSSE105 bit of the ECMmESSTR1 register (ECM compare error). Therefore, the ECMmESET or ECMmECLR register has to be set following the sequence below.

1. Set the ECMEMK105 bit of the ECMEMK1 register to “masked”.
2. Prevent the generation of interrupts by setting the ECMNMIE105 bit of the ECMNMICFG1 and the ECMMIE105 bit of the ECMMICFG1 register to “prohibited”.
3. Prevent generation of an internal reset by setting the ECMIRE105 bit of the ECMIRCFG1 register to “prohibited”.
4. Set or clear the error output bit in the ECMmESET or ECMmECLR register.
5. Clear error flags by setting the ECMCLSSE105 bit of the ECMESSTC1 register.
6. Make the following settings in accord with the condition of usage for the ECM compare error.
   - If an error is being output from the error pin, set the ECMEMK105 bit of the ECMEMK1 register to “not masked”.
   - If an interrupt is being enabled, set the ECMMIE105 bit of the ECMMICFG1 and the ECMNMIE105 bit of the ECMNMICFG1 register to “enabled”.
   - If an internal reset is being enabled, set the ECMIRE105 bit of the ECMIRCFG1 register to “enabled”.

48.2.3 ECM Master/Checker Error Clear Trigger Register (ECMmECLR, m = M/C)

The ECM master/checker error clear trigger register is for setting the error signal from the error pin to the high level (toggle). When the ECMmECT bit is set to 1, the error pin immediately outputs the high level (toggle) as long as there are no other sources that set the error pin to the low level. You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers, for the details of the write protection sequence. This register is always read as 00_H.

Access: This register can be written in 8-bit units.

Address: <ECMM_base> + 04_H
<ECMC_base> + 04_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ECMmECT
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	W

Table 48.8 ECMmECLR register contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When written, write the value after reset.
0	ECMmECT	Error clear trigger bit 0: Writing 0 is invalid 1: Set the output level from the error pin to the inactive (high) level.

CAUTIONS

Clearing of the error pin is only possible if all errors, not masked by ECMEMK0/1, are cleared beforehand.

Setting or clearing the error output via the ECMmECLR or ECMmESET register will generate the error. Therefore, the following has to be set in advance.

1. Set the ECMEMK105 (ECM compare error) bit of the ECMEMK1 register to “masked”.
2. Disable interrupts by setting the ECMMIE105 bit of the ECMMICFG1 and the ECMNMIE105 bit of the ECMNMICFG1 register to “disabled”.
3. Disable an internal reset by setting the ECMIRE105 bit of the ECMIRCFG1 register to “disabled”.
4. Set or clear error flags by setting the ECMmECLR or ECMmESET (m = M/C) register.
5. Clear error flags by setting the ECMCLSSE105 bit of the ECMESSTC1 register.

48.2.4 ECM Master/Checker Error Source Status Register 0 (ECMmESSTR0, m = M/C)

The ECM master/checker error source status register 0 is a read-only register.

The error status is only cleared by the application program (write to ECMESSTC0 and ECMESSTC1 register) or by the ECMSTATRES signal (see Section 9, Reset Controller).
In case of any other reset, the error status is kept and the error of the reset source can be confirmed by reading the ECM master/checker error source status register 0 and ECM master/checker error source status register 1 after reset release.

Access: This register can be read in 32-bit units.

Address: <ECMM_base> + 08_H
<ECMC_base> + 08_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMmSSE031	ECMmSSE030	ECMmSSE029	ECMmSSE028	ECMmSSE027	ECMmSSE026	ECMmSSE025	ECMmSSE024	ECMmSSE023	ECMmSSE022	ECMmSSE021	ECMmSSE020	ECMmSSE019	ECMmSSE018	ECMmSSE017	ECMmSSE016
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMmSSE015	ECMmSSE014	ECMmSSE013	ECMmSSE012	ECMmSSE011	ECMmSSE010	ECMmSSE009	ECMmSSE008	ECMmSSE007	ECMmSSE006	ECMmSSE005	ECMmSSE004	ECMmSSE003	ECMmSSE002	ECMmSSE001	ECMmSSE000
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 48.9 ECMmESSTR0 register contents

Bit Position	Bit Name	Function
31 to 0	ECMmSSE031 to ECMmSSE000	Error source status bit ECMmSSE031 to ECMmSSE000 correspond to error sources 31 to 0. 0: Error not occurred 1: Error occurred

48.2.5 ECM Master/Checker Error Source Status Register 1 (ECMmESSTR1, m = M/C)

The ECM master/checker error source status register 1 is a read-only register.

This register represents the status of individual internal error sources, which is irrelevant to the setting of the error mask. The status can be cleared only by software and a pin reset. An internal reset will not affect this register.

Access: This register can be read in 32-bit units.

Address: <ECMM_base> + $0C_H$
<ECMC_base> + $0C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMmSSE131	ECMmSSE130	ECMmSSE129	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMmSSE115	ECMmSSE114	ECMmSSE113	ECMmSSE112	ECMmSSE111	ECMmSSE110	ECMmSSE109	ECMmSSE108	ECMmSSE107	ECMmSSE106	ECMmSSE105	ECMmSSE104	ECMmSSE103	ECMmSSE102	ECMmSSE101	ECMmSSE100
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 48.10 ECMmESSTR1 register contents

Bit Position	Bit Name	Function
31	ECMmSSE131	Indicates the error output loopback status. 0: Error output ($\overline{\text{ERROROUT}}$) is the low level. 1: Error output ($\overline{\text{ERROROUT}}$) is the high level.
30	ECMmSSE130	Indicates the ECMmESET write status. 0: No error 1: Error is set by the ECMmEST bit of the ECMmESET register
29	ECMmSSE129	Indicates whether delay timer overflow has occurred. 0: Delay timer overflow not occurred 1: Delay timer overflow occurred
28 to 16	Reserved	When read, the value after reset is read.
15 to 0	ECMmSSE115 to ECMmSSE100	Error source status bit ECMmSSE115 to ECMmSSE100 correspond to error sources 47 to 32. 0: Error not occurred 1: Error occurred

48.2.6 ECM Master/Checker Protection Command Register (ECMmPCMD0, m = M/C)

The ECM master/checker protection command register is a write-only register and can be written in 32-bit units. Refer to Section 48.2.1, List of Registers, for the protected registers.

Refer to Section 48.1.9, Writing to Protected Registers, for the details of the write protection sequence. The initial value is undefined.

Access: This register can be written in 32-bit units.

Address: <ECMM_base> + 10_H
<ECMC_base> + 10_H

Initial value: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	Undefined															
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ECMmREG0[7:0]							
Initial value	Undefined															
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 48.11 ECMmPCMD0 register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When written, write the value after reset.
7 to 0	ECMmREG0[7:0]	Protection command that enables writing to write protected ECMm registers.

48.2.7 ECM Error Pulse Configuration Register (ECMEPCFG)

The ECM error pulse configuration register is a read/write register.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 8-bit units.

Address: <ECM_base>

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ECMSL0
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 48.12 ECMEPCFG register contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is read. When written, write the value after reset.
0	ECMSL0	Error pin operation configuration bit Error output operation setting for the error pin 0: Non-dynamic mode 1: Dynamic mode

48.2.8 ECM FE level maskable Interrupt Configuration Register 0 (ECMMICFG0)

The ECM FE level maskable interrupt configuration register 0 is used to set the generation of the ECMTI interrupts (FEINT: FE level maskable interrupts). The generation of FE level maskable interrupts in response to errors is selectable.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 04_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMMIE031	ECMMIE030	ECMMIE029	ECMMIE028	ECMMIE027	ECMMIE026	ECMMIE025	ECMMIE024	ECMMIE023	ECMMIE022	ECMMIE021	ECMMIE020	ECMMIE019	ECMMIE018	ECMMIE017	ECMMIE016
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMMIE015	ECMMIE014	ECMMIE013	ECMMIE012	ECMMIE011	ECMMIE010	ECMMIE009	ECMMIE008	ECMMIE007	ECMMIE006	ECMMIE005	ECMMIE004	ECMMIE003	ECMMIE002	ECMMIE001	ECMMIE000
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 48.13 ECMMICFG0 register contents

Bit Position	Bit Name	Function
31 to 0	ECMMIE031 to ECMMIE000	ECM maskable interrupt generation control bit ECMMIE031 to ECMMIE000 correspond to error sources 31 to 0. 0: Interrupt generation disabled 1: Interrupt generation enabled

48.2.9 ECM FE level maskable Interrupt Configuration Register 1 (ECMMICFG1)

The ECM FE level maskable interrupt configuration register 1 is used to set the generation of the ECMTI interrupts (FEINT: FE level maskable interrupts).

The generation of FE level maskable interrupts in response to errors is selectable.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 08_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMMIE115	ECMMIE114	ECMMIE113	ECMMIE112	ECMMIE111	ECMMIE110	ECMMIE109	ECMMIE108	ECMMIE107	ECMMIE106	ECMMIE105	ECMMIE104	ECMMIE103	ECMMIE102	ECMMIE101	ECMMIE100
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 48.14 ECMMICFG1 register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is read. When written, write the value after reset.
15 to 0	ECMMIE115 to ECMMIE100	ECM maskable interrupt generation control bit ECMMIE115 to ECMMIE100 correspond to error sources 47 to 32. 0: Interrupt generation disabled 1: Interrupt generation enabled

48.2.10 ECM FE level non-maskable Interrupt Configuration Register 0 (ECMNMICFG0)

The ECM FE level non-maskable interrupt configuration register 0 is used to set the generation of ECMTNMI interrupts (FENMI: FE level non-maskable interrupt).

The generation of FE level non-maskable interrupts in response to errors is selectable.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + $0C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMNMIE031	ECMNMIE030	ECMNMIE029	ECMNMIE028	ECMNMIE027	ECMNMIE026	ECMNMIE025	ECMNMIE024	ECMNMIE023	ECMNMIE022	ECMNMIE021	ECMNMIE020	ECMNMIE019	ECMNMIE018	ECMNMIE017	ECMNMIE016
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMNMIE015	ECMNMIE014	ECMNMIE013	ECMNMIE012	ECMNMIE011	ECMNMIE010	ECMNMIE009	ECMNMIE008	ECMNMIE007	ECMNMIE006	ECMNMIE005	ECMNMIE004	ECMNMIE003	ECMNMIE002	ECMNMIE001	ECMNMIE000
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 48.15 ECMNMICFG0 register contents

Bit Position	Bit Name	Function
31 to 0	ECMNMIE031 to ECMNMIE000	ECM FE level non-maskable interrupt generation control bit ECMNMIE031 to ECMNMIE000 correspond to error sources 31 to 0. 0: Interrupt generation disabled 1: Interrupt generation enabled

48.2.11 ECM FE level non-maskable Interrupt Configuration Register 1 (ECMNMICFG1)

The ECM FE level non-maskable interrupt configuration register 1 is used to set the generation of ECMTNMI interrupts (FENMI: FE level non-maskable interrupt).

The generation of FE level non-maskable interrupts in response to errors is selectable.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 10_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMNMIE115	ECMNMIE114	ECMNMIE113	ECMNMIE112	ECMNMIE111	ECMNMIE110	ECMNMIE109	ECMNMIE108	ECMNMIE107	ECMNMIE106	ECMNMIE105	ECMNMIE104	ECMNMIE103	ECMNMIE102	ECMNMIE101	ECMNMIE100
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 48.16 ECMNMICFG1 register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is read. When written, write the value after reset.
15 to 0	ECMNMIE115 to ECMNMIE100	ECM FE level non-maskable interrupt generation control bit ECMNMIE115 to ECMNMIE100 correspond to error sources 47 to 32. 0: Interrupt generation disabled 1: Interrupt generation enabled

48.2.12 ECM Internal Reset Configuration Register 0 (ECMIRCFG0)

The ECM internal reset configuration register 0 is used to set the generation of internal resets in response to internal errors.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 14_H

Initial value: 0000 000F_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMIRE031	ECMIRE030	ECMIRE029	ECMIRE028	ECMIRE027	ECMIRE026	ECMIRE025	ECMIRE024	ECMIRE023	ECMIRE022	ECMIRE021	ECMIRE020	ECMIRE019	ECMIRE018	ECMIRE017	ECMIRE016
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMIRE015	ECMIRE014	ECMIRE013	ECMIRE012	ECMIRE011	ECMIRE010	ECMIRE009	ECMIRE008	ECMIRE007	ECMIRE006	ECMIRE005	ECMIRE004	ECMIRE003	ECMIRE002	ECMIRE001	ECMIRE000
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 48.17 ECMIRCFG0 register contents

Bit Position	Bit Name	Function
31 to 0	ECMIRE031 to ECMIRE000	ECM internal reset generation control bit ECMIRE031 to ECMIRE000 correspond to error sources 31 to 0. 0: Internal reset generation disabled 1: Internal reset generation enabled

48.2.13 ECM Internal Reset Configuration Register 1 (ECMIRCFG1)

The ECM internal reset configuration register 1 is used to set the generation of internal resets in response to internal errors.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 18_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	ECMIRE129	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMIRE115	ECMIRE114	ECMIRE113	ECMIRE112	ECMIRE111	ECMIRE110	ECMIRE109	ECMIRE108	ECMIRE107	ECMIRE106	ECMIRE105	ECMIRE104	ECMIRE103	ECMIRE102	ECMIRE101	ECMIRE100
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 48.18 ECMIRCFG1 register contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is read. When written, write the value after reset.
29	ECMIRE129	ECM internal reset control bit Corresponds to delay timer overflow. 0: Internal reset generation disabled 1: Internal reset generation enabled
28 to 16	Reserved	When read, the value after reset is read. When written, write the value after reset.
15 to 0	ECMIRE115 to ECMIRE100	ECM internal reset generation control bit ECMIRE115 to ECMIRE100 correspond to error sources 47 to 32. 0: Internal reset generation disabled 1: Internal reset generation enabled

48.2.14 ECM Error Mask Register 0 (ECMEMK0)

The ECM error mask register 0 is used to mask the individual error sources of the error pin output.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + $1C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMEMK031	ECMEMK030	ECMEMK029	ECMEMK028	ECMEMK027	ECMEMK026	ECMEMK025	ECMEMK024	ECMEMK023	ECMEMK022	ECMEMK021	ECMEMK020	ECMEMK019	ECMEMK018	ECMEMK017	ECMEMK016
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMEMK015	ECMEMK014	ECMEMK013	ECMEMK012	ECMEMK011	ECMEMK010	ECMEMK009	ECMEMK008	ECMEMK007	ECMEMK006	ECMEMK005	ECMEMK004	ECMEMK003	ECMEMK002	ECMEMK001	ECMEMK000
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 48.19 ECMEMK0 register contents

Bit Position	Bit Name	Function
31 to 0	ECMEMK031 to ECMEMK000	ECM error output signal mask control bit ECMEMK031 to ECMEMK000 correspond to error sources 31 to 0. 0: Error signal output not masked 1: Error signal output masked

48.2.15 ECM Error Mask Register 1 (ECMEMK1)

The ECM error mask register 1 is used to mask the individual error sources of the error pin output.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	ECMEMK129	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMEMK115	ECMEMK114	ECMEMK113	ECMEMK112	ECMEMK111	ECMEMK110	ECMEMK109	ECMEMK108	ECMEMK107	ECMEMK106	ECMEMK105	ECMEMK104	ECMEMK103	ECMEMK102	ECMEMK101	ECMEMK100
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 48.20 ECMEMK1 register contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is read. When written, write the value after reset.
29	ECMEMK129	ECM error output signal mask control bit. 0: Error signal output not masked 1: Error signal output masked
28 to 16	Reserved	When read, the value after reset is read. When written, write the value after reset.
15 to 0	ECMEMK115 to ECMEMK100	ECM error output signal mask control bit ECMEMK115 to ECMEMK100 correspond to error sources 47 to 32. 0: Error signal output not masked 1: Error signal output masked

48.2.16 ECM Error Source Status Clear Trigger Register 0 (ECMESSTC0)

The ECM error source status clear trigger register 0 is a write-only register and can be written in 32-bit units. This register is used to clear the individual error source status of the ECM master/checker error source status register 0. Both the error status of the ECM master and the ECM checker are cleared simultaneously.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be written in 32-bit units.

Address: <ECM_base> + 24_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMCLSSE031	ECMCLSSE030	ECMCLSSE029	ECMCLSSE028	ECMCLSSE027	ECMCLSSE026	ECMCLSSE025	ECMCLSSE024	ECMCLSSE023	ECMCLSSE022	ECMCLSSE021	ECMCLSSE020	ECMCLSSE019	ECMCLSSE018	ECMCLSSE017	ECMCLSSE016
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMCLSSE015	ECMCLSSE014	ECMCLSSE013	ECMCLSSE012	ECMCLSSE011	ECMCLSSE010	ECMCLSSE009	ECMCLSSE008	ECMCLSSE007	ECMCLSSE006	ECMCLSSE005	ECMCLSSE004	ECMCLSSE003	ECMCLSSE002	ECMCLSSE001	ECMCLSSE000
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 48.21 ECMESSTC0 register contents

Bit Position	Bit Name	Function
31 to 0	ECMCLSSE031 to ECMCLSSE000	ECM error status clear bit ECMCLSSE031 to ECMCLSSE000 correspond to error sources 31 to 0. 0: Corresponding error status unchanged 1: Corresponding error status cleared

48.2.17 ECM Error Source Status Clear Trigger Register 1 (ECMESSTC1)

The ECM error source status clear trigger register 1 is a write-only register and can be written in 32-bit units. This register is used to clear the individual error source status of the ECM master/checker error source status register 1. Both the error status of the ECM master and the ECM checker are cleared simultaneously.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be written in 32-bit units.

Address: <ECM_base> + 28_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	ECMCLSSE130	ECMCLSSE129	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	W	W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMCLSSE115	ECMCLSSE114	ECMCLSSE113	ECMCLSSE112	ECMCLSSE111	ECMCLSSE110	ECMCLSSE109	ECMCLSSE108	ECMCLSSE107	ECMCLSSE106	ECMCLSSE105	ECMCLSSE104	ECMCLSSE103	ECMCLSSE102	ECMCLSSE101	ECMCLSSE100
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 48.22 ECMESSTC1 register contents

Bit Position	Bit Name	Function
31	Reserved	When read, the value after reset is read. When written, write the value after reset.
30, 29	ECMCLSSE130, ECMCLSSE129	ECM error status clear bit ECMCLSSE130, ECMCLSSE129 corresponds to ECMmSSE130, ECMmSSE129. 0: Error status unchanged 1: Error status cleared
28 to 16	Reserved	When read, the value after reset is read. When written, write the value after reset.
15 to 0	ECMCLSSE115 to ECMCLSSE100	ECM error status clear bit ECMCLSSE115 to ECMCLSSE100 correspond to ECMMSSE115 to ECMMSSE100. 0: Corresponding error status unchanged 1: Corresponding error status cleared

48.2.18 ECM Protection Command Register (ECMPCMD1)

The ECM protection command register is a write-only register and can be written in 32-bit units. Refer to Section 48.2.1, List of Registers, for the protected registers. Refer to Section 48.1.9, Writing to Protected Registers, for the details of the write protection sequence. The initial value is undefined.

Access: This register can be written in 32-bit units.

Address: <ECM_base> + $2C_H$

Initial value: Undefined

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	Undefined															
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	ECMREG1[7:0]							
Initial value	Undefined															
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 48.23 ECMPCMD1 register contents

Bit Position	Bit Name	Function
31 to 8	Reserved	When written, write the value after reset.
7 to 0	ECMREG1[7:0]	Protection command that enables writing to write protected ECM registers.

48.2.19 ECM Protection Status Register (ECMPS)

The ECM protection status register is a read-only register. This register is used to verify the write protected register has been written successfully or not. Refer to Section 48.1.9, Writing to Protected Registers.

Access: This register can be read in 8-bit units.

Address: <ECM_base> + 30_H

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ECMPRERR
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R

Table 48.24 ECMPS register contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is read. When written, write the value after reset.
0	ECMPRERR	ECM protection status bit Indicates whether writing to a write protected register failed or was successful. 0: Writing was successfully completed. 1: Writing failed

48.2.20 ECM Pseudo Error Trigger Register 0 (ECMPE0)

The ECM pseudo error trigger register 0 is a write-only register. This register is used to generate a pseudo error for test purposes. The ECM operation in response to the generation of a pseudo error is identical to that in response to a real error source.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be written in 8-, 16-, or 32-bit units.

Address: <ECM_base> + 34_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMPE031	ECMPE030	ECMPE029	ECMPE028	ECMPE027	ECMPE026	ECMPE025	ECMPE024	ECMPE023	ECMPE022	ECMPE021	ECMPE020	ECMPE019	ECMPE018	ECMPE017	ECMPE016
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMPE015	ECMPE014	ECMPE013	ECMPE012	ECMPE011	ECMPE010	ECMPE009	ECMPE008	ECMPE007	ECMPE006	ECMPE005	ECMPE004	ECMPE003	ECMPE002	ECMPE001	ECMPE000
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 48.25 ECMPE0 register contents

Bit Position	Bit Name	Function
31 to 0	ECMPE031 to ECMPE000	ECM pseudo error trigger bit ECMPE031 to ECMPE000 correspond to error sources 31 to 0. 0: Pseudo error not generated 1: Generates corresponding pseudo error generated

48.2.21 ECM Pseudo Error Trigger Register 1 (ECMPE1)

The ECM pseudo error trigger register 1 is a write-only register. This register is used to generate a pseudo error for test purposes. The ECM operation in response to the generation of a pseudo error is identical to that in response to a real error source.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be written in 8-, 16-, or 32-bit units.

Address: <ECM_base> + 38_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	ECMPE129	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	W	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMPE115	ECMPE114	ECMPE113	ECMPE112	ECMPE111	ECMPE110	ECMPE109	ECMPE108	ECMPE107	ECMPE106	ECMPE105	ECMPE104	ECMPE103	ECMPE102	ECMPE101	ECMPE100
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 48.26 ECMPE1 register contents

Bit Position	Bit Name	Function
31, 30	Reserved	When read, the value after reset is read. When written, write the value after reset.
29	ECMPE129	ECM pseudo error trigger bit Corresponds to delay timer overflow. 0: Pseudo error not generated 1: Generates corresponding pseudo error
28 to 16	Reserved	When read, the value after reset is read. When written, write the value after reset.
15 to 0	ECMPE115 to ECMPE100	ECM pseudo error trigger bit ECMPE115 to ECMPE100 correspond to error sources 47 to 32. 0: Pseudo error not generated 1: Generates corresponding pseudo error

48.2.22 ECM Delay Timer Control Register (ECMDTMCTL)

The ECM delay timer control register is a read/write register. This register is used to control the delay timer.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 8-bit units.

Address: <ECM_base> + $3C_H$

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	ECMSTP	ECMSTA
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	W	R/W

Table 48.27 ECMDTMCTL register contents

Bit Position	Bit Name	Function
7 to 2	Reserved	When read, the value after reset is read. When written, write the value after reset.
1	ECMSTP	Delay timer stop bit Writing 1 to this bit initializes the ECM delay timer register, causing the delay timer to stop. Simultaneously, the ECMSTA bit is set to 0.
0	ECMSTA	Delay timer start bit Specifies the operation of the overflow timer when an interrupt is occurred. 0: Timer stops 1: Timer in operation

48.2.23 ECM Delay Timer Register (ECMDTMR)

The ECM delay timer register is a read-only register. The ECM delay timer register is initialized by setting the ECMSTA bit of the ECM delay timer control register from 1 (timer in operation) to 0 (timer stops). ECMDTMR register indicates the current count value of the ECM delay timer.

Access: This register can be read in 16-bit units.

Address: <ECM_base> + 40_H

Initial value: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMDTMR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

48.2.24 ECM Delay Timer Compare Register (ECMDTMCMP)

The ECM delay timer compare register is a read/write register. The ECMmSSE129 bit is set when this register matches with the value of the ECM delay timer register. Writing data to this register has to be conducted while the delay timer is stopped.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 16-bit units.

Address: <ECM_base> + 44_H

Initial value: 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMDTMCMP[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

48.2.25 ECM Delay Timer Configuration Register 0 (ECMDTMCFG0)

The ECM delay timer configuration register 0 is used to set enable/disable of the delay timer start caused by FE level maskable interrupts in response to errors.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 48_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMTE031	ECMTE030	ECMTE029	ECMTE028	ECMTE027	ECMTE026	ECMTE025	ECMTE024	ECMTE023	ECMTE022	ECMTE021	ECMTE020	ECMTE019	ECMTE018	ECMTE017	ECMTE016
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMTE015	ECMTE014	ECMTE013	ECMTE012	ECMTE011	ECMTE010	ECMTE009	ECMTE008	ECMTE007	ECMTE006	ECMTE005	ECMTE004	ECMTE003	ECMTE002	ECMTE001	ECMTE000
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 48.28 ECMDTMCFG0 register contents

Bit Position	Bit Name	Function
31 to 0	ECMTE031 to ECMTE000	ECM delay timer start control bit ECMTE031 to ECMTE000 correspond to FE level maskable interrupts generated by error sources 31 to 0. 0: Delay timer start disabled 1: Delay timer start enabled

48.2.26 ECM Delay Timer Configuration Register 1 (ECMDTMCFG1)

The ECM delay timer configuration register 1 is used to set enable/disable of the delay timer start caused by FE level maskable interrupts in response to errors.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + $4C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMTE115	ECMTE114	ECMTE113	ECMTE112	ECMTE111	ECMTE110	ECMTE109	ECMTE108	ECMTE107	ECMTE106	ECMTE105	ECMTE104	ECMTE103	ECMTE102	ECMTE101	ECMTE100
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 48.29 ECMDTMCFG1 register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is read. When written, write the value after reset.
15 to 0	ECMTE115 to ECMTE100	ECM delay timer start control bit ECMTE115 to ECMTE100 correspond to FE level maskable interrupts generated by error sources 47 to 32. 0: Delay timer start disabled 1: Delay timer start enabled

48.2.27 ECM Delay Timer Configuration Register 2 (ECMDTMCFG2)

The ECM delay timer configuration register 2 is used to set enable/disable of the delay timer start caused by FE level non-maskable interrupts in response to errors.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 50_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECMTE231	ECMTE230	ECMTE229	ECMTE228	ECMTE227	ECMTE226	ECMTE225	ECMTE224	ECMTE223	ECMTE222	ECMTE221	ECMTE220	ECMTE219	ECMTE218	ECMTE217	ECMTE216
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMTE215	ECMTE214	ECMTE213	ECMTE212	ECMTE211	ECMTE210	ECMTE209	ECMTE208	ECMTE207	ECMTE206	ECMTE205	ECMTE204	ECMTE203	ECMTE202	ECMTE201	ECMTE200
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 48.30 ECMDTMCFG2 register contents

Bit Position	Bit Name	Function
31 to 0	ECMTE231 to ECMTE200	ECM delay timer start control bit ECMTE231 to ECMTE200 correspond to FE level non-maskable interrupts generated by error sources 31 to 0. 0: Delay timer start disabled 1: Delay timer start enabled

48.2.28 ECM Delay Timer Configuration Register 3 (ECMDTMCFG3)

The ECM delay timer configuration register 3 is a read/write register and can be written in 8-bit units. This register is used to set enable/disable of the delay timer start caused by FE level non-maskable interrupts in response to errors.

You have to follow a predetermined sequence for writing data to this register. Refer to Section 48.1.9, Writing to Protected Registers for the details of the write protection sequence.

Access: This register can be read/written in 32-bit units.

Address: <ECM_base> + 54_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECMTE315	ECMTE314	ECMTE313	ECMTE312	ECMTE311	ECMTE310	ECMTE309	ECMTE308	ECMTE307	ECMTE306	ECMTE305	ECMTE304	ECMTE303	ECMTE302	ECMTE301	ECMTE300
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 48.31 ECMDTMCFG3 register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the value after reset is read. When written, write the value after reset.
15 to 0	ECMTE315 to ECMTE300	ECM delay timer start control bit ECMTE315 to ECMTE300 correspond to FE level non-maskable interrupts generated by error sources 47 to 32. 0: Delay timer start disabled 1: Delay timer start enabled

48.2.29 ECM Error Pulse Control Register (ECMEPCTL)

This register selects the timer output in dynamic mode.

Access: This register can be read/written in 8-bit units.

Address: FFC0 600C$_H$

Initial value: 00$_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ECMTMSL
Initial value	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W

Table 48.32 ECMEPCTL Register Contents

Bit Position	Bit Name	Function
7 to 1	Reserved	When read, the value after reset is read. When written, write the value after reset.
0	ECMTMSL	Timer output selection to $\overline{\text{ERROROUT}}$ 0: Channel 15 in TAUB2 1: OSTM0

Section 49 Data CRC (DCRA)

This section contains a generic description of the data CRC function A (DCRA).

The first part in this section describes the RH850/D1L/D1Mspecific properties, such as the number of units, register base addresses, etc.

The remainder of the section describes the functions and registers of the DCRA.

49.1 Features of RH850/D1L/D1M DCRA

49.1.1 Number of Units

This microcontroller has the following number of DCRA units.

Table 49.1 Number of Units

Product Name	All products
Number of Units	1
Name	DCRAn (n = 0)

49.1.2 Register Base Address

DCRAn base addresses are listed in the following table.

DCRAn register addresses are given as offsets from the base addresses.

Table 49.2 Register Base Address

Base Address Name	Base Address
<DCRA0_base>	FFF7 0000_H

49.1.3 Clock Supply

The DCRAn clock supply is shown in the following table.

Table 49.3 Clock Supply

Unit Name	Unit Clock Name	Clock Supply Name
DCRAn	PCLK	C_ISO_PCLK

49.1.4 Reset Sources

DCRAn reset sources are listed in the following table. DCRAn is initialized by these reset sources.

Table 49.4 Reset Sources

Unit Name	Reset Source
DCRAn	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

49.2 Overview

49.2.1 Functional Overview

The data CRC function A can be used to verify or generate CRC protected data streams of arbitrary length and different bit widths.

- 32-bit Ethernet CRC
  $(X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X^{1}+1)$
- 16-bit CCITT CRC
  $(X^{16}+X^{12}+X^{5}+1)$
- CRC of an arbitrary data block length can be generated.
- After initialization of the CRC data register, every write access to the CRC input register generates a new CRC according to the selected polynomial, and the result is stored in the CRC data register.
- DCRA calculates CRC from least significant byte (LSB) to most significant byte (MSB).

49.2.2 Block Diagram

The following picture shows the block diagram of the data CRC function A.

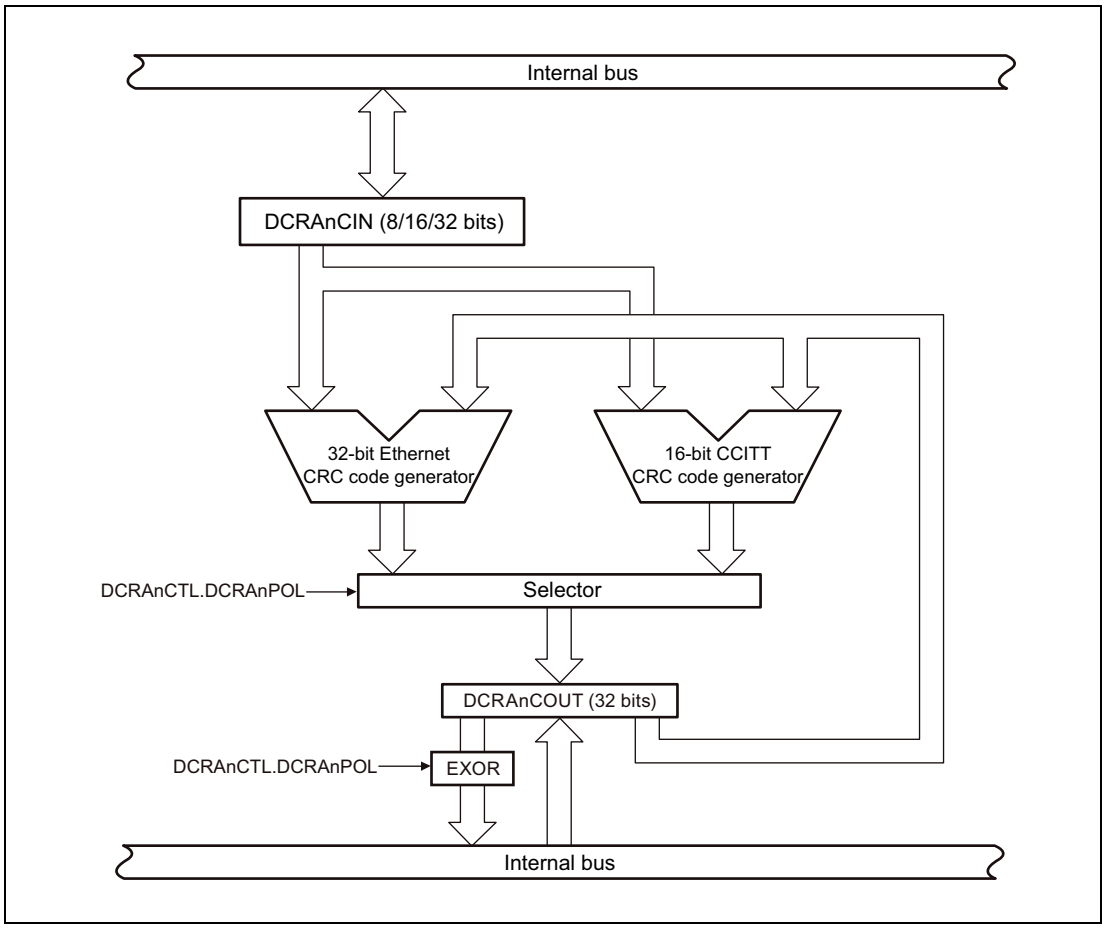
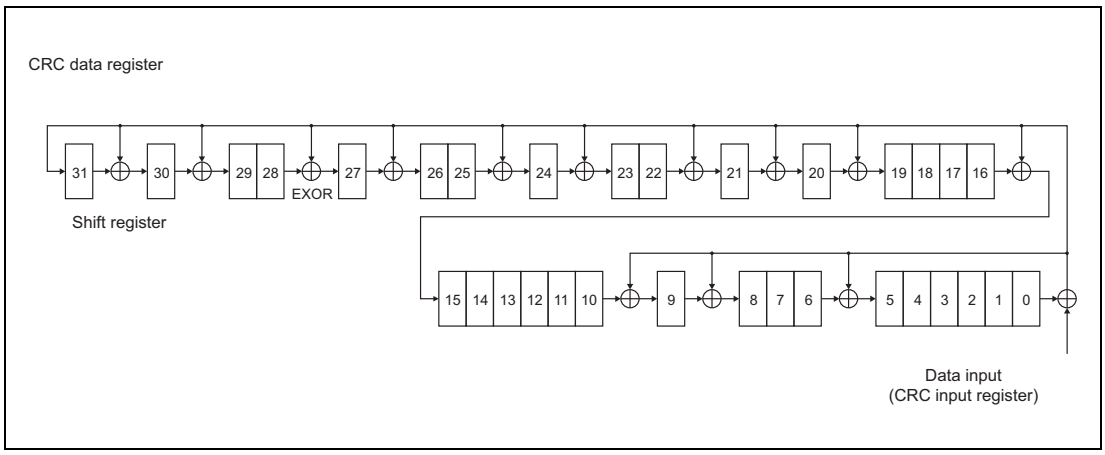


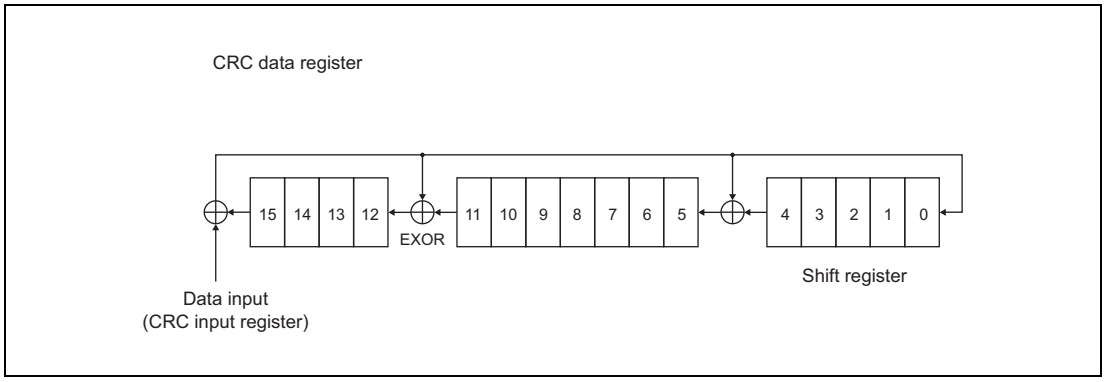
Figure 49.1 Block Diagram of Data CRC Function A

49.2.3 Operational Circuit

- 32-bit Ethernet



- 16-bit CCITT



49.3 Registers

49.3.1 List of Registers

DCRA registers are listed in the following table.

Table 49.5 List of Registers

Module Name	Register Name	Symbol	Address
DCRAn	CRC input register	DCRAnCIN	<DCRAn_base> + 00_H
DCRAn	CRC data register	DCRAnCOUT	<DCRAn_base> + 04_H
DCRAn	CRC control register	DCRAnCTL	<DCRAn_base> + 20_H

49.3.2 DCRAnCIN — CRC Input Register

This register holds the input data for CRC calculation. The effective bit width used for CRC calculation must be set by DCRAnCTL.DCRAnISZ[1:0].

When data is written to this register, the CRC code is generated.

The CRC calculation is immediately started after the DCRAnCIN register is written. The DCRAnCOUT register must be initialized with the initial starting value, before the first data of the data block is written to DCRAnCIN register.

Access: This register can be read/written in 32-bit units.

Address: <DCRAn_base>

Value after reset: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCRAnCIN[31:16]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCRAnCIN[15:0]															
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 49.6 DCRAnCIN Register Contents

Bit Position	Bit Name	Function
31 to 0	DCRAnCIN [31:0]	Input Data for CRC Calculation The valid bits are: • For 32 bit effective bit width: DCRAnCIN[31:0] • For 16 bit effective bit width: DCRAnCIN[15:0] • For 8 bit effective bit width: DCRAnCIN[7:0]

49.3.3 DCRAnCOUT — CRC Data Register

This register stores the result of the CRC code generated by the 32-bit Ethernet polynomial or the 16-bit CCITT polynomial.

Access: This register can be read/written in 32-bit units.

Address: <DCRAn_base> + 4_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCRAnCOUT[31:16]															
Value after reset *1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCRAnCOUT[15:0]															
Value after reset *1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note 1. The read value after reset is 0000 0000_H since the 32-bit Ethernet CRC polynomial is selected as the CRC generating function after reset.

Table 49.7 DCRAnCOUT Register Contents

Bit Position	Bit Name	Function
31 to 0	DCRAnCOUT [31:0]	Result of the CRC Code Generation When the 16-bit CCITT polynomial is enabled, the bits 15 to 0 show the CRC result. The bits 31 to 16 are undefined. The read value of this register is a value obtained by performing EXOR calculation for the following value: • For 32-bit Ethernet polynomial: FFFF $FFFF_H$ • For 16-bit CCITT polynomial: 0000_H For example, when DCRAnCOUT = 5555 5555_H for the 32-bit Ethernet polynomial, AAAA $AAAA_H$ is read.

CAUTION

This register must initialized by setting the initial start value before the first data of the data block is written to DCRAnCIN register.

49.3.4 DCRAnCTL — CRC Control Register

This register controls the CRC generation process.

Access: This register can be read/written in 8-bit units.

Address: <DCRAn_base> + 20_H

Value after reset: 00_H

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	DCRAnISZ[1:0]		DCRAnPOL
Value after reset	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W

Table 49.8 DCRAnCTL Register Contents

Bit Position	Bit Name	Function
7 to 3	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
2, 1	DCRAnISZ[1:0]	Specifies the CRC input bit width: 00: 32 bits (DCRAnCIN[31:0]) 01: 16 bits (DCRAnCIN[15:0]) 10: 8 bits (DCRAnCIN[7:0]) 11: Setting prohibited
0	DCRAnPOL	Specifies the CRC generation method: 0: 32-bit Ethernet CRC polynomial generation. The byte order of the DCRAnCIN register is LSB (least significant bit) first. This means that, if the input bit width is 8 bits (DCRAnISZ[1:0] = 10_B), bit positions 7 to 0 of the DCRAnCIN register contain the input data and bit position 0 (LSB) is the start bit of the input data. 1: 16-bit CCITT CRC polynomial generation. The byte order of the DCRAnCIN register is MSB (most significant bit) first. This means that, if the input bit width is 8 bits (DCRAnISZ[1:0] = 10_B), bit positions 7 to 0 of the DCRAnCIN register contain the input data and bit position 7 (MSB) is the start bit of the input data.

CAUTION

- If the CRC generation method (DCRAnCTL.DCRAnPOL) is changed, the DCRAnCOUT register must be initialized by setting the initial start value.
- The CRC bit width (DCRAnCTL.DCRAnISZ[1:0]) must be set according to the data block bit width. Changing the CRC bit width is not allowed during processing of a data block (a data block consists of N bytes, half-words or one word). After the final CRC result is read from DCRAnCOUT register, the bit width can be changed. In that case, the DCRAnCOUT register must be initialized with the initial start value.

49.4 Operation

The data CRC function A generates a CRC (cyclic redundancy check) of an arbitrary data block length. The data is forwarded to the data CRC function in 8-, 16- or 32-bit units. The CRC polynomial can either be selected for 32-bit Ethernet or 16-bit CCITT. The initial starting value must be set at the DCRAnCOUT register before the first write access to the CRC input register (DCRAnCIN) is performed.

The flowchart below shows the CRC generating procedure.

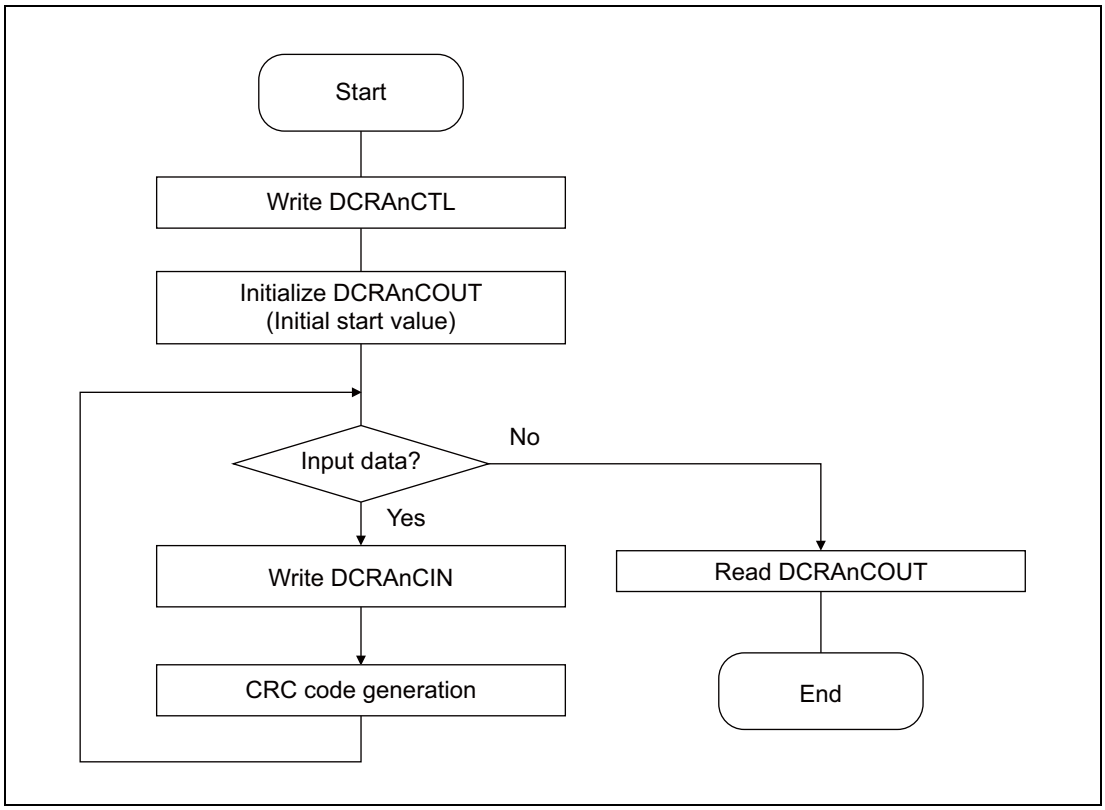


Figure 49.2 Flowchart of Data CRC Function A

NOTES

1. Before writing the first data to DCRAnCIN, the CRC output register DCRAnCOUT must be initialized with the initial start value.
2. DCRAnCOUT must be re-initialized by setting the initial start value when the polynomial is changed by changing DCRAnCTL.DCRAnPOL.
3. Setting example of the initial start values of the respective polynomials
   The following is the example of setting values.

Table 49.9 Setting Example of Initial Start Values (When Read at a Reset)

	Initial Start Value	EXOR Value	DCRAnCOUT Read Value
16-bit CCITT	XXXX $FFFF_H$	XXXX 0000_H	XXXX $FFFF_H$
32-bit Ethernet	$FFFF\ FFFF_H$	$FFFF\ FFFF_H$	$0000\ 0000_H$

Note: X: undefined

Section 50 Intelligent Cryptographic Unit (ICU-S2)

The contents of this section is available upon non-disclosure agreement.

For details, contact your local sales representatives.

Section 51 On-Chip Debug Unit (OCD)

This microcontroller has an on-chip debug function. By using the on-chip debug emulator, programs can be debugged with the microcontroller mounted in the target system.

The debug functions incorporated in this microcontroller conform to IEEE-ISTO 5001™-2003 Class 1, a Nexus debug interface standard.

CAUTION

The debug functions described in this section are supported by the microcontroller but whether they are usable depends on the debugger. For details on debugging, see the user's manual of the debugger.

51.1 Overview of RH850/D1L/D1M OCD

51.1.1 Channels

This microcontroller has the following number of channels of the OCD.

Table 51.1 Channels of OCD

	OCD
Number of channels	1
Name	OCD0

51.1.2 Interrupts

The On-Chip Debug Unit can generate the following interrupt request:

Table 51.2 OSTMn interrupt requests

OCDn signals	Connected to
Hot-attach interrupt	Interrupt Controller INTDCUTDI

51.2 Overview

The on-chip debug functions described below are supported by the microcontroller.

Whether they are usable depends on the specifications of the debugger being used.

For details on on-chip debug functions, see the user's manual of the debugger.

(1) Debug interface

This microcontroller supports as debug interfaces: NEXUS Interface, Low Pin Debug Interface (1-pin) - hereinafter called “LPD (1-pin)”, and Low Pin Debug Interface (4-pin) - hereinafter called “LPD (4-pin)”.

On-chip debug can be performed using these debug interfaces.

(2) Debug monitoring function

Debug-dedicated monitor program space is mounted and is used during debugging.

The basic debug functions below can be used by running a monitoring program.

- downloading the user-created program
- reading and writing the memory and registers
- running the user-created program starting at any address

(3) On-chip break

A maximum of 12 breakpoints can be specified to any execution address. Of the 12 breakpoints, a maximum of four breakpoints can be specified to any access (access address, access data).

(4) Software break

Software break points can be specified at any execution address.
Up to 2000 software break points can be added to any execution address in the code flash.
Note that the usage of software break points in the code flash area leads to re-programming.

(5) Peripheral Break

Peripheral Break is a function that stops peripheral modules when a debugger has stopped operation of the microcontroller.

During peripheral break, internal modules operate as follows.

a. Modules that stop unconditionally

Table 51.3 Modules with unconditional peripheral break

Module
Window Watchdog Timer A (WDTAn)

b. Modules that optionally support functions to stop or to continue

Table 51.4 Modules with optional peripheral break

Module	Control register bit	n
A/D Converter (ADCE)	ADCEnEMU.SVSDIS 0: STOP during break 1: continue during break	0
PWM Generators and Diagnostic (PWM-Diag)	PWBAnEMU.PWBAnSVSDIS 0: STOP during break 1: continue during break	0
OS Timer (OSTM)	OSTMnEMU.OSTMnSVSDIS 0: STOP during break 1: continue during break	0, 1
Always-On-Area Timer (AWOT)	AWOTnEMU.AWOTnSVSDIS 0: STOP during break 1: continue during break	0
Timer Array Unit B (TAUB)	TAUBnEMU.TAUBnSVSDIS 0: STOP during break 1: continue during break	0 to 2
Timer Array Unit J (TAUJ)	TAUJnEMU.TAUJnSVSDIS 0: STOP during break 1: continue during break	0
Real-Time Clock (RTCA)	RTCAnEMU.RTCAnSVSDIS 0: STOP during break 1: continue during break	0
Clocked Serial Interface G (CSIG)	CSIGnEMU.CSIGnSVSDIS 0: STOP during break 1: continue during break	0 to 3
Clocked Serial Interface H (CSIH)	CSIHnEMU.CSIHnSVSDIS 0: STOP during break 1: continue during break	4, 5
Sound Generator (SG)	SGnEMU.SGnSVSDIS 0: STOP during break 1: continue during break	0 to 4
PCM-PWM Converter (PCMP)	PCMPnEMU.PCMPnSVSDIS 0: STOP during break 1: continue during break	0
A/D Converter E (ADCE)	ADCEnEMU.ADCEnSVSDIS 0: STOP during break 1: continue during break	0
LCD Bus Interface (LCBI)	LCBInEMU.LCBInSVSDIS 0: STOP during break 1: continue during break	0
Stepper Motor Driver (ISM)	ISMnEMU.ISMnSVDIS 0: STOP during break 1: continue during break	0

CAUTION

For details on the registers, see the register description of the corresponding section.

c. Modules that stop when EPC.SVSTOP = 1

Table 51.5 Modules that stop when EPC.SVSTOP = 1

Module
LIN / UART Interface (RLIN3)
Serial Sound Interface (SSIF)

(6) Forced break

Execution of the user-created program can be interrupted forcibly.

(7) Forced reset

This device can be forcibly reset.

(8) Real time RAM monitoring (RRM)

The memory can be read during program execution. Because this read access uses debug-dedicated DMA, it has minimal effect on program execution.

(9) Dynamic memory modification (DMM)

The memory can be written during program execution. Because this write access uses debug-dedicated DMA, it has minimal effect on program execution.

(10) Timer function

Using a 32-bit counter, the time for running the user-created program can be measured based on the clock for debug.

For the measurement accuracy, see the user's manual of the debugger.

(11) Mask function

Masking the following factors are possible.

- All reset factors except for a POC reset and a wake-up reset

(12) Hot plug-in function

Debugging can be started in normal operating mode without external reset input.

NOTE

When the hot plug-in function is used in power save mode, the INTDCUTDI interrupt is required to return from power save mode as the wake-up handling.

(13) Security function

To prevent the contents of the flash memory from being read by an unauthorized person, a 128-bit ID code can be written to the microcontroller. If the code the user inputs when starting a debugger does not match the ID code written to the microcontroller, the flash memory cannot be accessed.

For details on how to set the ID code, see the user's manual of the debugger.

51.3 Registers

51.3.1 Peripheral Break Control

The peripheral break function generates a stop request to the peripheral modules of the microcontroller if the user-created program is stopped, for unit upon a breakpoint hit.

The peripheral modules stop their operation if all of the following conditions are met:

- Peripheral modules support the peripheral break function.
  (For the list of peripheral modules that support peripheral break, see **51.2** (5) Peripheral Break.)
- Peripheral break function is enabled in the peripheral modules.
  (For the setting of emulation registers each peripheral module has, see **51.2** (5) Peripheral Break.)
- Peripheral break function itself is enabled by setting EPC.SVSTOP = 1.
  (Refer to the EPC register description below.)

51.3.1.1 EPC — Emulation Peripheral Control Register

This register stops macros (timer, serial interface, and A/D converter) in debug mode (SVSTOP).

Access: Accessing from the user program is prohibited.

Address: —

Initial value: 00_H

Bit	7	6	5	4	3	2	1	0
	—	SVSTOP	—	—	—	—	—	—
Initial value	0*1	0	0	0	0	0	0	0
R/W	—	—	—	—	—	—	—	—

Note 1. When accessed from debugger, this bit is read as 1.

Table 51.6 EPC register contents

Bit Position	Bit Name	Function
7	Reserved	—
6	SVSTOP	Stops operation of macros (timer, serial interface, and A/D converter) during debugging. 0: Does not stop operation 1: Stops operation
5 to 0	Reserved	—

NOTES

1. EPC is set by the debugger. Setting by the user program is prohibited. For the setting of the debugger, see the user's manual of the debugger.
2. In the header files the module name of the above register is defined as: SYSCTRL.

51.3.2 Temporary On-Chip Debug enable

On-Chip debugging can be temporarily enabled in normal operation mode, i.e. by the user's software.

51.3.2.1 IDMODI — On-Chip Debug Control Register

This register allows to enable/disable On-Chip Debugging in normal operation mode.

Access: This register can be written in 32-bit units.
Writing to this register is protected by a special sequence of instructions by using the protection command register PROTCMDIDMODI. For details, refer to Section 4, Write-Protected Registers.

Address: FFC0 6030_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IDEN	IDDATA
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W	W

Table 51.7 IDMODI register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the value after reset is read. When written, write the value after reset.
1	IDEN	Enable/disable On-Chip debugging
0	IDDATA	

IDEN	IDDATA	Function
0	X	OCD ID authentication change not possible
1	0	OCD ID authentication disabled
	1	OCD ID authentication enabled

NOTE

In the header files the module name of the above register is defined as:

SELB.

51.4 Caution on Using On-Chip Debugging

51.4.1 Treatment of devices using debugging

Do not use a device that was used for system debugging on a mass-produced product, because the flash memory was rewritten during system debugging and thus the write/erase endurance of the flash memory cannot be guaranteed.

51.4.2 Cautions when Shifting to DEEPSTOP Mode or Setting reset during Debugger use

When using a debugger, executing a program that causes the transition to DEEPSTOP mode (see Section 13, Stand-by Controller (STBC)) or the setting of any reset (see Section 9, Reset Controller) immediately after the program is started may cause improper communication between the OCD emulator and microcontroller because the microcontroller will enter DEEPSTOP mode or initialized before the preparations for communication between the OCD emulator and microcontroller are completed.

The communication preparation period depends on the OCD emulator's host PC environment and the operating frequency of microcontroller, so when performing debugging that causes the program to enter DEEPSTOP mode or to set any reset immediately after the program starts, insert waits in the interval between debugger reset release ($\overline{\text{DBRES}}$) and these execution instruction so that the debugger starts normally.

NOTES

1. The recommended interval is 50 ms for the RH850/D1L/D1M.
2. The resets related to this issue are all of described in 9.1, Overview.
3. In DEEPSTOP mode, the debugging controller stops. For return from DEEPSTOP mode by the debugger, see Section 13.1.2.1, Wake-up factors.

Section 52 Flash Memory

The RH850/D1L/D1M incorporates code flash memory and data flash memory.

Series Name	Product Name	Code Flash Memory	Data Flash Memory
D1L1	R7F701401, R7F701421	2 Mbytes	64 Kbytes
D1L2	R7F701402, R7F701422	4 Mbytes	64 Kbytes
D1L2H	R7F701403, R7F701423		
D1M1_3.75MB	R7F701404	3.75 Mbyte	64 Kbytes
D1M1H_3.75MB	R7F701406		
D1M2_3.75MB	R7F701408, R7F701428		
D1M2H_3.75MB	R7F701411, R7F701431		
D1M1-V2	R7F701442, R7F701462	4 MB	64 Kbytes
D1M1A	R7F701441, R7F701461		
D1M1_5MB	R7F701405	5 Mbyte	64 Kbytes
D1M1H_5MB	R7F701407		
D1M2_5MB	R7F701410, R7F701430		
D1M2H_5MB	R7F701412, R7F701432		

52.1 Features

- Code flash memory capacity: Up to 5 Mbytes of user area and up to an additional 32 Kbytes of extended user area
- Data flash memory capacity: 64 Kbytes for data area
- Methods of programming
  - Programming by communicating with the dedicated flash memory programmer via the serial interface (serial programming)
  - Flash memory programming by a user program (self-programming)
- Support for security functions to protect against illicit tampering with or reading out of data in flash memory
- Support for protection functions to protect against erroneous overwriting of the flash memory
- Support for the detection and correction of errors in the flash memory
- Support for the BGO (Back Ground Operation) function
  - Code flash memory can be read while data flash memory is being programmed
- The initial settings of RH850/D1L/D1M can be configured in the extended area (option bytes) of flash memory

52.2 Memory Configuration

User area in code flash memory of RH850/D1L/D1M is divided into 8 Kbytes or 32 Kbytes blocks, which can be erased individually.

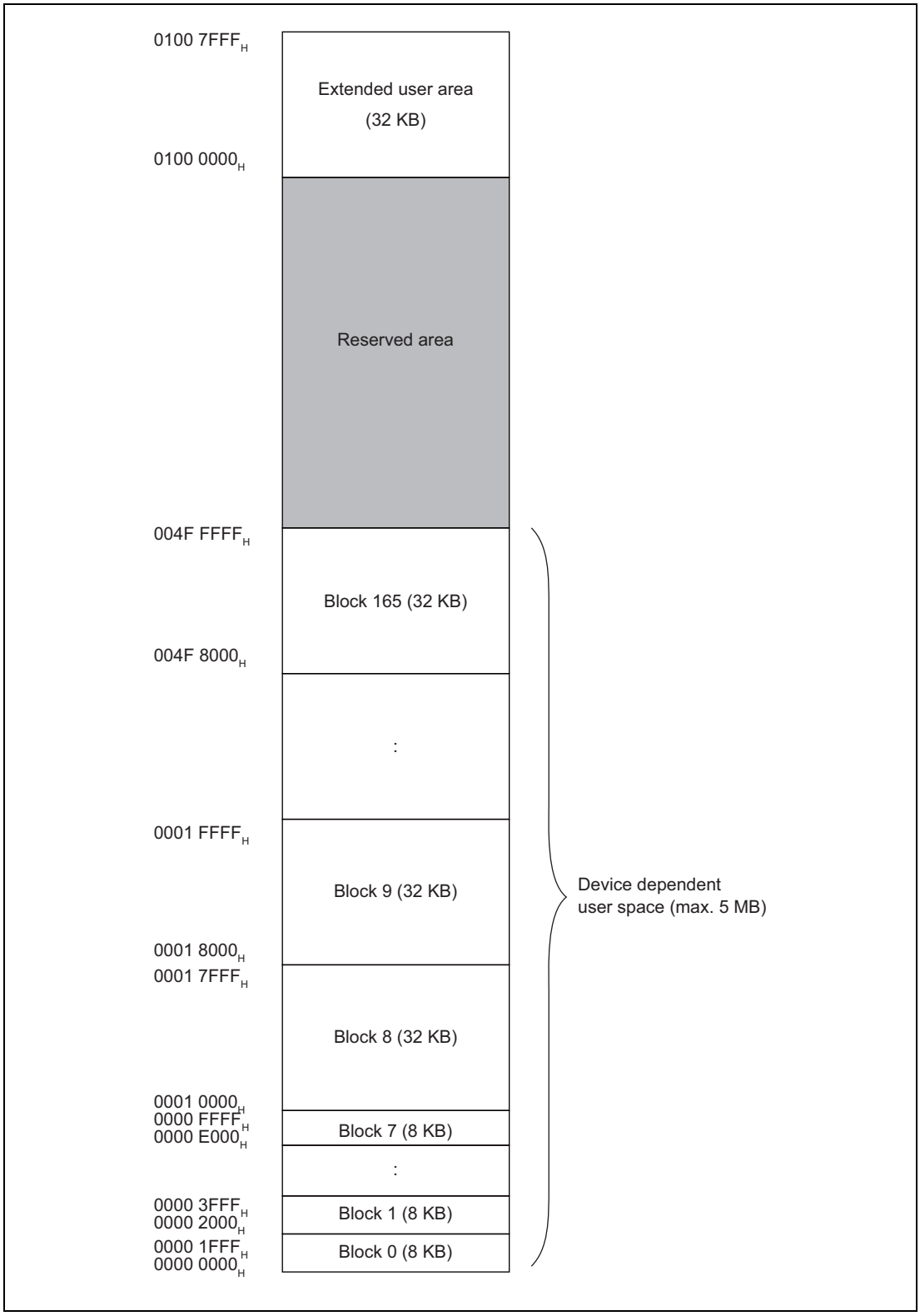


Figure 52.1 Code Flash Memory Mapping

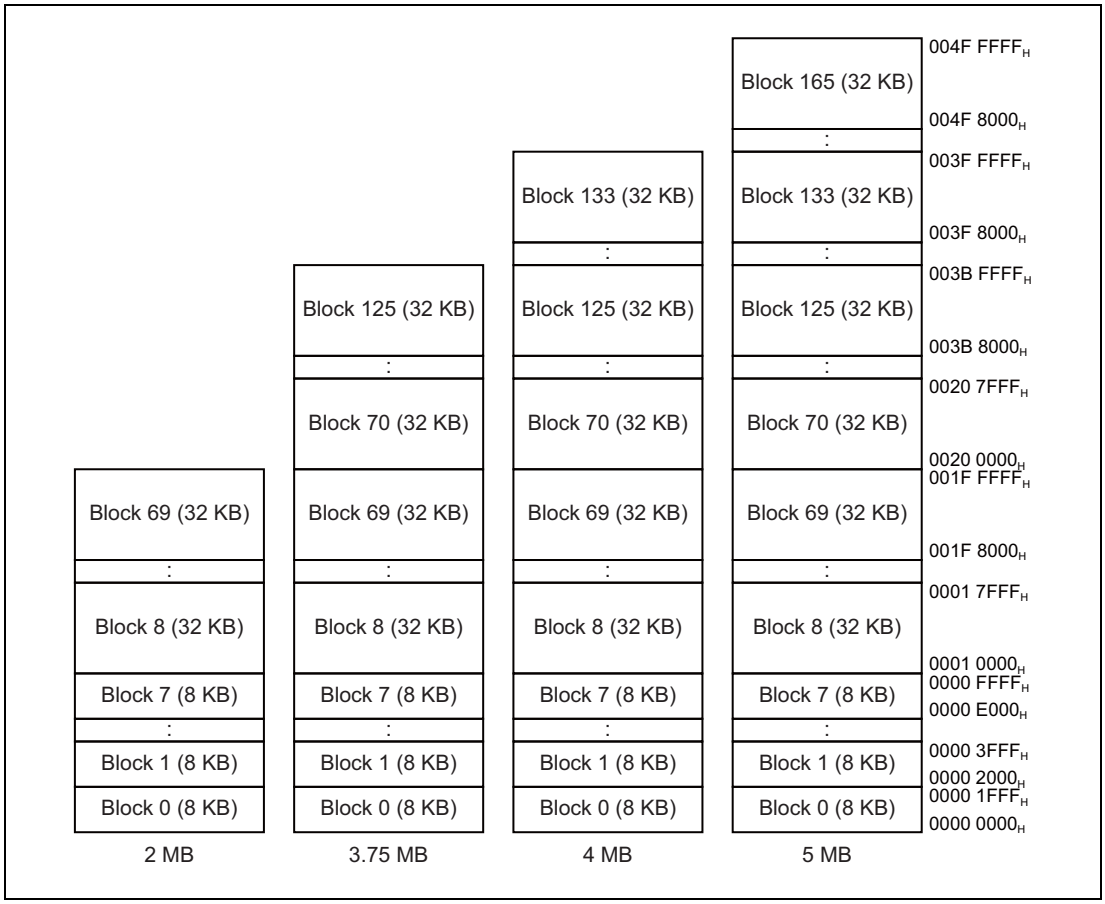


Figure 52.2 Relation between Memory Capacity and Blocks of the User Space

Data area in data flash memory of RH850/D1L/D1M is divided into 64 bytes blocks, which can be erased individually.

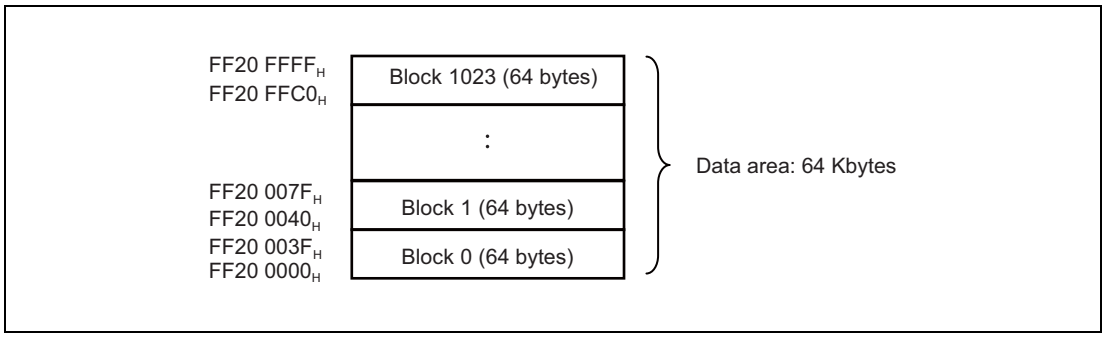


Figure 52.3 Data Flash Memory Mapping

52.3 Operating Modes Associated with Flash Memory

Figure 52.4 is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, refer to Section 6, Operating Modes.

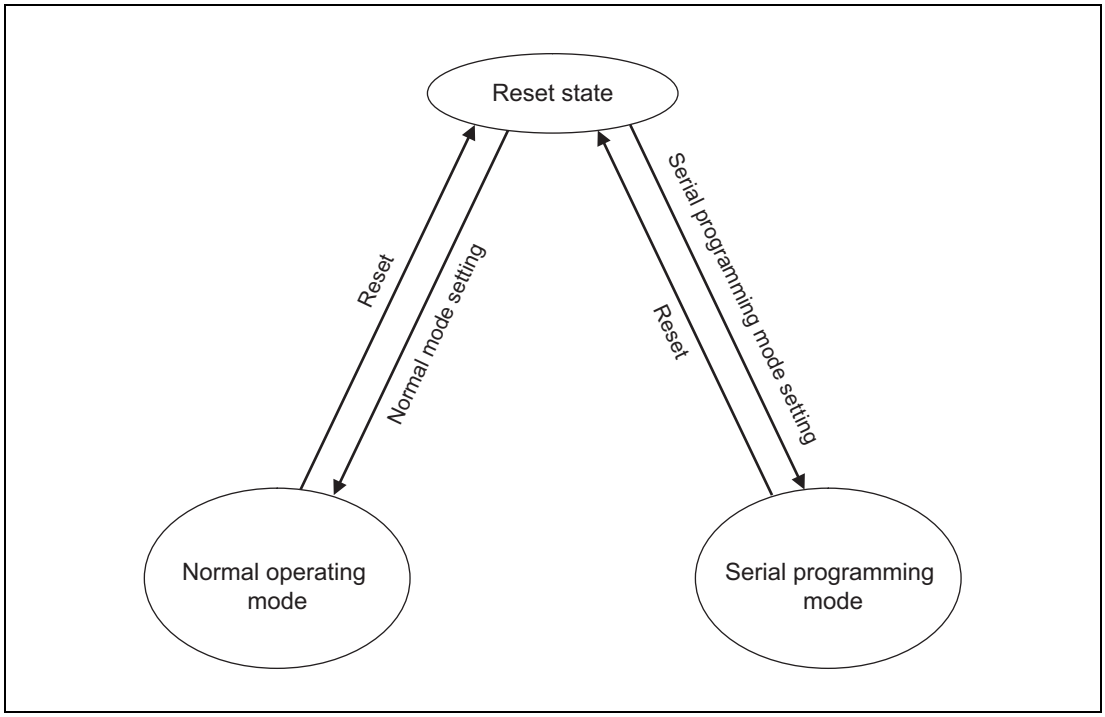


Figure 52.4 Mode Transition Associated with Flash Memory

The flash memory areas which are programmable and erasable and the boot program after a reset depend on the selected mode. The differences between modes are indicated in **Table 52.1**.

Table 52.1 Differences between Modes

Item	Normal Operating Mode	Serial Programming Mode
Programmable and erasable area	• User area • Extended user area • Data area	• User area • Extended user area • Data area
Boot program at a reset	Program in user area or extended user area	Firmware program for serial programming

52.4 Functional Overview

On-chip flash memory of RH850/D1L/D1M can be programmed regardless of before and after the mounting to the target system with the programming function that employs the dedicated flash memory programmer (serial programming).

In addition, a security function that prohibits the programming of a user program written in on-chip flash memory is supported to address the protection against falsification of programs by outsiders.

The programming function using a user program (self-programming) is the method suitable for applications where modifying the program after the production or shipment of the target system is expected. A protection function for secure programming to flash memory area is also supported. Furthermore, programming can be conducted under various conditions, such as in parallel with communicating with outside, by utilizing the support for interrupt processing during self-programming. **Table 52.2** gives an overview of the methods of programming and the corresponding operating modes.

Table 52.2 Programming Methods

Programming Method	Functional Overview	Operating mode
Serial programming	A dedicated flash-memory programmer is capable of on-board programming the flash memory after the device is mounted on the target system.	Serial programming mode
	A dedicated flash memory programmer and dedicated programming adapter board allow off-board programming of the flash memory, i.e. programming of the device before it is mounted on the target system.	
Self-programming	Flash memory can be programmed by executing a user program preprogrammed in code flash memory with serial programming. When data flash memory is being programmed with self-programming, the BGO function enables instruction fetch and data read from code flash memory. Thus, data flash memory can be programmed by executing a program on code flash memory prepared for flash programming. When code flash memory is being programmed with self-programming, instruction fetch and data read from code flash memory are prohibited. This programming needs to be carried out by executing a program prepared for flash programming that has been transferred to Local RAM in advance.	Normal operating mode

Table 52.3 lists the functions of the on-chip flash memory. Serial programmer commands realize serial programming, while reading of the on-chip flash memory by a library function or the user program realizes self-programming.

Table 52.3 Basic Functions at a Glance

Function	Description in Overview	Level of Support (√: Supported, △: Conditionally Supported, ×: Not Supported)	
		Serial programming	Self-programming
Blank checking	This is used to check a specified block to ensure that writing to it has not already proceeded. Results of reading from code flash memory and data flash memory to which nothing has been written after erasure are not guaranteed, so use blank checking to confirm that writing to memory has not proceeded after erasure.	√	√
Block erasure	This is for erasing the contents of a specified block of memory.	√	√
Programming	This is for writing to a specified address.	√	√
Verification and checksum	Data that are read out from flash memory are compared with data transferred from the flash memory programmer.	√	× (reading of data by the user program is possible)
Reading	Data that have been written to the flash memory are read out.	√	√
Setting for OTP (one-time programming)	A specified block of code flash memory is set for OTP (OTP can only be set, that is, it is not possible to release a block's OTP setting).	√	√
Setting an ID	An ID setting is made for use in controlling the connection of a serial programmer for serial programming, and programming of the code flash memory by self-programming.	√	√
Security settings	Security settings are for use in serial programming.	√	△ (only when setting is prohibited after being permitted)
Protection settings	Lock bits for all blocks of code flash memory are provided.	√	√
Setting of option bytes	Option bytes are set to change them from the initial values for the RH850/D1L/D1M	√	√
Clearing the configuration	ID setting, security settings, protection settings, and option byte settings are initialized.	√	×

For details on serial programming, refer to "*PG-FP5 Flash Memory Programmer Users' Manual*" and "*Renesas Flash Programmer Flash Programming Software User's Manual"*.

For details on self-programming, refer to the user's manuals for the code flash library and data flash library which this device targets.

The OTP setting and authentication of the ID code are security functions for use with serial programming and self-programming.

In serial programming, authentication of the ID code, prohibiting connection of a serial programmer, and prohibition of commands (for block erasure, programming, and reading) are available for use as security functions.

Table 52.4 Summary of Security Functions

Function	Description
OTP	OTP can be individually set for each block of the user area and the extended user area of code flash memory. When the OTP setting is made for an area, programming by serial programming and by self programming is prohibited. Once set, the OTP setting cannot be released. Furthermore, since execution of the configuration clearing command is prohibited for any area for which OTP has been set, changing a security setting from "prohibited" to "permitted" is not possible in D1M1A and D1M1-V2. For the RH850/D1L/D1M products except D1M1A and D1M1-V2, execution of the configuration clearing command is not prohibited. Thus, changing a security setting from "prohibited" to "permitted" is possible.
ID authentication	The result of ID authentication can be used to control the connection of a serial programmer for serial programming. The result of ID authentication is also used for enabling/disabling of code flash self-programming. Re-programing of the data flash in self-programming is always possible without any authentication.
Prohibition of connection of a serial programmer	The connection of a serial programmer for serial programming is prohibited. Since execution of the configuration clearing command is also prohibited when the connection of a serial programmer is prohibited, changing a security setting from "prohibited" to "permitted" is not possible.
Prohibition of block erasure commands	Block erasure commands at the time of serial programming are prohibited. Since execution of the configuration clearing command is also prohibited when block erasure commands are prohibited, changing a security setting from "prohibited" to "permitted" is not possible.
Prohibition of programming commands	Block erasure commands and programming commands at the time of serial programming are prohibited. Only through execution of the configuration clearing command can the prohibition be lifted.
Prohibition of read commands	Read commands at the time of serial programming are prohibited. Only through execution of the configuration clearing command can the prohibition be lifted.

Table 52.5 Available Operations and Security Settings

Function	All Security Settings and Erasure, Programming, and Read Operations (√: Executable, ×: Not Executable, —: Not Supported)		Point for Caution Regarding the Security Setting	
	Serial programming	Self programming	Serial programming	Self programming
OTP	• Areas for which OTP is set – Block erasure commands: × – Programming commands: × – Read commands: √ • Areas for which OTP is not set – Block erasure commands: √ – Programming commands: √ – Read commands: √	• Areas for which OTP is set – Block erasure: × – Programming: × – Reading: √ • Areas for which OTP is not set – Block erasure: √ – Programming: √ – Reading: √	• The OTP setting cannot be released. • D1M1A, D1M1-V2: Execution of the configuration clearing command is not possible. • Other products: Execution of the configuration clearing command is possible.	The OTP setting cannot be released.
ID authentication	• When the ID codes do not match – Block erasure commands: × – Programming commands: × – Read commands: × • When the ID codes match – Block erasure commands: √ – Programming commands: √ – Read commands: √	• When the ID codes do not match – Code flash memory - Block erasure: × - Programming: × - Reading: √ – Data flash memory - Block erasure: √ - Programming: √ - Reading: √ • When the ID codes match – Block erasure: √ – Programming: √ – Reading: √	• The configuration clearing command can initialize the setting for prohibition. • The setting for prohibition of block erasure commands is not available. • The setting for prohibition of programming commands is not available. • The setting for prohibition of read commands is not available.	ID authentication is always in effect.
Prohibition of the connection of a serial programmer	• Block erasure commands: × • Programming commands: × • Read commands: ×	• Block erasure: √ • Programming: √ • Reading: √	Since execution of the configuration clearing command is prohibited, initialization of the setting for prohibition is not possible.	Initialization of the settings prohibited is impossible because the configuration clearing command is not supported.
Prohibition of block erasure commands	• Block erasure commands: × • Programming commands: √ • Read commands: √	• Block erasure: √ • Programming: √ • Reading: √	• Since execution of the configuration clearing command is prohibited, initialization of the setting for prohibition is not possible. • The setting for ID authentication to be effective for serial programming is not available.	Initialization of the settings prohibited is impossible because the configuration clearing command is not supported.
Prohibition of programming commands	• Block erasure commands: ×*1 • Programming commands: × • Read commands: √	• Block erasure: √ • Programming: √ • Reading: √	• Executing the configuration clearing command only can initialize the settings prohibited. • The setting for ID authentication to be effective for serial programming is not available.	Initialization of the settings prohibited is impossible because the configuration clearing command is not supported.
Prohibition of read commands	• Block erasure commands: √ • Programming commands: √ • Read commands: ×	• Block erasure: √ • Programming: √ • Reading: √		

Note 1. Block erase command can be used only in ascending order.
Code Flash (block 0→ 1→ 2... 165)
→ Extended user area
→ Data Flash (block 0→ 1→ 2... 1023)

Table 52.6 Summary of Protection Functions

Function	Description
Block protection	Block protection can be individually made to enable or disable programming and erasure of each block of the user area and the extended user area of code flash memory. Programming and erasure by self programming of an area for which the lock bit is set and the lock bit function is enabled are prohibited. Programming or erasure can proceed again when the lock bit function is disabled after having been enabled. When a block erasure command is executed, the lock bit for that block is also erased.
Hardware protection	The level on the FLMD pin can be set to prohibit programming and erasure of the code flash memory in self programming mode. - FLMD0 = 0: Programming prohibited - FLMD0 = 1: Programming permitted
Variable reset vector	The protection settings include control of the reset vector. As shown in **Figure 52.5**, after programming of a new boot program while leaving the existing boot program in place, changing the reset vector is a safe way to change to the area holding the new boot program.

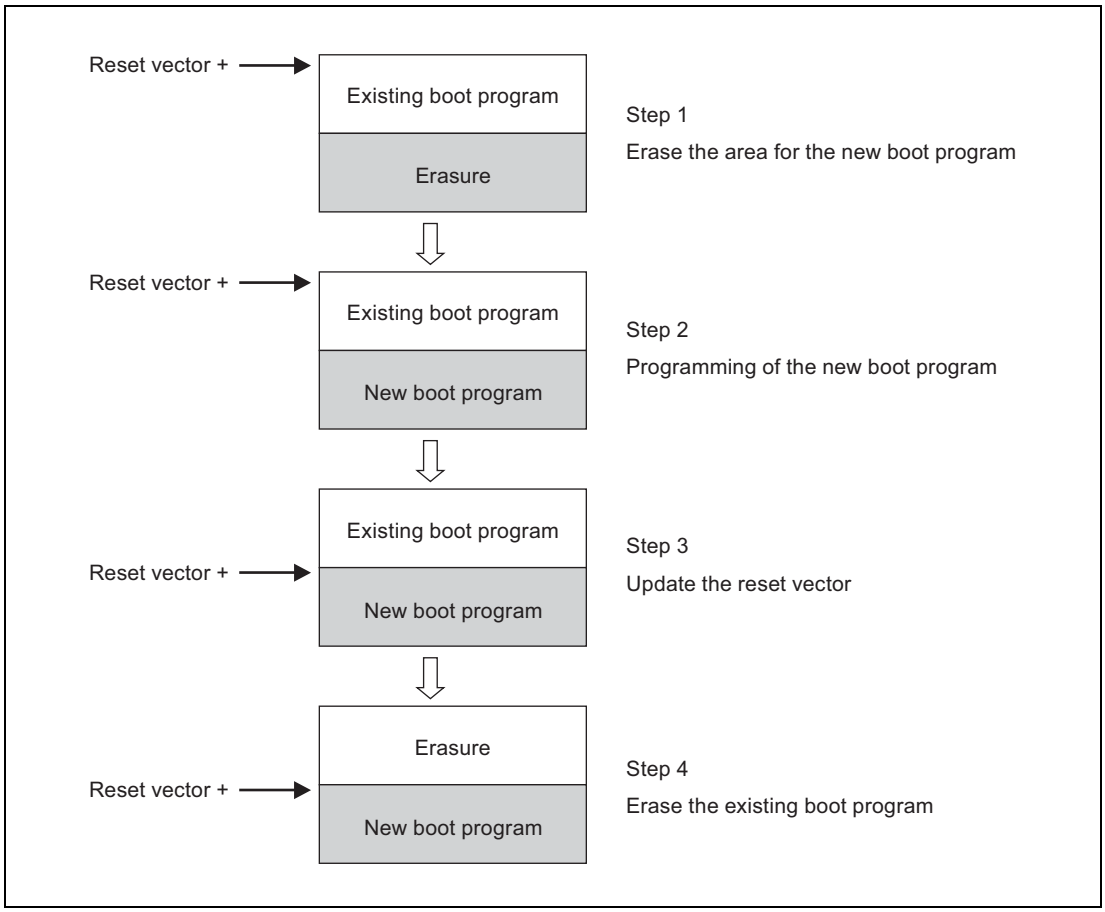


Figure 52.5 Utilizing the variable reset vector function to update the boot program

52.5 Communications for Serial Programming Mode

52.5.1 One-wire UART as an Asynchronous Flash Programming Interface

To use a 1-wire UART as a single-wire asynchronous serial programming interface, connect the following pins and the flash memory programmer.

- FLSCI3RXD, FLSCI3TXD(FPDR)/JP0_0: Receive data input/transmit data output

52.5.2 Two-wire UART as an Asynchronous Flash Programming Interface

To use a 2-wire UART as a double-wire asynchronous serial programming interface, connect the following pins and the flash memory programmer.

- FLSCI3RXD(FPDR)/JP0_0: Receive data input
- FLSCI3TXD(FPDT)/JP0_1: Transmit data output

52.5.3 CSI as a Synchronous Flash Programming Interface

To use a CSI as a synchronous serial programming interface, connect the following pins and the flash memory programmer.

- FLSCI3RXD (FPDR)/JP0_0: Receive data input
- FLSCI3TXD (FPDT)/JP0_1: Transmit data output
- FLSCI3SCKI (FPCK)/JP0_2: Serial clock input

The flash memory programmer outputs the serial data clock signal (SCK) and the microcontroller operates as a slave.

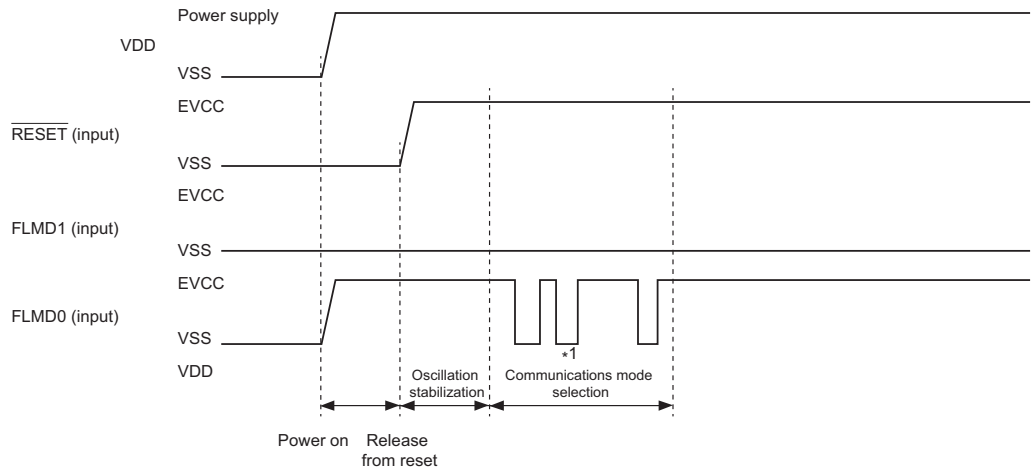
NOTE

For details on the flash programming software (Renesas Flash Programmer), refer to the "*Renesas Flash Programmer Flash Programming Software User's Manual*".

52.5.4 Selecting the Communications System

In the RH850/D1L/D1M, the communications system is selected by the input of pulses (up to 7) to the FLMD0 pin after the chip shifts to the flash memory programming mode. These pulses are generated by the dedicated flash memory programmer.

The relationship between the communications system and the number of pulses is shown below.



Note 1. The number of input clocks depends on the communications system as shown below.

Pulses on FLMD0	Communications System	Notes
0	2 wire UART	
7	1 wire UART	
3	CSI	
	Others	Setting prohibited

Figure 52.6 Selecting the Communications Mode

52.6 Programming with Serial Programmer

Flash memory can be programmed in serial programming mode with the dedicated flash memory programmer.

Serial Programming

In serial programming, the microcontroller is mounted on the board. Mounting the connecter on the board allows the flash memory programmer to program the target microcontroller.

52.6.1 Programming Environment

The figure below shows the environment recommended for programming data to flash memory in the microcontroller.

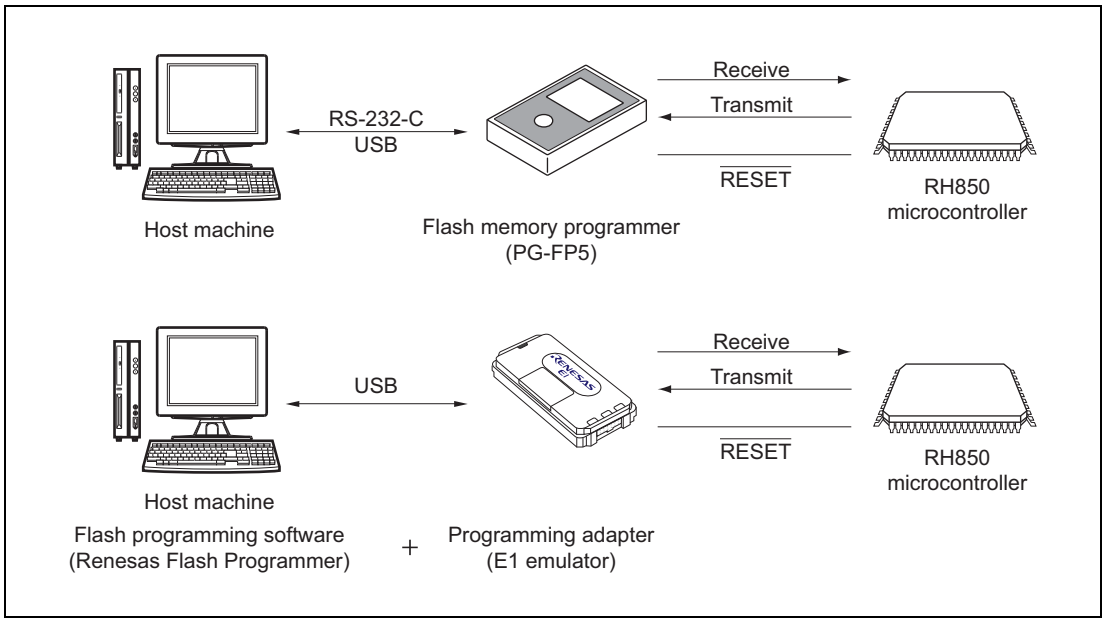


Figure 52.7 Environment for Programming Flash Memory

Using PG-FP5 flash memory programmer or Renesas Flash Programmer flash programming software (running on the host machine) in combination with E1 emulator used as the programming adaptor, you can easily conduct programming operations, such as erasure, programming, and verification, to Renesas Electronics' microcontroller with on-chip flash memory mounted on the user's board without the need for unmounting it.

PG-FP5 flash memory programmer supports programming from the host machine or programming in the standalone mode. The flash programming software (Renesas Flash Programmer) supports programming from the host machine.

NOTE

Refer to *PG-FP5 flash memory programmer user's manual* for the details of PG-FP5, and *Renesas Flash Programmer flash programming software user's manual* for the details of Renesas Flash Programmer flash programming software.

52.7 Programming with Self-programming

52.7.1 Overview

RH850/D1L/D1M supports the flash memory programming of user program itself.

When programming data flash memory, you can execute a program on code flash memory prepared for flash programming using the BGO function to program data flash memory. Instead, you can also execute a program prepared for flash programming that has been transferred to Local RAM in advance to program data flash memory.

In addition, you can execute a program prepared for flash programming that has been transferred to Local RAM in advance to program code flash memory.

52.7.2 BGO Function

When the combination of the flash memory to be programmed and the flash memory to be read meets the conditions in the following list, the BGO function can be used.

Table 52.7 Conditions Required to Use BGO Function

Area to Be Programmed	Area to Be Read
Data flash memory	Code flash memory

52.7.3 Enabling of Self-Programming

The self-programming function can be activated in normal operating mode.

Erasure and programming of the flash memory by the self-programming function is enabled by making the FLMD0 pin high level.

Having FLMD0 pin low level prevents unnecessary overwriting of the program if the device operates incorrectly.

The FLMD0 pin is made high level by using one of the following methods.

- The FLMD0 pin is externally pulled up.
- The FLMD0 pin is pulled up by the FLMDCNT register.

The outline of the FLMDCNT register is described in Section 52.7.3.1, FLMDCNT Register.

52.7.3.1 FLMDCNT Register

This register specifies the internal pull-up or pull-down of the FLMD0 pin.

Access: This register can be read/written in 32-bit units.
Writing to this register is protected by a special sequence of instructions by using the protection command register FLMDPCMD. For details, see Section 4, Write-Protected Registers.

Address: FFA0 0000$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FLMDPUP
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 52.8 FLMDCNT Register Contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the value after reset is returned. When writing to these bits, write the value after reset.
0	FLMDPUP	FLMD0 Pin Software Control 0: Pull-down selected 1: Pull-up selected

NOTE

In the header files the module name of the above register is defined as "FLMD", and the register name as "CNT".

52.8 Flash Memory Read

52.8.1 Code Flash Memory Read

Special settings are not required to read code flash memory in normal mode. Data can simply be read out through access to addresses in the code flash memory.

Reading from an area of code flash memory that has been erased but not yet been programmed again (i.e. that is in the non-programmed state) can lead to the detection of an ECC error and generation of the corresponding exception. Furthermore, since the values of data cannot be guaranteed when an ECC error has been generated, use blank checking when you need to confirm that an area is in the non-programmed state.

52.8.2 Data Flash Memory Read

Configure the number of read cycles in the EEPRDCYCL register prior to reading data from data flash memory in normal mode. Once this register is properly configured, data can be read by simply accessing addresses in the data flash memory. Once the setting for the number of cycles is made, data can simply be read out through access to addresses in the data flash memory.

Values read from data flash memory that has been erased but not yet been programming again are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

52.9 Description of Registers

52.9.1 Registers Related to Data Flash Memory

Table 52.9 shows the list of registers related to data flash memory.

Table 52.9 List of Registers Related to Data Flash Memory

Register Name	Symbol	R/W	Initial Value	Address	Access Size
Data flash memory read cycle setting register	EEPRDCYCL	R/W	$0F_H$	FFC5 9810_H	8

52.9.1.1 Data Flash Memory Read Cycle Setting Register (EEPRDCYCL)

This register sets the read cycle of data flash memory.

Access: This register can be read/written in 8-bit units.

Address: FFC5 9810_H

Initial value: $0F_H$

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	FRDCYCLD[3:0]			
Initial value	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 52.10 EEPRDCYCL register contents

Bit Position	Bit Name	Function
7 to 4	Reserved	
3 to 0	FRDCYCLD[3:0]	Number of data flash memory read cycles Data flash memory is read in setting value + 1 cycles. 0x0 to 0x6: Setting prohibited 0x7: Read cycle 8 0x8: Read cycle 9 0x9 to 0xF: Read cycle 10

NOTE

In the header files the module name of the above register is defined as:
DCIB.

52.9.2 Registers Related to Product Information

The following tables show the list of registers related to product information.

Table 52.11 List of Registers Related to Product Information (1/4)

Register Name	Symbol	R/W	Initial Value					
			R7F701401	R7F701402	R7F701403	R7F701404	R7F701405	R7F701406
			D1L1	D1L2	D1L2H	D1M1_3.75M	D1M1_5M	D1M1H_3.75M
Product name storage register (1)	PRDNAME1	R	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$
Product name storage register (2)	PRDNAME2	R	3034 3130$_H$	3034 3130$_H$	3034 3130$_H$	3034 3130$_H$	3034 3130$_H$	3034 3130$_H$
Product name storage register (3)	PRDNAME3	R	2020 2031$_H$	2020 2032$_H$	2020 2033$_H$	2020 2034$_H$	2020 2035$_H$	2020 2036$_H$
Product name storage register (4)	PRDNAME4	R	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$

Table 52.12 List of Registers Related to Product Information (2/4)

Register Name	Symbol	R/W	Initial Value					
			R7F701407	R7F701408	R7F701410	R7F701411	R7F701412	R7F701421
			D1M1H_5M	D1M2_3.75M	D1M2_5M	D1M2H_3.75M	D1M2H_5M	D1L1
Product name storage register (1)	PRDNAME1	R	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$
Product name storage register (2)	PRDNAME2	R	3034 3130$_H$	3034 3130$_H$	3134 3130$_H$	3134 3130$_H$	3134 3130$_H$	3234 3130$_H$
Product name storage register (3)	PRDNAME3	R	2020 2037$_H$	2020 2038$_H$	2020 2030$_H$	2020 2031$_H$	2020 2032$_H$	2020 2031$_H$
Product name storage register (4)	PRDNAME4	R	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$

Table 52.13 List of Registers Related to Product Information (3/4)

Register Name	Symbol	R/W	Initial Value					
			R7F701422	R7F701423	R7F701428	R7F701430	R7F701431	R7F701432
			D1L2	D1L2H	D1M2_3.75M	D1M2_5M	D1M2H_3.75M	D1M2H_5M
Product name storage register (1)	PRDNAME1	R	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$
Product name storage register (2)	PRDNAME2	R	3234 3130$_H$	3234 3130$_H$	3234 3130$_H$	3334 3130$_H$	3334 3130$_H$	3334 3130$_H$
Product name storage register (3)	PRDNAME3	R	2020 2032$_H$	2020 2033$_H$	2020 2038$_H$	2020 2030$_H$	2020 2031$_H$	2020 2032$_H$
Product name storage register (4)	PRDNAME4	R	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$

Table 52.14 List of Registers Related to Product Information (4/4)

Register Name	Symbol	R/W	Initial Value			
			R7F701441	R7F701461	R7F701442	R7F701462
			D1M1A	D1M1A	D1M1-V2	D1M1-V2
Product name storage register (1)	PRDNAME1	R	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$	3746 3752$_H$
Product name storage register (2)	PRDNAME2	R	3434 3130$_H$	3634 3130$_H$	3434 3130$_H$	3634 3130$_H$
Product name storage register (3)	PRDNAME3	R	2020 2031$_H$	2020 2031$_H$	2020 2032$_H$	2020 2032$_H$
Product name storage register (4)	PRDNAME4	R	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$	2020 2020$_H$

52.9.2.1 Product Name Storage Register (PRDNAMEn, n = 1 to 4)

This register stores the product name. The product model name is stored in 16-byte ASCII code, and PRDNAME1, PRDNAME2, PRDNAME3, and PRDNAME4 correspond to the fourth to first bytes, eighth to fifth bytes, twelfth to ninth bytes, and sixteenth to thirteenth bytes of the product model name respectively.

Access: This register can be read in 32-bit units.

Address: PRDNAME1: FFCD 00D0$_H$
PRDNAME2: FFCD 00D4$_H$
PRDNAME3: FFCD 00D8$_H$
PRDNAME4: FFCD 00DC$_H$

Initial value: See Table 52.15, Contents of Product Name Storage Register.

Bit	31	30	29	28	27	26	25	24
	PRDNAMEn[31:24]*1							
Initial value								
R/W	R	R	R	R	R	R	R	R
Bit	23	22	21	20	19	18	17	16
	PRDNAMEn[23:16]*1							
Initial value								
R/W	R	R	R	R	R	R	R	R
Bit	15	14	13	12	11	10	9	8
	PRDNAMEn[15:8]*1							
Initial value								
R/W	R	R	R	R	R	R	R	R
Bit	7	6	5	4	3	2	1	0
	PRDNAMEn[7:0]*1							
Initial value								
R/W	R	R	R	R	R	R	R	R

Note 1. n = 1 to 4

Table 52.15 Contents of Product Name Storage Register

Bit Position	Bit Name	Function
31 to 24	—	Product name fourth byte (PRDNAME1), eighth byte (PRDNAME2), twelfth byte (PRDNAME3), sixteenth byte (PRDNAME4)
23 to 16	—	Product name third byte (PRDNAME1), seventh byte (PRDNAME2) eleventh byte (PRDNAME3), fifteenth byte (PRDNAME4)
15 to 8	—	Product name second byte (PRDNAME1), sixth byte (PRDNAME2) tenth byte (PRDNAME3), fourteenth byte (PRDNAME4)
7 to 0	—	Product name first byte (PRDNAME1), fifth byte (PRDNAME2) ninth byte (PRDNAME3), thirteenth byte (PRDNAME4)

NOTE

In the header files the module name of the above register is defined as:
SCDS0.

52.9.2.2 Chip ID Register

The unique chip ID can be read out of device registers.

Module name	Register name	Register address	R/W	Access size	Initial value
SCDS0	CHIPID1LL	FFCD 00E0$_H$	R	32 bit	Unique code
SCDS0	CHIPID1LH	FFCD 00E4$_H$	R	32 bit	Unique code
SCDS0	CHIPID1HL	FFCD 00E8$_H$	R	32 bit	Unique code
SCDS0	CHIPID1HH	FFCD 00EC$_H$	R	32 bit	Unique code
SCDS0	CHIPID2LL	FFCD 00F0$_H$	R	32 bit	Unique code
SCDS0	CHIPID2LH	FFCD 00F4$_H$	R	32 bit	Unique code
SCDS0	CHIPID2HL	FFCD 00F8$_H$	R	32 bit	Unique code
SCDS0	CHIPID2HH	FFCD 00FC$_H$	R	32 bit	Unique code

NOTE

In the header files the module name of the above register is defined as:

SCDS0.

52.10 Option Bytes

The flash memory has the extended area (option bytes) to store data specified by the user for various purposes.

52.10.1 Option Byte 0 Register (OPBT0)

Changes in settings such as initial setting of peripheral functions using Option Byte 0 become effective after release from the reset state.

Access: This register can be read in 32-bit units.
This register can only be written in serial programming mode and self-programming mode.

Address: FFCD 0030_H

Initial value: Specified by the user

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OPJTAG[1:0]	
Initial value	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	WDT1 OPWD VAC	WDT1 OPWD TPR	WDT1 OPWD RUN	WDT1 OPWD EN	WDT1 OPWD OVF[2:0]			—	WDT0 OPWD VAC	WDT0 OPWD TPR	WDT0 OPWD RUN	WDT0 OPWD EN	WDT0 OPWD OVF[2:0]		
Initial value	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note 1. Be sure to set 1.

Table 52.16 OPBT0 register contents (1/2)

Bit Position	Bit Name	Function
31 to 18	Reserved	(The write value should be 1.)
17 to 16	OPJTAG[1:0]	Controls the functions of the JTAG port group JP0. 00: JP0 used for general purpose/alternative functions ports 01: JP0 used as LPD 4-pin mode 10: JP0 used as LPD 1-pin mode 11: JP0 used as Nexus I/F
15	Reserved	(The write value should be 1.)
14	WDT1OPWD VAC	Enables/disables the Variable Activation Code function (VAC) of WDTA1 0: VAC is disabled 1: VAC is enabled
13	WDT1OPWD TPR	WDTA1 start mode signal selector: 0: OPWDRUN start-up option 1: WDTATRTYP input signal
12	WDT1OPWD RUN	Specifies the start mode of WDTA1: 0: Software trigger start mode 1: Automatic start mode
11	WDT1OPWD EN	Enables/disables WDTA1: 0: WDTA1 is disabled 1: WDTA1 is enabled

Table 52.16 OPBT0 register contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>WDT1OPWD OVF[2:0]</td><td>These bits select the overflow time.
<table>
<tr><th>OPWDOVF2</th><th>OPWDOVF1</th><th>OPWDOVF0</th><th>Overflow time</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>2^9 / WDTATCKI</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2^{10} / WDTATCKI</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2^{11} / WDTATCKI</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>2^{12} / WDTATCKI</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>2^{13} / WDTATCKI</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>2^{14} / WDTATCKI</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>2^{15} / WDTATCKI</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>2^{16} / WDTATCKI</td></tr>
</table></td></tr>
<tr><td>7</td><td>Reserved</td><td>(The write value should be 1.)</td></tr>
<tr><td>6</td><td>WDT0OPWD VAC</td><td>Enables/disables the Variable Activation Code function (VAC) of WDTA0
0: VAC is disabled
1: VAC is enabled</td></tr>
<tr><td>5</td><td>WDT0OPWD TPR</td><td>WDTA0 start mode signal selector:
0: OPWDRUN start-up option
1: WDTATRTYP input signal</td></tr>
<tr><td>4</td><td>WDT0OPWD RUN</td><td>Specifies the start mode of WDTA0:
0: Software trigger start mode
1: Automatic start mode</td></tr>
<tr><td>3</td><td>WDT0OPWD EN</td><td>Enables/disables WDTA0:
0: WDTA0 is disabled
1: WDTA0 is enabled</td></tr>
<tr><td>2 to 0</td><td>WDT0OPWD OVF[2:0]</td><td>These bits select the overflow time.
<table>
<tr><th>OPWDOVF2</th><th>OPWDOVF1</th><th>OPWDOVF0</th><th>Overflow time</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>2^9 / WDTATCKI</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2^{10} / WDTATCKI</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2^{11} / WDTATCKI</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>2^{12} / WDTATCKI</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>2^{13} / WDTATCKI</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>2^{14} / WDTATCKI</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>2^{15} / WDTATCKI</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>2^{16} / WDTATCKI</td></tr>
</table></td></tr>
</table>

NOTE

In the header files the module name of the above register is defined as:
SCDS0.

Table 52.17 Debug Interface

OPJTAG1	OPJTAG0	Mode	JP0_0	JP0_1	JP0_2	JP0_3	JP0_4	JP0_5
1	1	Nexus I/F	DCUTDI input	DCUTDO output	DCUTCK input	DCUTMS input	DCUTRST input	DCURDY output
0	1	LPD (4-pin)	LPDI input	LPDO output	LPDCLK input	Port/ alternative function	Port/ alternative function	LPDCLK OUT output
1	0	LPD (1-pin)	LPDIO input/ output	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function	Port/ alternative function

52.10.2 Option Byte 9 Register (OPBT9)

Access: This register can be read in 32-bit units.

Address: FFCD 0164_H

Initial value: Result of the device production test

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	SDRTRC[31:16]															
Initial value	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SDRTRC[15:0]															
Initial value	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 52.18 OPBT9 register contents

Bit Position	Bit Name	Function
31 to 0	SDRTRC[31:0]	DDR2-SDRAM Memory Controller (SDRB) trimming code lower bits

NOTE

In the header files the module name of the above register is defined as:
SCDS0.

52.10.3 Option Byte 10 Register (OPBT10)

Access: This register can be read in 32-bit units.

Address: FFCD 0168_H

Initial value: Result of the device production test.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TSNREFDH[11:0]												TSNREFDL[11:8]			
Initial value	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TSNREFDL[7:0]								SDRTRC[39:32]							
Initial value	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 52.19 OPBT10 register contents

Bit Position	Bit Name	Function
31 to 20	TSNREFDH [11:0]	Temperature sensor A/D conversion reference value R_{150} for 150 °C
19 to 8	TSNREFDL [11:0]	Temperature sensor A/D conversion reference value R_{25} for 25 °C
7 to 0	SDRTRC[39:32]	DDR2-SDRAM Memory Controller (SDRB) trimming code upper bits

NOTE

In the header files the module name of the above register is defined as:
SCDS0.

52.11 Code Flash ECC

The ECC of the Code Flash generates an interrupt request listed in the following table:

Table 52.20 Interrupt Request by ECC of Code Flash

Function	Connected to
ECC 1-bit error interrupt during access by the CPU	These interrupts are logically OR combined and input to the Error Control Module INTECCSFLI.
ECC 1-bit error interrupt during access by a GVCI master	
ECC 2-bit error interrupt during access by the CPU	These interrupts are logically OR combined and input to the Error Control Module INTECCDFLI.
ECC 2-bit error interrupt during access by a GVCI master	

The 9-bit ECC value is calculated for each 128-bit transfer.

The ECC of the Code Flash and its registers are initialized by the following reset signal:

Table 52.21 Reset sources

ECC unit	Reset signal
Code Flash ECC	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

52.11.1 Overview

The code flash ECC is summarized in the table below.

Item	Description
ECC error detection and correction	ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected. • ECC error detection and correction are carried out (2-bit error detection and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit error detection, correction and notification and 2-bit error detection and notification are carried out.
Error notification	Upon occurrence of an ECC error, it is notified to the Error Control Module. **ECC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is disabled upon detection of an ECC 1-bit error. The error notification signal is issued to the ECM.
Error status	A status register is provided, which indicates the statuses of ECC 2-bit error detection and ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Address capture	If an ECC error occurs while no error status is set, the address at which the associated error has occurred is captured. The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected. The error status serves as the enable bit of the capture address.

52.11.2 List of Registers

Registers with "_VCI2CFB_OS" appendix are for code flash access from GVCI masters.

Registers with "_PE1_OS" appendix are for code flash access from CPU(PE1).

Table 52.22 List of Registers

Module Name	Register Name	Symbol	Address
ECCFLI	Code flash ECC control register (VCI)	CFECCCTL_VCI2CFB_OS	FFC6 2200_H
ECCFLI	Code flash error information control register (VCI)	CFERRINT_VCI2CFB_OS	FFC6 2204_H
ECCFLI	Code flash status clear register (VCI)	CFSTCLR_VCI2CFB_OS	FFC6 2208_H
ECCFLI	Code flash error count overflow status register (VCI)	CFOVFSTR_VCI2CFB_OS	FFC6 $220C_H$
ECCFLI	Code flash 1st error status register (VCI)	CF1STERSTR_VCI2CFB_OS	FFC6 2210_H
ECCFLI	Code flash 1st error address register (VCI)	CF1STEADR0_VCI2CFB_OS	FFC6 2250_H
ECCFLI	Code Flash sub-test control register (VCI)	CFSTSTCTL_VCI2CFB_OS	FFC6 2350_H
ECCFLI	Code flash ECC control register (PE1)	CFECCCTL_PE1_OS	FFC6 2400_H
ECCFLI	Code flash error information control register (PE1)	CFERRINT_PE1_OS	FFC6 2404_H
ECCFLI	Code flash status clear register (PE1)	CFSTCLR_PE1_OS	FFC6 2408_H
ECCFLI	Code flash error count overflow status register (PE1)	CFOVFSTR_PE1_OS	FFC6 $240C_H$
ECCFLI	Code flash 1st error status register (PE1)	CF1STERSTR_PE1_OS	FFC6 2410_H
ECCFLI	Code flash 1st error address register (PE1)	CF1STEADR0_PE1_OS	FFC6 2450_H
ECCFLI	Code Flash sub-test control register (PE1)	CFSTSTCTL_PE1_OS	FFC6 2550_H

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

52.11.3 Details of Registers

52.11.3.1 CFECCCTL_VCI2CFB/PE1_OS — Code flash ECC control register

CFECCCTL enables or disables ECC error detection and correction and 1-bit error correction. Set PROT[1:0] bits to 01_B when writing to CFECCCTL.

Access: This register can be read/written in 32-bit units.

Address: CFECCCTL_VCI2CFB_OS: FFC6 2200_H
CFECCCTL_PE1_OS: FFC6 2400_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT[1:0]		—	—	—	—	—	—	—	—	—	—	—	—	SEC DIS	ECC DIS
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 52.23 CFECCCTL_VCI2CFB/PE1_OS register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, these bits are always read as 0. When writing, always write 0.
15 to 14	PROT[1:0]	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set PROT[1:0] = 01_B when writing to CFECCCTL.
13 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Write a value to this bit simultaneously with setting PROT[1:0] = 01_B. 0: Enables correction of the 1-bit error detected. 1: Disables correction of the 1-bit error detected.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with setting PROT[1:0] = 01_B. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

52.11.3.2 CFERRINT_VCI2CFB/PE1_OS — Code flash error information control register

CFERRINT enables or disables generation of the error notification signal to the ECM upon detection of an ECC 2-bit error or ECC 1-bit error.

Access: This register can be read/written in 32-bit units.

Address: CFERRINT_VCI2CFB_OS: FFC6 2204_H
CFERRINT_PE1_OS: FFC6 2404_H

Value after reset: 0000 0006_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDIE	SEDIE
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 52.24 CFERRINT_VCI2CFB/PE1_OS register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	This bit should not be modified. When writing, always write initial value.
1	DEDIE	ECC 2-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

52.11.3.3 CFSTCLR_VCI2CFB/PE1_OS — Code flash status clear register

CFSTCLR clears the error flags in the error status register (CF1STERSTR), the overflow flag in the error overflow status register (CFOVFSTR), and the error address register (CF1STEADR).

Access: This register can be written only in 32-bit units.

Address: CFSTCLR_VCI2CFB_OS: FFC6 2208_H
CFSTCLR_PE1_OS: FFC6 2408_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ST CLR0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 52.25 CFSTCLR_VCI2CFB/PE1_OS register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	STCLR0	Error Overflow Flag Clear 0: No effect (Setting 0 does not affect the DEDF0 and SEDF0 flags in CF1STERSTR; ERROVF0 flag in CFOVFSTR; and CF1STEADR0.) 1: Writing 1 to this bit clears the DEDF0 and SEDF0 flags in CF1STERSTR; ERROVF0 flag in CFOVFSTR; and CF1STEADR0.

52.11.3.4 CFOVFSTR_VCI2CFB/PE1_OS — Code flash error count overflow status register

CFOVFSTR monitors occurrence of error overflow. If the second error occurs after the first error (= while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set.

Access: This register can be read in 32-bit units.

Address: CFOVFSTR_VCI2CFB_OS: FFC6 220C$_H$
CFOVFSTR_PE1_OS: FFC6 240C$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR OVF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 52.26 CFOVFSTR_VCI2CFB/PE1_OS register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, these bits are always read as 0. When writing, always write 0.
0	ERROVF0	Error Overflow Flag ERROVF0 is set if the second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register is set, except when both of the error address and source of the second error are the same as those of the first error. 0: Not occurred 1: Occurred Clearing condition: Set the STCLR0 bit in CFSTCLR to 1.

52.11.3.5 CF1STERSTR_VCI2CFB/PE1_OS — Code flash 1st error status register

CF1STERSTR monitors occurrence of the first error. The error status is set if an error occurs while the error flag is 0. The error flag is overwritten if an ECC 2-bit error occurs while the ECC 1-bit error monitor flag is set.

Access: This register can be read in 32-bit units.

Address: CF1STERSTR_VCI2CFB_OS: FFC6 2210_H
CF1STERSTR_PE1_OS: FFC6 2410_H

Value after reset: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF0	SEDF0
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 52.27 CF1STERSTR_VCI2CFB/PE1_OS register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, these bits are always read as 0. When writing, always write 0.
1	DEDF0	ECC 2-bit error Monitor Flag 0: Indicates that an ECC 2-bit error has not occurred. 1: Indicates that an ECC 2-bit error has occurred while the error flag DEDF0 is 0. Clearing condition: Set the STCLR0 bit in CFSTCLR to 1.
0	SEDF0	ECC 1-bit error Monitor Flag 0: Indicates that an ECC 1-bit error has not occurred. 1: Indicates that an ECC 1-bit error has occurred while the error flags DEDF0 and SEDF0 are 0. Clearing condition: Set the STCLR0 bit in CFSTCLR to 1.

52.11.3.6 CF1STEADR0_VCI2CFB/PE1_OS — Code flash 1st error address register

CF1STEADR0 holds the address at which an error has occurred.

The error address is set if an error occurs while all the error flags are 0 in CF1STERSTR. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.

Access: This register can be read in 32-bit units.

Address: CF1STEADR0_VCI2CFB_OS: FFC6 2250$_H$
CF1STEADR0_PE1_OS: FFC6 2450$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	EADR[24:16]								
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EADR[15:4]												—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 52.28 CF1STEADR0_VCI2CFB/PE1_OS register contents

Bit Position	Bit Name	Function
31 to 25	Reserved	When read, these bits are always read as 0. When writing, always write 0.
24 to 4	EADR[24:4]	1st Error Address Monitors the address of the first error. The error address is set if an error occurs while all the error flags are 0 in CF1STERSTR. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated. Clearing condition: Set the STCLR0 bit in CFSTCLR to 1.
3 to 0	Reserved	When read, these bits are always read as 0. When writing, always write 0.

52.11.3.7 CFSTSTCTL_VCI2CFB/PE1_OS — Code flash sub-test control register

The CFSTSTCTL register is used for the ECC test (self-diagnosis). This register is dedicated to code flash. After ECC test mode is enabled by setting ECCTST to 1, the ECC bits and address parity bit can be read directly.

Set PROT1 to 0 and PROT0 to 1 when writing to this register.

Access: This register can be read/written in 32-bit units.

Address: CFSTSTCTL_VCI2CFB_OS: FFC6 2350$_H$
CFSTSTCTL_PE1_OS: FFC6 2550$_H$

Value after reset: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT[1:0]		—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST
Value after reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 52.29 CFSTSTCTL_VCI2CFB/PE1_OS register contents

Bit Position	Bit Name	Function
31 to16	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
15 to 14	PROT[1:0]	Enables or disables modification of the ECCTST bit. The value written is not retained. These bits are always red as 0. Set PROT1 to 0 and PROT0 to 1 when writing to this register.
13 to 1	Reserved	When read, the value after reset is returned. When writing, write the value after reset.
o	ECCTST	ECC Test After ECC test mode is enabled by setting ECCTST to 1, the data in the ECC bits and the address parity bit can be read directly.

Correctly reading instructions from the code flash access port is not possible while ECC test mode is selected (ECCTST = 1). While the access port for the CPU is set to test mode (including during changes to the value of the ECCTST bit), the CPU must run a program from the local RAM or retention RAM and must not fetch instructions from the code flash memory.

The CPU has a small data buffer. If an old value remains in this buffer, the correct value cannot be read even when the ECCTST bit is switched. When switching the ECCTST bit, be sure to clear the data buffer. For how to clear the data buffer, see “RH850G3M User’s Manual Software Edition”.

From the code flash access port with ECC test mode selected, access must be made by reading 4 bytes aligned to 16n address. The results of reading code flash are as follows:

Table 52.30 Results of Reading Code Flash

Bit number	Description
31 to 10	These bits are always 0.
9	Address parity bit
8 to 0	ECC bits

52.12 Data Flash ECC

The ECC of the Data Flash generates an interrupt request listed in the following table:

Table 52.31 Data Flash ECC Interrupt Request

Function	Connected to
2 bit error detection interrupt	Error Control Module INTECCDEDEEP
1 bit error detection interrupt	Error Control Module INTECCSEDEEP

The 7-bit ECC value is calculated for each 32-bit transfer.

The ECC of the Data Flash and its registers are initialized by the following reset signal:

Table 52.32 Reset sources

ECC unit	Reset signal
Data Flash ECC	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

52.12.1 Overview

The data flash ECC is summarized in the table below.

Item	Description
ECC error detection and correction	ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected. • ECC error detection and correction are carried out (2-bit error detection and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, this function is enabled, and 1-bit error detection and correction and 2-bit error detection and notification are carried out.
Error notification	Upon occurrence of an ECC error, it is notified to the Error Control Module. **ECC Error:** • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, error notification is enabled upon detection of an ECC 2-bit error, and error notification is disabled upon detection of an ECC 1-bit error. The error notification signal is output, where an ECC 2-bit error is handled as one source, and an ECC 1-bit error is handled as one source.
Error status	A status register is provided, which indicates the statuses of ECC 2-bit error detection and ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.

52.12.2 List of Registers

Table 52.33 List of Registers

Module Name	Register Name	Symbol	Address
ECCEEP	Data Flash ECC control register	DFECCCTL	FFC6 2C00$_H$
ECCEEP	Data Flash error status register	DFERSTR	FFC6 2C04$_H$
ECCEEP	Data Flash error status clear register	DFERSTC	FFC6 2C08$_H$
ECCEEP	Data Flash error notification control register	DFERRINT	FFC6 2C14$_H$
ECCEEP	Data flash test control register	DFTSTCTL	FFC6 2C1C$_H$

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

52.12.3 Details of Registers

52.12.3.1 DFECCCTL — Data flash ECC control register

DFECCCTL enables or disables ECC error detection and 1-bit error correction for read access.

Set PROT[1:0] = 01_B when writing to the DFECCCTL register.

Access This register can be read/written in 32-bit units.

Address FFC6 2C00$_H$

Initial value 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 52.34 DFECCCTL register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the initial value is read. When written, write the initial value.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to DFECCCTLn.
14	PROT0	
13 to 2	Reserved	When read, the initial value is read. When written, write the initial value.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Write a value to this bit simultaneously with setting (PROT1, PROT0) = (0, 1). 0: Enables correction of the 1-bit error detected. 1: Disables correction of the 1-bit error detected.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with setting (PROT1, PROT0) = (0, 1). In the initial state, ECC error detection and correction are enabled. 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

52.12.3.2 DFERSTR — Data flash error status register

DFERSTR monitors occurrence of errors.

The SEDF bit is set if an ECC 1-bit error is detected while ECC error detection and correction are enabled, and the DEDF bit is set if an ECC 2-bit error is detected.

Access This register can be read only in 32-bit units.

Address FFC6 2C04$_H$

Initial value 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDF	SEDF
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 52.35 DFERSTR register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the initial value is read. When written, write the initial value.
1	DEDF	ECC 2-Bit Error Monitor Flag 0: An ECC 2-bit error is not generated. 1: An ECC 2-bit error is generated. Clearing condition: ERRCLR bit is set in Data Flash error status clear register. Setting condition: ECC 2-bit error is generated with both SEDF and DEDF being 0.
0	SEDF	ECC 1-bit error Monitor Flag 0: An ECC 1-bit error is not generated. 1: An ECC 1-bit error is generated. Clearing condition: ERRCLR bit is set in Data Flash error status clear register. Setting condition: ECC 1-bit error is generated with both SEDF and DEDF being 0.

52.12.3.3 DFERSTC — Data flash error status clear register

DFERSTC clears the error flags in the Data Flash error status register.

Access This register can be written only in 32-bit units.

Address FFC6 2C08$_H$

Initial value 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ERR CLR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 52.36 DFERSTC register contents

Bit Position	Bit Name	Function
31 to 1	Reserved	When read, the initial value is read. When written, write the initial value.
0	ERRCLR	SEDF/DEDF Flag Clear 0: No effect (Setting the ERRCLR bit to 0 does not affect the DEDF0 and SEDF flags in DFERSTR.) 1: The SEDF/DEDF flag in DFERSTR is cleared.

52.12.3.4 DFERRINT — Data flash error notification control register

DFERRINT enables or disables generation of the error notification signal upon detection of an ECC 2-bit error or an ECC 1-bit error.

Access This register can be read/written in 32-bit units.

Address FFC6 2C14$_H$

Initial value 0000 0002$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDIE	SEDIE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 52.37 DFERRINT register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	When read, the initial value is read. When written, write the initial value.
1	DEDIE	ECC 2-bit error Notification Control Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-bit error Notification Control Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

52.12.3.5 DFTSTCTL — Data flash test control register

DFTSTCTL is used for ECC testing.

The data of the ECC bit can be read after setting the ECC test mode (ECCTST = 1).

Set PROT[1:0] = 01_B when writing to the DFTSTCTL register.

Access This register can be read/written in 32-bit units.

Address FFC6 2C1C$_H$

Initial value 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC TST
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 52.38 DFTSTCTL register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	When read, the initial value is read. When written, write the initial value.
15	PROT1	Enables or disables modification of the ECCTST bit. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to DFECCCTL.
14	PROT0	
13 to 1	Reserved	When read, the initial value is read. When written, write the initial value.
0	ECCTST	ECC Test By setting ECC test mode bit to “1” (ECCTST=1), CPU can read ECC bit. Write a value to this bit simultaneously with setting (PROT1, PROT0) = (0, 1).

52.13 Usage Notes

1. Reading areas where programming or erasure was interrupted
   When programming or erasure of an area of flash memory is interrupted, the data stored in the area become undefined. To avoid undefined data that are read out becoming the source of faulty operation, take care not to fetch instructions or read data from areas where programming or erasure was interrupted.
2. Reading the code flash memory that has been erased but not yet been programming again
   Note that reading from an area of code flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state) can lead to the detection of an ECC error and generation of the corresponding exception. Use blank checking when you need to confirm that an area is in the non-programmed state.
3. Prohibition of additional writing
   Writing to a given area twice is not possible. If you want to overwrite data in an area of flash memory after writing to the area has been completed, erase the area first.
4. Resets during programming and erasure
   In the case of a reset due to the signal on the $\overline{\text{RESET}}$ pin during programming and erasure, wait for at least 20 µs once the operating voltage is within the range stipulated in the electrical characteristics after assertion of the reset signal before releasing the device from the reset state.
5. Allocation of vectors for interrupts and other exceptions during programming and erasure
   Generation of an interrupt or other exception during programming or erasure may lead to fetching of the vector from the code flash memory. If this does not satisfy the conditions for using background operation, set the address for vector fetching to an address that is not in the code flash memory.
6. Abnormal termination of programming and erasure (1)
   Even if programming/erasure ends abnormally due to reset input or power off, the programming/erasure state of the flash memory with undefined data cannot be verified or checked. Therefore, before using the area where programming/erasure has ended abnormally, erase the area again to prove that the corresponding area is completely erased.
7. Abnormal termination of programming and erasure (2)
   If programming and erasure of code flash memory are not completed normally, the lock bit for the target area may be enabled. In such cases, erase the block to erase the lock bit while the lock bit is in the disabled state.
8. Items prohibited during programming, erasure and blank check
   Do not perform the following operations during programming, erasure and blank check.
   - Have the operating voltage from the power supply go beyond the allowed range.
   - Change the frequency of the peripheral clock.

Section 53 RAM

The RH850/D1L/D1M incorporates following on-chip RAM:

Series Name	Product Name	Local RAM	Retention RAM	Video RAM
D1L1	R7F701401, R7F701421	256 Kbytes	16 Kbytes	–
D1L2	R7F701402, R7F701422	512 Kbytes		144 Kbytes
D1L2H	R7F701403, R7F701423			
D1M1_3.75MB	R7F701404			1.55 Mbytes
D1M1_5MB	R7F701405			
D1M1H_3.75MB	R7F701406			
D1M1H_5MB	R7F701407			
D1M1-V2	R7F701442, R7F701462			
D1M1A	R7F701441, R7F701461			2.4 Mbytes
D1M2_3.75MB	R7F701408, R7F701428			3.1 Mbytes
D1M2_5MB	R7F701410, R7F701430			
D1M2H_3.75MB	R7F701411, R7F701431			
D1M2H_5MB	R7F701412, R7F701432			

53.1 Local CPU RAM (LRAM)

The Local RAM is a RAM area that can be accessed immediately, without requiring the system to wait. Values are not retained in this area in DEEPSTOP mode.

53.1.1 Local RAM banks

The Local RAM consists of 4 banks of 32-bit wide RAM and an 7-bit wide ECC RAM.

- Bank 0: 32-bit RAM data at addresses xxxx $xxx0_H$
- Bank 1: 32-bit RAM data at addresses xxxx $xxx4_H$
- Bank 2: 32-bit RAM data at addresses xxxx $xxx8_H$
- Bank 3: 32-bit RAM data at addresses xxxx $xxxC_H$

Each access to the Local RAM writes or reads four 32-bit words, i.e. is performed via the 128-bit RAM bus.

The 7-bit ECC value is calculated for each 32-bit transfer.

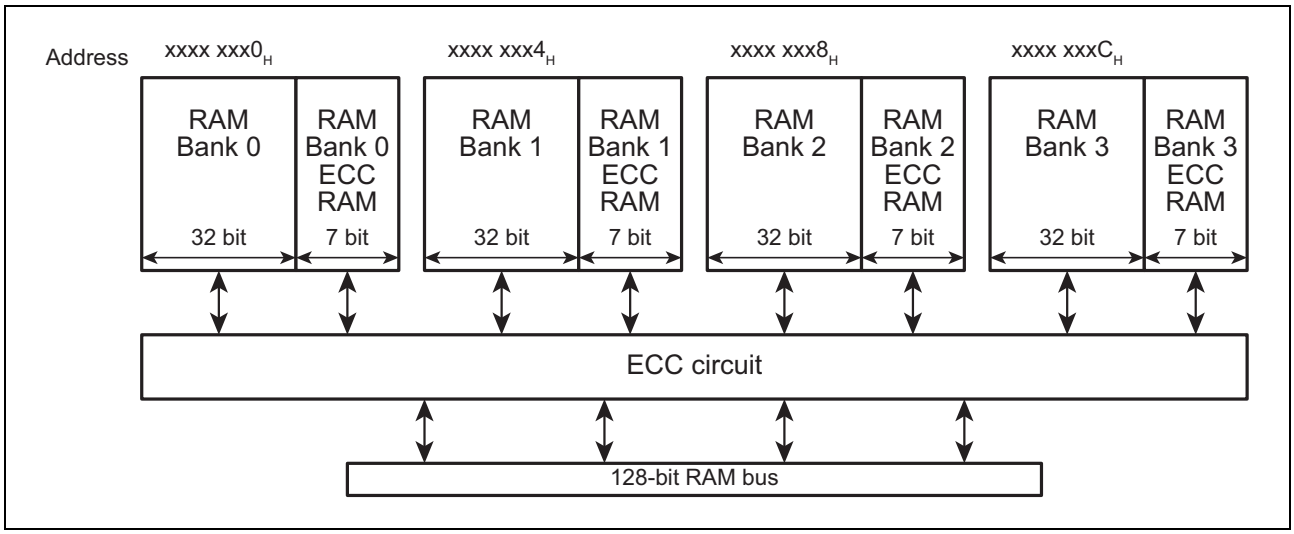


Figure 53.1 Local RAM block diagram

53.1.2 Local RAM initialization

Before reading from the Local RAM, while the ECC error detection and correction is enabled, the entire Local RAM must be initialized.

CAUTIONS

1. **RAM initialization must be performed by 32-bit write accesses in order to ensure that all RAM data is initialized and the respective ECC value fits to the RAM content.**
2. **Reading from un-initialized Local RAM may lead to the detection of ECC errors.**

53.1.3 ECC of Local RAM

The ECC of the Local RAM generates an interrupt request listed in the following table:

Table 53.1 Interrupt Request by ECC of Local RAM

Function	Connected to
ECC 1-bit error interrupt of Local RAM	Error Control Module INTECCSLRAM
ECC 2-bit error interrupt of Local RAM	Error Control Module INTECCDLRAM

The ECC of the Local RAM and its registers are initialized by the following reset signal:

Table 53.2 Reset sources

ECCn unit	Reset signal
Local RAM ECC	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

The following table gives an outline of the ECC functions for the Local RAM.

Table 53.3 Local RAM ECC functions

Item	Description
ECC error detection and correction	ECC error detection and correction can be either enabled or disabled. When enabled, either of the following settings can be selected. • ECC error detection and correction are carried out (2-bit error detection and 1-bit error detection and correction are carried out). • ECC error detection is carried out (2-bit error detection and 1-bit error detection are carried out). When disabled, neither error detection nor correction is carried out. In the initial state, the ECC function is enabled; 1-bit error detection and correction and 2-bit error detection and notification are carried out.
Error notification	Upon occurrence of an ECC error, it is notified to the Error Control Module. • Error notification can be either enabled or disabled upon detection of an ECC 2-bit error. • Error notification can be either enabled or disabled upon detection of an ECC 1-bit error. In the initial state, notification of the 2-bit error is enabled and notification of the 1-bit error is disabled.
Error status	A status register is provided, which indicates the statuses of ECC 2-bit error detection and ECC 1-bit error detection. If an error occurs while no error status is set, the corresponding status is set. The error status can be cleared using the clear register.
Address capture	If an ECC error occurs while no error status is set, the address at which the associated error has occurred is captured. The address is captured when an ECC 2-bit error or an ECC 1-bit error is detected. The error status serves as the enable bit of the capture address.

53.1.4 Registers

This section describes all registers relating to the Local RAM ECC.

Table 53.4 List of LRAM ECC Registers

Module Name	Address	Symbol	Register Name	R/W	Initial Value	Access Size
ECCCPU1	FFC65400	LRECCCTL	Local RAM ECC control register	R/W	0000 0000$_H$	16/32
ECCCPU1	FFC65404	LRERRINT	Local RAM error information control register	R/W	0000 0006$_H$	8/16/32
ECCCPU1	FFC65408	LRSTCLR	Local RAM status clear register	W	0000 0000$_H$	8/16/32
ECCCPU1	FFC6540C	LROVFSTR	Local RAM error count overflow status register	R	0000 0000$_H$	8/16/32
ECCCPU1	FFC65410	LR1STERSTR	Local RAM 1st error status register	R	0000 0000$_H$	8/16/32
ECCCPU1	FFC65450	LR1STEADR0	Local RAM 1st error address register 0	R	0000 0000$_H$	8/16/32
ECCCPU1	FFC65454	LR1STEADR1	Local RAM 1st error address register 1	R	0000 0000$_H$	8/16/32
ECCCPU1	FFC65458	LR1STEADR2	Local RAM 1st error address register 2	R	0000 0000$_H$	8/16/32
ECCCPU1	FFC6545C	LR1STEADR3	Local RAM 1st error address register 3	R	0000 0000$_H$	8/16/32
ECCCPU1	FFC65004	LRTSTCTL	Local RAM test control register	R/W	0000 0000$_H$	16/32
ECCCPU1	FFC65008	LRTDATBF0	Local RAM test data read buffer 0	R	0000 0000$_H$	32
ECCCPU1	FFC6500C	LRTDATBF1	Local RAM test data read buffer 1	R	0000 0000$_H$	32

NOTE

In the header files the names of the above registers are defined in the following format:

<ModuleName> + <Symbol>.

<ModuleName> and <Symbol> are defined in the above table.

53.1.4.1 LRECCCTL — Local RAM ECC control register

LRECCCTL enables or disables ECC error detection and correction and 1-bit error correction. Set the PROT1 and PROT0 bits to 01_B when writing to LRECCCTL.

Access: This register can be accessed in 16-bit and 32-bit units.

Address: FFC6 5400_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT1	PROT0	—	—	—	—	—	—	—	—	—	—	—	—	SECDIS	ECCDIS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 53.5 LRECCCTL register contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved. These bits are always read as 0. The write value should also be 0.
15	PROT1	Enables or disables modification of the ECCDIS and SECDIS bits. The value written is not retained. These bits are always read as 0. Set (PROT1, PROT0) = (0, 1) when writing to LRECCCTL.
14	PROT0	
13 to 2	—	Reserved. These bits are always read as 0. The write value should also be 0.
1	SECDIS	1-Bit Error Correction Disable Enables or disables 1-bit error correction when ECC error detection and correction are enabled. Write a value to this bit simultaneously with setting (PROT1, PROT0) = (0, 1). 0: Enables correction of the 1-bit error detected. 1: Disables correction of the 1-bit error detected.
0	ECCDIS	ECC Disable Enables or disables ECC error detection and correction. Write a value to this bit simultaneously with setting (PROT1, PROT0) = (0, 1). 0: Enables ECC error detection and correction. 1: Disables ECC error detection and correction.

53.1.4.2 LRERRINT — Local RAM error information control

LRERRINT enables or disables generation of the error notification signal to the ECM upon detection of an ECC 2-bit error or an ECC 1-bit error.

Access: This register can be accessed in 8-bit, 16-bit and 32-bit units.

Address: FFC6 5404$_H$

Initial value: 0000 0006$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEDIE	SEDIE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 53.6 LRERRINT register contents

Bit Position	Bit Name	Function
31 to 3	—	Reserved. These bits are always read as 0 or 1, respectively. The write value should also be the initial value.
2	—	Reserved. This bit is always read as 1. The write value should also be 1.
1	DEDIE	ECC 2-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 2-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 2-bit error. 1: Enables notification of the ECC 2-bit error.
0	SEDIE	ECC 1-bit error Notification Enable Enables or disables generation of the error notification signal upon detection of a 1-bit error when ECC error detection and correction are enabled. 0: Disables notification of the ECC 1-bit error. 1: Enables notification of the ECC 1-bit error.

53.1.4.3 LRSTCLR — Local RAM status clear register

LRSTCLR clears the error flags in the error status register (LR1STERSTR), the overflow flag in the error overflow status register (LROVFSTR), and the error address register (LR1STEADR).

Access: This register can be accessed in 8-bit, 16-bit and 32-bit units.

Address: FFC6 5408$_H$

Initial value: Reading this register returns always 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	STCLR3	STCLR2	STCLR1	STCLR0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	W	W	W	W

Table 53.7 LRSTCLR register contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved. These bits are always read as 0. The write value should also be 0.
3	STCLR3	Error Status / Error Overflow Flag Clear (for bank 3) Writing 1 to this bit clears the DEDF3 and SEDF3 flags in LR1STERSTR; ERROVF3 flag in LROVFSTR; and LR1STEADR3.
2	STCLR2	Error Status / Error Overflow Flag Clear (for bank 2) Writing 1 to this bit clears the DEDF2 and SEDF2 flags in LR1STERSTR; ERROVF2 flag in LROVFSTR; and LR1STEADR2.
1	STCLR1	Error Status / Error Overflow Flag Clear (for bank 1) Writing 1 to this bit clears the DEDF1 and SEDF1 flags in LR1STERSTR; ERROVF1 flag in LROVFSTR; and LR1STEADR1.
0	STCLR0	Error Status / Error Overflow Flag Clear (for bank 0) Writing 1 to this bit clears the DEDF0 and SEDF0 flags in LR1STERSTR; ERROVF0 flag in LROVFSTR; and LR1STEADR0.

53.1.4.4 LROVFSTR — Local RAM error count overflow status register

LROVFSTR monitors occurrence of error overflow. If the second error occurs after the first error (= while any of the error flags in the error status register is set), the flag in this register is set. However, if the second error is identical to the first error (both the source and address are same), this flag is not set.

An ERROVF is cleared by setting the STCLR bit to 1 in LRSTCLR.

Access: This register can be accessed in 8-bit, 16-bit and 32-bit units.

Address: FFC6 540C$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	ERROVF3	ERROVF2	ERROVF1	ERROVF0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 53.8 LROVFSTR register contents

Bit Position	Bit Name	Function
31 to 4	—	Reserved. These bits are always read as 0. The write value should also be 0.
3	ERROVF3	Error Overflow Flag (for bank 3) ERROVF3 is set if the second error occurs while any of the error flags (DEDF3 and SEDF3) in the error status register is set, except when both of the error address and source of the second error are the same as those of the first error.
2	ERROVF2	Error Overflow Flag (for bank 2) ERROVF2 is set if the second error occurs while any of the error flags (DEDF2 and SEDF2) in the error status register is set, except when both of the error address and source of the second error are the same as those of the first error.
1	ERROVF1	Error Overflow Flag (for bank 1) ERROVF1 is set if the second error occurs while any of the error flags (DEDF1 and SEDF1) in the error status register is set, except when both of the error address and source of the second error are the same as those of the first error.
0	ERROVF0	Error Overflow Flag (for bank 0) ERROVF0 is set if the second error occurs while any of the error flags (DEDF0 and SEDF0) in the error status register is set, except when both of the error address and source of the second error are the same as those of the first error.

53.1.4.5 LR1STERSTR — Local RAM 1st error status register

LR1STERSTR monitors occurrence of the first error. The error status is set if an error occurs while the error flag is 0. The error flag is overwritten if an ECC 2-bit error occurs while the ECC 1-bit error flag is set.

An error status bit is cleared by setting 1 to the STCLR bit in LRSTCLR.

Access: This register can be accessed in 8-bit, 16-bit and 32-bit units.

Address: FFC6 5410$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	DEDF3	SEDF3	—	—	—	—	—	—	DEDF2	SEDF2
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DEDF1	SEDF1	—	—	—	—	—	—	DEDF0	SEDF0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 53.9 LR1STERSTR register contents

Bit Position	Bit Name	Function
7+8n:4+8n	—	Reserved. These bits are always read as 0. The write value should also be 0.
1+8n	DEDFn	ECC 2-bit error Monitor Flag 0: Cleared to 0 by setting 1 to the STCLRn bit in LRSTCLR. 1: Indicates that an ECC 2-bit error has occurred while the error flags DEDFn are 0.
0+8n	SEDFn	ECC 1-bit error Monitor Flag 0: Cleared to 0 by setting 1 to the STCLRn bit in LRSTCLR. 1: Indicates that an ECC 1-bit error has occurred while all the error flags DEDFn and SEDFn are 0.

Note: n = 0 to 3

53.1.4.6 LR1STEADRn — Local RAM 1st error address register n (n = 0 to 3)

LR1STEADRn holds the address at which an error has occurred.

The error address is set if an error occurs while all the error flags are 0 in LR1STERSTR. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set in LR1STERSTR. Once an ECC 2-bit error occurs, the address is not updated.

This register stores the actual address [18:4]. Align the bit position with the base address (FED8 0000_H for all devices) and add n * 4 when transform the internal address to the actual address.

LR1STEADRn is also cleared by setting the STCLR bit to 1 in LRSTCLR.

Access: This register can be accessed in 8-bit, 16-bit and 32-bit units.

Address: FFC6 5450_H + n x 4_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	EADR[14:0]														
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 53.10 LR1STEADRn register contents

Bit Position	Bit Name	Function
31 to 15	—	Reserved. These bits are always read as 0. The write value should also be 0.
14 to 0	EADR[14:0]	1st Error Address (for bank n) Monitors the address of the first error. The error address is set if an error occurs while all the error flags for bank n are 0 in LR1STERSTR. The address is updated if an ECC 2-bit error occurs while the ECC 1-bit error flag is set as the first error. Once an ECC 2-bit error occurs, the address is not updated.

53.1.4.7 LRTSTCTL — Local RAM Test Control Register

This register is used for the ECC test (self-diagnosis).
After ECC test mode is enabled by setting ECCTST = 1, desired data can be written to the ECC bits.
The DATSEL bit is used to select RAM data or the ECC bits.

Access: This register can be accessed in 16-bit and 32-bit units.

Address: FFC6 5004_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PROT[1:0]		—	—	—	—	—	—	—	—	—	—	—	—	ECC TST	DATSEL
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 53.11 LRTSTCTL register contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved. These bits are always read as 0. The write value should also be 0.
15, 14	PROT[1:0]	These bits are used to enable or disable rewriting ECCTST, and DATSEL bits. The value written is not retained. These bits are always read as 0. Set PROT[1:0] = (0,1) when writing to LRTSTCTL.
13 to 2	—	Reserved. These bits are always read as 0. The write value should also be 0.
1	ECCTST	ECC Test When ECC test mode is set (ECCTST = 1), reading through LRTDATBFn is enabled, and direct writing is enabled by setting DATSEL = 1.
0	DATSEL	This bit is valid when ECCTST = 1. This bit selects the RAM bit which can be accessed when writing. When this bit is set to 1, bits [6:0] in the 32-bit write data are written to the ECC bits. 0: RAM data is selected. 1: The ECC bits are selected.

CAUTION

When ECC test mode for the local RAM is enabled (ECCTST = 1), the local RAM should be accessed in 4-byte units.

53.1.4.8 LRTDATBFn — Local RAM Test Data Read Buffer n (n = 0, 1)

In ECC test mode (self-diagnosis), the ECC bits can be read. If the local RAM is read while ECCTST = 1 in the local RAM test control register LRTSTCTL, reading from the local RAM reads out the ECC bits, and these bits are stored in this buffer.

Access: This register can be accessed in 32-bit units.

Address: LRTDATBF0: FFC6 5008_H
LRTDATBF1: FFC6 $500C_H$

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	LRDATBF[24:16]								
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	LRDATBF[8:0]								
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 53.12 LRTDATBFn register contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved. These bits are always read as 0. The write value should also be 0.
24 to 16	LRDATBF (2n + 1) [24:16]	These bits are valid when ECCTST = 1 (selecting test mode) in the local RAM test control register. When reading from the corresponding bank in the local RAM, the ECC bits for the local RAM (bank (2n + 1)) are stored in LRDATBF(2n + 1)[22:16] and LRDATBF(2n + 1)[24:23].
31 to 25	—	Reserved. These bits are always read as 0. The write value should also be 0.
15 to 9	LRDATBF (2n) [8:0]	These bits are valid when ECCTST = 1 (selecting test mode) in the local RAM test control register. When reading from the corresponding bank in the local RAM, the ECC bits for the local RAM (bank (2n)) are stored in LRDATBF(2n)[6:0] and LRDATBF (2n)[8:7].

NOTE

When n is 1, the bit name of bank 2 becomes LRDATBF2[8:0].

53.1.5 Usage Notes

On path between Local RAM and CPU, buffers are implemented to realize fast Local RAM access.

Therefore, when a load instruction is executed for the same address after a store instruction to Local RAM, the load instruction may read out data from buffers instead of data on Local RAM. If stored data on Local RAM is needed, execute either of following procedures.

1. Read out data in the order written address after writing more than 32 byte data into Local RAM.
2. Execute SYNCM instruction before a load instruction is executed for the same address after a store instruction to Local RAM.

NOTE

DMAC or other bus masters also read data from buffers, when data have not been stored to Local RAM yet.

53.2 Retention RAM (RRAM)

The retention RAM is a RAM area in which values are retained in this area in DEEPSTOP mode.

53.2.1 Retention RAM Error Correction Coding (RRAMECC)

53.2.2 Units

This microcontroller has the following number of units of the Retention RAM ECC module.

Table 53.13 Units

Retention RAM ECC	D1L1	D1L2(H)	D1M1(H) D1M1-V2 D1M1A	D1M2(H)
Units	1	1	1	1
Names	RRAMECC0	RRAMECC0	RRAMECC0	RRAMECC0

Unit index n

Throughout this section, the individual units of the Retention RAM is identified by the index “n” (n = 0), for example RRAMECCnCTL for the BRMn control register.

Error detection stage index m

Throughout this section, the individual number of an error detection stage is identified by the index “m” (m = 7), for example the register RRAMECCnEADm holds the address of the m^{th} error detection.

53.2.2.1 Register base addresses <RRAMECCn_base>

All Retention RAM ECC module register addresses are given as address offsets from the individual base address <RRAMECCn_base>.
The <RRAMECCn_base> address of the retention RAM ECC module is listed in the following table:

Table 53.14 Register base addresses <RRAMECC_base>

RRAMECCn unit	<RRAMECCn_base> address
RRAMECC0	FFC7 9000_H

53.2.2.2 Retention RAM ECC interrupts

The ECC of the Retention RAM generates interrupt requests listed in the following table:

Table 53.15 Interrupt Request by ECC of Retention RAM

Function	Connected to
ECC 1-bit error interrupt of Retention RAM	Error Control Module INTECCSEDRRAM
ECC 2-bit error interrupt of Retention RAM	Error Control Module INTECCDEDRRAM

53.2.2.3 Retention RAM initialization

Before reading from the Retention RAM, while the ECC error detection and correction is enabled, the entire Retention RAM must be initialized.

CAUTIONS

1. **RRAM initialization must be performed by 64-bit write accesses in order to ensure that all RRAM data is initialized and the respective ECC value fits to the RRAM content.**
2. **Reading from un-initialized Retention RAM may lead to the detection of ECC errors.**

53.2.2.4 Retention RAM ECC reset sources

The ECC of the Retention RAM and its registers are initialized by the following reset signal:

Table 53.16 Reset sources

RRAMECCn unit	Reset signal
RRAMECC0	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

53.2.2.5 Retention RAM ECC functions

The Retention RAM Error Correction Coding (RRAMECC) module generates and tests Error Correction Coding (ECC) information for Retention RAM data.

8 bit ECC information is calculated and added to each 64-bit data unit, that is written to the Retention RAM.

During read-back of the data the ECC information is checked against the read data.

Upon detection of a single erroneous bit in a 64-bit data unit

- the erroneous bit of the read data is corrected (but the erroneous memory data is retained)
- a Single Error Detection (SED) flag RRAMECCnCTL.ECSEDFm is set
- the read access address is stored in the RRAMECCnEADm register
- an INTECCSEDRRAM interrupt is generated

Upon detection of two erroneous bits in a 64-bit data unit

- a Double Error Detection (DED) flag RRAMECCnCTL.ECDEDFm is set
- the read access address is stored in the RRAMECCnEADm register
- an INTECCDEDRRAM interrupt is generated

ECC process during data write with less than 64 bit

If data with less than 64 bit is written (e.g. 32 bit, 16 bit and 8 bit) through the direct 32-bit access with ECC area, the respective 64-bit data unit is read and the ECC is checked.

If this ECC check detects a single erroneous bit in the read 64-bit data unit

- the erroneous bit is corrected in the RAM
- the Single Error Detection (SED) flag RRAMECCnCTL.ECSEDFm is set

- the read access address is stored in a RRAMECCnEADm register
- the INTECCSEDRRAM interrupt is generated
- a new (and now correct) ECC value with the correct data is calculated.

If this ECC check detects two erroneous bits in the read 64-bit data unit

- the Double Error Detection (DED) flag RRAMECCnCTL.ECDEDFm is set
- the read access address is stored in a RRAMECCnEADm register
- the INTECCDEDRRAM interrupt is generated

A one bit error in the target 64-bit data unit is corrected in the RAM during write.

A two bit error in the target 64-bit data unit is not corrected during the write and a new ECC value will be calculated with the uncorrected data.

CAUTION

Under the above conditions an ECC error interrupt can occur during a write access.

53.2.2.6 Error detection

(1) Error detection stage

Information about up to eight error detections are stored, i.e.

- up to eight read addresses with errors are stored in the error address registers RRAMECCnEADm, with m = 0 to 7
- single or double error detection indication is stored in the ECSEDFm and ECDEDFm of the ECC control register RRAMECCnCTL.

The error detection flags ECSEDFm and ECDEDFm are related to the error address stored in RRAMECCnEADm. I.e. each error detection address has separate assigned error flags.

The address of the detected errors are stored in ascending order of m, starting from RRAMECCnEAD0.

RRAMECCnEADm is set to its default value $0000\ 0000_H$ by clearing the error detection flag ECER1F or ECER2F in the ECC control register RRAMECCnCTL.

1. If only single bit errors are present ECER1F clears all single bit errors.
   ECER2F does not clear single bit errors in this case.
2. If only double bit errors are present ECER2F clears all double bit errors.
   ECER1F does not clear double bit errors in this case.
3. If the detected errors are a mixture of single and double bit errors, ECER2F or ECER1F clears all of them. It is not possible to clear only one or the other.
4. In each case all RRAMECCnEADm registers are cleared by ECER1F if a single bit error was detected.
5. In each case all RRAMECCnEADm registers are cleared by ECER2F if a double bit error was detected.

(2) Error detection stages overflow

An error detection while all error detection stages hold valid error indications (i.e. any of the error flags ECSEDFm or ECDEDFm of stage m is active), are treated as follows:

- If the current error detection reveals a single bit error, all error flags ECSEDFm or ECDEDFm and error addresses RRAMECCnEADm are not changed.
- If the current error detection reveals a double bit error and
  - all error detection stages m = 0 to 7 indicate single bit errors (i.e. ECSEDF[7:0] = 1 and ECDEDF[7:0] = 0),
    - the current error address overwrites the previous error address in RRAMECCnEAD7
    - the double bit error flag is set, i.e. ECDEDF7 = 1. The double bit error interrupt INTECCDEDRRAM is asserted, provide this is permitted by RRAMECCnCTL.EC2EDIC = 1.
  - if any of the error detection stages m = 0 to 7 indicate a double bit errors (i.e. any ECDEDF[7:0] = 1) all error flags ECSEDFm or ECDEDFm and error addresses RRAMECCnEADm are not changed.

In any of the above cases the overflow flag RRAMECCnCTL.ECOVFF is set.

(3) Additional error flags

- RRAMECCnCTL.ECER2F: Double bit error detection flag
  This flag is set with the 1st occurrence of a double bit error. It is only cleared by setting RRAMECCnCTL.ECER2C = 1.
- RRAMECCnCTL.ECER1F: Single bit error detection flag
  This flag is set with the 1st occurrence of a single bit error. It is only cleared by setting RRAMECCnCTL.ECER1C = 1.
- RRAMECCnCTL.ECEMF: ECC error has occurred during the last read
  This bit is cleared by the next read if the ECC value is correct.

(4) Error interrupt control

The generation of error detection interrupts can be controlled by control bits of the RRAMECCnCTL register:

- EC2EDIC = 0/1 disables/enables generation of the double error detection interrupt INTECCDEDRRAM
- EC1EDIC = 0/1 disables/enables generation of the single error detection interrupt INTECCSEDRRAM

53.2.2.7 ECC mode control

The ECC module can be set in two different modes:

- RRAMECCnCTL.ECTHM = 0: normal operation mode
  Error judgement can be enabled/disabled by RRAMECCnCTL.ECERVF
- RRAMECCnCTL.ECTHM = 1: pass through mode
  ECC calculation and storage of the ECC bits in the RAM is performed even in pass through mode. ECC decoding upon RAM data read is not executed.

The following table gives an overview about the different modes:

Generation of the 8-bit ECC data during write of data to the Retention RAM is not affected by RRAMECCnCTL.ECTHM.

Table 53.17 ECC modes overview

ECC mode setting		Error correction	Error address storage	Interrupts assertion	Error flags settings
ECTHM*1	ECERVF*1	Single bit errors	RRAMECCnEADm *2, ECDEDFm*1, ECSEDFm*1	INTECCDED RRAM, INTECCSED RRAM	ECER2F, ECER1F, ECEMF*1
0: normal operation	0: error judgement disabled	yes	no	no	no
	1: error judgement enabled	yes	yes	yes	yes
1: pass through	x	no	no	no	no

Note 1. Bits in the ECC control register RRAMECCnCTL.
Note 2. ECC error address registers.

ECC mode setting protection

Modification of the ECC mode control bits ECTHM and ECERVF in the RRAMECCnCTL register is depends on two protection bits in the RRAMECCnCTL register:

- ECMA[1:0] = 01_B: modification of ECTHM and ECERVF is possible
- All other ECMA[1:0] settings: modification of ECTHM and ECERVF is not possible

CAUTION

Since an 8-bit ECC code is first calculated during the write to the respective 64-bit data, the ECC code does most likely not fit to the data unit after power-up of the Retention RAM.

Consequently a read of un-initialized Retention RAM content will most likely lead to ECC error detections.

53.2.3 Registers

This section contains a description of all registers of the Retention RAM ECC.

Table 53.18 Retention RAM ECC registers overview

Register Name	Shortcut	Address
ECC control register	RRAMECCnCTL	<RRAMECCn_base> + 00_H
ECC error address register 0	RRAMECCnEAD0	<RRAMECCn_base> + 14_H
ECC error address register 1	RRAMECCnEAD1	<RRAMECCn_base> + 18_H
ECC error address register 2	RRAMECCnEAD2	<RRAMECCn_base> + $1C_H$
ECC error address register 3	RRAMECCnEAD3	<RRAMECCn_base> + 20_H
ECC error address register 4	RRAMECCnEAD4	<RRAMECCn_base> + 24_H
ECC error address register 5	RRAMECCnEAD5	<RRAMECCn_base> + 28_H
ECC error address register 6	RRAMECCnEAD6	<RRAMECCn_base> + $2C_H$
ECC error address register 7	RRAMECCnEAD7	<RRAMECCn_base> + 30_H
ECC test mode control register	RRAMECCnTMC	<RRAMECCn_base> + 04_H
ECC encode/decode input/output substitution register 1	RRAMECCnTED1	<RRAMECCn_base> + 08_H
ECC encode/decode input/output substitution register 2	RRAMECCnTED2	<RRAMECCn_base> + $0C_H$
ECC test redundant bit data control register	RRAMECCnTRC	<RRAMECCn_base> + 10_H

<RRAMECCn_base>

The base addresses <RRAMECCn_base> of the Retention RAM is defined in “Register base addresses <RRAMECCn_base>” in the section above.

NOTE

Memory access right after register access has possibility to cause the racing between two accesses via different routes.
Therefore, the last accessed register value must be confirmed at least before memory access. Refer to Section 3.4.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation for details.

53.2.3.1 RRAMECCnCTL – ECC control register

This register holds various status and control bits.

Access: This register can be accessed in 32-bit units.

Address: <RRAMECCn_base> + 00_H

Initial value: 0000 0000 0001 $000x_B$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECDEDF7	ECSEDF7	ECDEDF6	ECSEDF6	ECDEDF5	ECSEDF5	ECDEDF4	ECSEDF4	ECDEDF3	ECSEDF3	ECDEDF2	ECSEDF2	ECDEDF1	ECSEDF1	ECDEDF0	ECSEDF0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA[1:0]		0	0	ECOVF	ECER2C	ECER1C	0	ECTHM	ECERVF	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	ECEMF
Initial value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	x
R/W	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 53.19 RRAMECCnCTL register contents (1/3)

Bit position	Bit name	Function
31 to 16	ECDEDFm ECSEDFm	These bits indicate the type of error, that is related to the stored error address of stage m in the RRAMECCnEADm register. • ECDEDFm = 0 and ECSEDFm = 0 No error address stored in RRAMECCnEADm. • ECDEDFm = 1 and ECSEDFm = 0 RRAMECCnEADm address of double error detection • ECDEDFm = 0 and ECSEDFm = 1 RRAMECCnEADm address of single error detection • ECDEDFm = 1 and ECSEDFm = 1 Does not occur. These bits are cleared under following conditions: • While ECER2F bit is set, writing RRAMECCnCTL.ECER2C = 1 clears ECDEDF7-0 and ECSEDF7-0 While ECER1F bit is set, writing RRAMECCnCTL.ECER1C = 1 clears ECDEDF7-0 and ECSEDF7-0 • ECC pass through mode (RRAMECCnCTL.ECTHM = 1) • ECC error judgment disabled (RRAMECCnCTL.ECERVF = 0)
15 to 14	EMCA[1:0]	ECC mode protection • EMCA[1:0] = 01_B: modification of the ECC mode control bits ECTHM and ECERVF is permitted • All other EMCA[1:0] setting: modification of the ECC mode control bits ECTHM and ECERVF is not possible
13 to 12	Reserved	When read, the value after reset is read. When written, write the value after reset.

Table 53.19 RRAMECCnCTL register contents (2/3)

Bit position	Bit name	Function
11	ECOVFF	Overflow detection flag In case of an error detection, while all error address registers RRAMECCnEADm hold already a valid error address, this overflow flag is set. 0: no overflow detected 1: overflow detected Writing to this bit has no effect. These bits are cleared under following conditions: • Writing RRAMECCnCTL.ECER2C = 1 while ECER2F bit is set • Writing RRAMECCnCTL.ECER1C = 1 while ECER1F bit is set • ECC pass through mode (RRAMECCnCTL.ECTHM = 1) • ECC error judgment disabled (RRAMECCnCTL.ECERVF = 0)
10	ECER2C	Clear double bit error detection flag ECER2F 0: no function 1: clears ECER2F Reading this bit returns always 0.
9	ECER1C	Clear single bit error detection flag ECER1F 0: no function 1: clears ECER1F Reading this bit returns always 0.
8	Reserved	When read, the value after reset is read. When written, write the value after reset.
7	ECTHM	Pass through mode enable bit. 0: Normal operation mode 1: Pass through mode enable. ECC decoding upon RAM data read is not executed, i.e. no error judgment or the bit correction is performed. ECC calculation and storage of the ECC bits in the RAM is performed even in pass through mode. Modification of this bit is only possible, if EMCA[1:0] = 01_B. Otherwise any write to this bit is ignored.
6	ECERVF	ECC error judgment enable 0: error judgement disabled 1: error judgement enabled Error judgement is only effective in normal operation mode, i.e. if RRAMECCnCTL.ECTHM = 0. If error judgement is disabled no interrupts are asserted in case of a single or double bit error detection. However detected single bit errors are corrected. Modification of this bit is only possible, if EMCA[1:0] = 01_B. Otherwise any write to this bit is ignored.
5	EC1ECP	Single bit error correction permission 0: Single bit error is corrected 1: Single bit error is not corrected
4	EC2EDIC	Double bit error detection interrupt INTRRAMECCnDED control 0: INTRRAMECCnDED is not generated 1: INTRRAMECCnDED is generated upon double bit error detection
3	EC1EDIC	Single bit error detection interrupt INTRRAMECCnSED control 0: INTRRAMECCnSED is not generated 1: INTRRAMECCnSED is generated upon single bit error detection
2	ECER2F	Double bit error detection flag 0: no double bit error detected 1: double bit error detected Writing to this bit has no affect. This bit is cleared under following conditions: • writing RRAMECCnCTL.ECER2C = 1 • ECC pass through mode (RRAMECCnCTL.ECTHM = 1) • ECC error judgment disabled (RRAMECCnCTL.ECERVF = 0)

Table 53.19 RRAMECCnCTL register contents (3/3)

Bit position	Bit name	Function
1	ECER1F	Single bit error detection flag 0: no single bit error detected 1: single bit error detected and corrected Writing to this bit has no affect. This bit is cleared under following conditions: • writing RRAMECCnCTL.ECER1C = 1 • ECC pass through mode (RRAMECCnCTL.ECTHM = 1) • ECC error judgment disabled (RRAMECCnCTL.ECERVF = 0)
0	ECEMF	ECC error indication of current Retention RAM data read 0: no ECC error detected at current read 1: ECC error detected at current read This bit is updated by every RAM output data. This bit is cleared under following conditions: • current Retention RAM data read without ECC error • ECC pass through mode (RRAMECCnCTL.ECTHM = 1) • ECC error judgment disabled (RRAMECCnCTL.ECERVF = 0)

53.2.3.2 RRAMECCnEADm – ECC error address register m (m = 0 to 7)

These registers store the read address, when an ECC error was detected.

Access: This register can be accessed in 32-bit units.

Address: <RRAMECCn_base> + 14_H + m x 04_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEAD[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEAD[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 53.20 RRAMECCnEADm register contents

Bit position	Bit name	Function
31 to 0	ECEAD[31:0]	Address of read access, when an ECC error was detected. The type of error, i.e. single or double bit error, is indicated by RRAMECCnCTL.ECDEDm and RRAMECCnCTL.ECSEDm. ECEAD[31:0] is the address of the erroneous 64-bit data unit. Base address is not included. The complete read address is calculated by use of this formula: read address = 3FCE 4000_H + ECEAD[31:0] * 8 Under certain conditions the address of the RRAMECCnEAD7 register is overwritten upon detection of an ECC error. Refer to Section (2), Error detection stages overflow for details. RRAMECCnEADm is set to its default value by clearing the error detection flag ECER1C or ECER2C in the ECC control register RRAMECCnCTL.

53.2.3.3 RRAMECCnTMC – ECC test mode control register

The RRAMECCnTMC register switches to and controls the test mode.

This register can be used when RRAM is not accessed.

Access: This register can be accessed in 16-bit units.

Address: <RRAMECCn_base> + 04_H

Initial value: $0000\ 0000_H$

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA[1:0]		0	0	0	0	0	0	EC TMCE	0	0	EC TRRS	EC REOS	EC ENS	EC DCS	EC REIS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 53.21 RRAMECCnTMC register contents (1/2)

Bit position	Bit name	Function
15, 14	ETMA[1:0]	Access control bits 1 and 0 to ECC test mode bit These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 10_B, writing to bit 7 is enabled.
7	ECTMCE	ECC test mode enable bit This bit specifies whether to enable access to the test registers and test control bits. When writing to this bit, 10_B should be written to ETMA[1:0] at the same time. 0: Access to the test mode registers and bits is disabled. 1: Access to the test mode registers and bits is enabled. Test registers: RRAMECCnTED, RRAMECCnTRC, RRAMECCnHORD, RRAMECCnECRD, RRAMECCnERDB Register test control bit: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
4	ECTRRS	ECC RAM read test mode selection bit Select reading destination when reading RRAMECCnTED register and reading destination when reading RRAMECCnERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: The read value of the RRAMECCnTED register is the write value of the RRAMECCnTED register. The read value of the RRAMECCnERDB register is the write value of the RRAMECCnERDB register. 1: The read value of the RRAMECCnTED register can read RAM data. The read value of the RRAMECCnERDB register is the ECC data to be written to RAM.
3	ECREOS	ECC redundant bit output data selection bit This bit is used to identify and select the ECC data to be stored in RAM as ECC data generated for write data or the value of the RRAMECCnERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Store ECC data generated for write data to RAM. 1: Store the value of RRAMECCnERDB register to RAM.

Table 53.21 RRAMECCnTMC register contents (2/2)

Bit position	Bit name	Function
2	ECENS	ECC encoder input selection bit This bit is used to identify the targeted data for ECC data generation as write data to RAM or the value of the RRAMECCnTED register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate ECC data from the write data to RAM. 1: Generate ECC data from the value of the RRAMECCnTED register.
1	ECDCS	ECC decoder input selection bit This bit is used to identify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the RRAMECCnTED register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate syndrome code from RAM data and detect errors. 1: Generate syndrome code from the value of the RRAMECCnTED register and detect errors.
0	ECREIS	ECC redundant bit input data selection bit This bit is used to identify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the RRAMECCnERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate syndrome code from ECC data stored in RAM and detect errors. 1: Generate syndrome code from the value of the RRAMECCnERDB register and detect errors.

53.2.3.4 RRAMECCnTED1 – ECC encode/decode input/output substitution register 1

In ECC test mode, this register handles upper 32 bit data of ECC test data.
The value of the register can be used to generate ECC data or syndrome code.
It is accessible when ECC test mode is enabled (RRAMECCnTMC.ECTMCE = 1)
When RRAMECCnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when RRAM is not accessed.

Access: This register can be accessed in 32-bit units.

Address: <RRAMECCn_base> + 08_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[63:48]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
							R									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[47:32]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 53.22 RRAMECCnTED1 register contents

Bit position	Bit name	Function
31 to 0	ECEDB[63:32]	When RRAMECCnTMC.ECENS = 1, the value of this register is used to generate ECC data and the value this register to RAM is stored to RAM. When RRAMECCnTMC.ECDCS = 1, the value of this register is used to generate syndrome code. In addition, when RRAMECCnTMC.ECTRRS = 1, RAM data [63:32], instead of written data, is read for the value of this register.

53.2.3.5 RRAMECCnTED2 – ECC encode/decode input/output substitution register 2

In ECC test mode, this register handles lower32 bit data of ECC test data.
The value of the register can be used to generate ECC data or syndrome code.
It is accessible when ECC test mode is enabled (RRAMECCnTMC.ECTMCE = 1).
When RRAMECCnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when RRAM is not accessed.

Access: This register can be accessed in 32-bit units.

Address: <RRAMECCn_base> + $0C_H$

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
							R									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 53.23 RRAMECCnTED2 register contents

Bit position	Bit name	Function
31 to 0	ECEDB[31:0]	When RRAMECCnTMC.ECENS = 1, the value of this register is used to generate ECC data and the value this register to RAM is stored to RAM. When RRAMECCnTMC.ECDCS = 1, the value of this register is used to generate syndrome code. In addition, when RRAMECCnTMC.ECTRRS = 1, RAM data [31:0], instead of written data, is read for the value of this register.

53.2.3.6 RRAMECCnTRC – ECC test redundant bit data control register

In ECC test mode, this test register, for ECC data, consists of three 8-bit registers, RRAMECCnHORD, RRAMECCnECRD, and RRAMECCnERDB.
It is accessible when ECC test mode is enabled (RRAMECCnTMC.ECTMCE = 1).
When RRAMECCnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when RRAM is not accessed.

Access: This register can be accessed in 32-bit units.

Address: <RRAMECCn_base> + 10_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	HORD[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECRD[7:0]								ERDB[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 53.24 RRAMECCnTRC register contents (1/2)

Bit position	Bit name	Function
31 to 24	Reserved	Writing to this register is ignored. When ECC test mode is enabled (RRAMECCnTMC.ECTMCE = 1), the read values of these bits are undefined. When ECC test mode is disabled (RRAMECCnTMC.ECTMCE = 0), 00_H is read.
23 to 16	HORD[7:0]	In ECC test mode, this register is used to store ECC data for read RAM data. Writing to this register is ignored. When ECC test mode is enabled (RRAMECCnTMC.ECTMCE = 1), this register can be accessed. When ECC test mode is disabled (RRAMECCnTMC.ECTMCE = 0), 00_H is read. HORD[7:0] ECC code for read RAM data is stored as needed. When RRAMECCnTMC.ECTRRS = 1 and RRAMECCnTED register is read, ECC code is stored.
15 to 8	ECRD[7:0]	In ECC test mode, this is a read-only register for storing ECC data generated for write RAM data. Writing to this register is ignored. When ECC test mode is enabled (RRAMECCnTMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (RRAMECCnTMC.ECTMCE = 0), 00_H is read. ECRD[7:0] These bits can read ECC data generated at the time of RAM data writing, and can read ECC data for the data written to the RRAMECCnTED register when RRAMECCnTMC.ECENS = 1.

Table 53.24 RRAMECCnTRC register contents (2/2)

Bit position	Bit name	Function
7 to 0	ERDB[7:0]	In ECC test mode, this register handles ECC data. The value of this register can be used as ECC data generated at the time of RAM writing or ECC data read at the time of RAM data reading. When ECC test mode is enabled (RRAMECCnTMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (RRAMECCnTMC.ECTMCE = 0), 00_H is read. ERDB[7:0] When RRAMECCnTMC.ECREOS = 1, these bits store the value of this register as ECC data to RAM. When RRAMECCnTMC.ECREIS = 1, the value of this register is read as ECC data read from RAM. When RRAMECCnTMC.ECTRRS = 1, ECC data to be stored in RAM is read instead of the write data for the read value of this register.

53.3 Video RAM (VRAM)

Refer to Section 54, Video RAM and Video RAM Wrapper (VRAM) for a detailed description about the Video RAM.

Section 54 Video RAM and Video RAM Wrapper (VRAM)

54.1 Overview of the Video RAM and Video RAM Wrapper (VRAM)

54.1.1 Units

This microcontroller has the following number of units of the Video RAM and VRAM Wrapper.

Table 54.1 Video RAM and Wrapper overview Units

Video RAM and Wrapper	D1L1	D1L2(H)	D1M1(H) D1M1-V2	D1M1A	D1M2(H)
Video RAM	–	144 KB	1.55 MB	2 x 1.195 MB	2 x 1.55 MB
VRAM Wrapper	–	yes	yes	yes	yes
Names	–	VRAM0	VRAM0	VRAM0, VRAM1	VRAM0, VRAM1

Unit index n

Throughout this section, the individual units of the Video RAM and Wrapper is identified by the index “n” (n = 0, 1), for example VRAMnCTL for the VRAMn control register.

Error detection stage index m

Throughout this section, the individual number of an error detection stage is identified by the index “m” (m = 7), for example the register VRAMnEADm holds the address of the m^{th} error detection.

54.1.1.1 Register addresses <VRAMn_base>

All Video RAM and Wrapper module register addresses are given as address offsets from the individual base addresses <VRAMn_base>.
The <VRAMn_base> addresses of the VRAMn is listed in the following table:

Table 54.2 Register base addresses <VRAMn_base>

VRAMn unit	<VRAMn_base> address
VRAM0	FFFD 1000_H
VRAM1	FFFD 2000_H

54.1.1.2 Interrupts

The Video RAM and Wrapper modules can generate the following interrupt requests:

Table 54.3 VRAMn interrupt requests

VRAMn signals	Function	Connected to
VRAM0:		
INTECCSED	Single bit error detection interrupt	Error Control Module INTECCSEDVRAM*1
INTECCDED	Double bit error detection interrupt	Error Control Module INTECCDEDVRAM*2
INTECCOVF	Overflow detection interrupt	not connected
VRAM1:		
INTECCSED	Single bit error detection interrupt	Error Control Module INTECCSEDVRAM*1*3
INTECCDED	Double bit error detection interrupt	Error Control Module INTECCDEDVRAM*2
INTECCOVF	Overflow detection interrupt	not connected

Note 1. This Error Control Module input signal INTECCSEDVRAM is a logical OR combination of both single bit error detection interrupts.
Refer to Section 48, Error Control Module (ECM) for details.

Note 2. This Error Control Module input signal INTECCDEDVRAM is a logical OR combination of both double bit error detection interrupts.
Refer to Section 48, Error Control Module (ECM) for details.

Note 3. In D1M2(H), INTECCSEDVRAM of VRAM1 has possibility to be lost. To detect single bit error with certainty, read VRAM1CTL register periodically or allocate ranges, which need to generate interrupt, to VRAM0.

54.1.1.3 Reset sources

The VRAM Error Correction Coding modules and their registers are initialized by the following reset signal:

Table 54.4 Reset sources

VRAMn unit	Reset signal
VRAMn	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

54.2 Video RAM Wrapper function overview

The Video RAM Wrapper takes care for densely packing different color format data in the video memory in order to minimize unused memory gaps. It also unwraps the data when it is read, so the data wrapping/unwrapping is transparent to the master, that writes and reads the data.
This way the on-chip Video RAM is virtually enlarged for storing color data formats, which does not occupy an entire 32-bit word.

Video RAM Wrapper features

- Generation and tests of Error Correction Coding (ECC) information for non-wrapped internal 32-bit video RAM data. Refer to Section 54.6, VRAM Error Correction Coding (VRMECC) for details.
  Hence the 32-bit mode - with and without ECC - is suitable for storage of any kind of data, not just color data.
- Several access modes of 32 bit Video RAM accesses from masters:
  - direct 32-bit access without Error Correction Coding (ECC)
  - direct 32-bit access with Error Correction Coding (ECC)
  - wrap/unwrap of 24 bpp RGB888 color data
  - wrap/unwrap of 24 bpp αRGB6666 color data
  - wrap/unwrap of 18 bpp RGB666 color data

  All modes can be used in parallel by using dedicated areas in the memory map, which determine the used data/color mode, refer to Section 54.5, Video RAM and SDRAM memory map.
- Memory bandwidth impact:
  - No impact for packing/unpacking, correct ECC data,
  - Slight decrease with non-64-bit non-burst single accesses

Video RAM organization and VRAM Wrapper color conversion

One VRAM word consists of 72 bit. Within four 72-bit words an integer number of pixel color data of each color format can be stored. Thus the storage of the data is aligned to 4 x 72 bit = 288 bit block.

NOTE

The 288-bit block alignment does not impact the data write/read by the application program, i.e. the user does not need to take care.

54.3 VRAM Transaction Restrictor

The VRAM transaction restrictor can be used to serve cross-connect masters read request with higher priority than write request or vice versa.

It allows to contiguously process a programmable number of read (or write) transactions, while suspending any write (or read, respectively) request.

Transaction restrictor features overview

- The transaction restrictor can control the ratio between read and write transactions
- Three transaction restrictor modes:
  - write transaction restriction mode: gives priority to read requests
  - read transaction restriction mode: gives priority to write requests
  - normal mode: read and write requests are handled with same priority
- Special read transaction restrictor to control read requests issued by the 2D Graphics Processing Unit (GPU2D)

54.3.1 Write transaction restriction mode (VRMTRCTL.RCTL[1:0] = 01_B)

This mode gives priority to read transaction requests by securing a selectable number of read transactions before a write transaction can take place.

The following diagram shows the state transitions, when the write transaction restrictor is enabled.

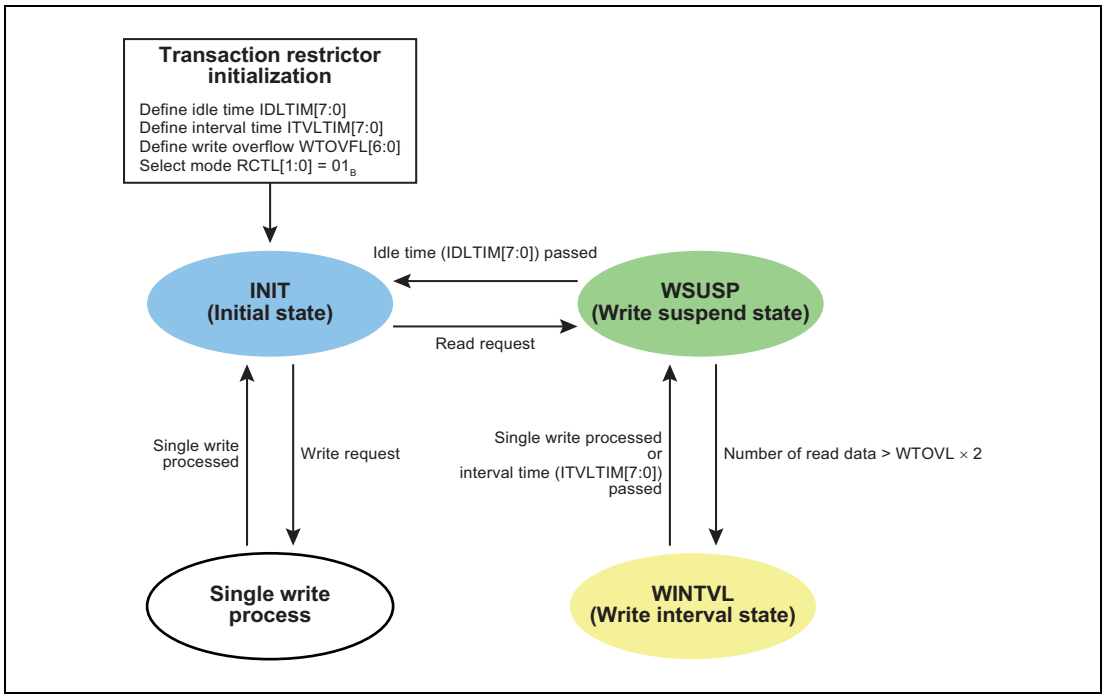


Figure 54.1 State transitions during write transaction restriction (VRMTRCTL.RCTL[1:0] = 01_B)

Initial state (INIT)

After the reset release the INIT state is entered.
Upon a write request a single write transaction is process and the INIT state is re-entered.
Upon a read request transition to the write suspend request state (WSUSP) takes place.

Write suspend state (WSUSP)

In the WSUP state any write request is not served (i.e. write is suspended), until either of the following events occurs:

- The number of read data, determined by VRAMnVRMTRCTL.WTOVFL[6:0], have been processed and the write interval state (WINTVL) is entered.
  The number of read data is WTOVFL[6:0] × 2.
- The idle time has passed without any read request and the INIT state is re-entered.
  The idle time is determined by the number of C_ISO_XCCLK clock cycles defined in VRAMnVRMTRCTL.IDLTIM[7:0].

Write interval state (WINTVL)

In the WINTVL state a single write transaction can take place, if a write request has occurred during WSUSP state.
Completion of a single write transaction causes an immediate return to the WSUSP state.
If no write request has occurred during WSUSP state the WINTVL state is left after the interval time has passed and the WSUSP state is entered again.
The interval time is determined by the number of C_ISO_XCCLK clock cycles defined in VRAMnVRMTRINTVL.ITVLTIM[7:0].
After return to the WSUSP state the next read requests can be served.

54.3.2 Read transaction restriction mode (VRMTRCTL.RCTL[1:0] = 10_B)

This mode gives priority to write transaction requests by securing a selectable number of write transactions before a read transaction can take place.

The following diagram shows the state transitions, when the read transaction restrictor is enabled.

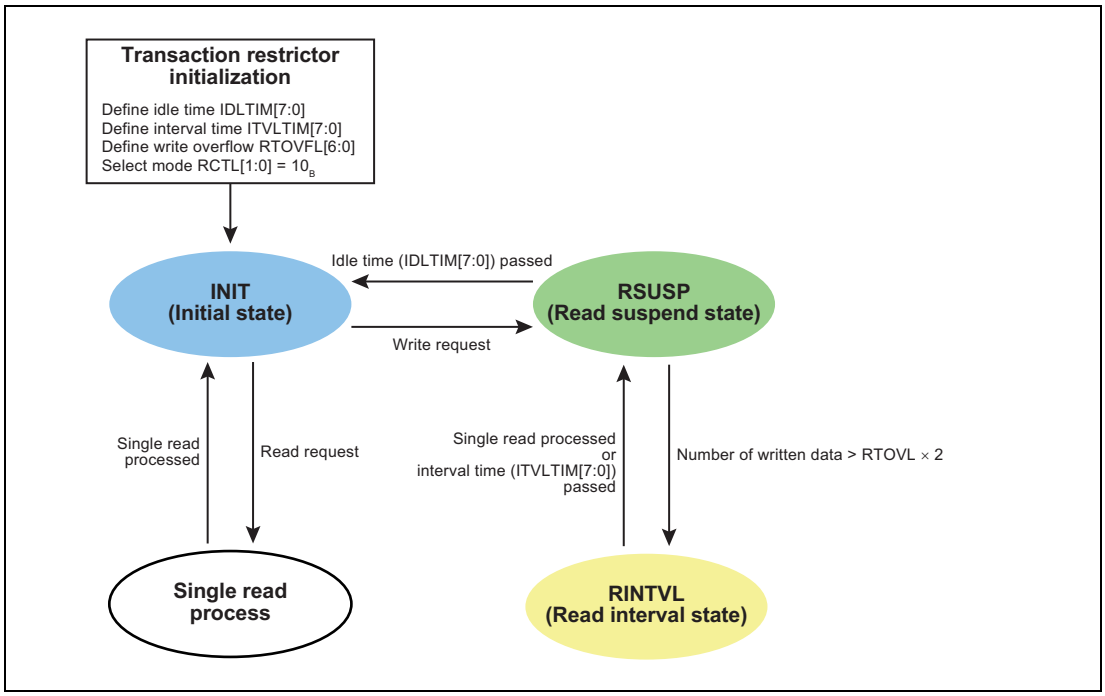


Figure 54.2 State transitions during read transaction restriction (VRMTRCTL.RCTL[1:0] = 10_B)

Initial state (INIT)

After the reset release the initial state (INIT) is entered.
Upon a read request a single read transaction is process and the INIT state is re-entered.
Upon a write request transition to the read suspend request state (RSUSP) takes place.

Read suspend state (RSUSP)

In the RSUSP state any read request is not served (i.e. read is suspended), until either of the following events occurs:

- The number of written data, determined in VRAMnVRMTRCTL.RTOVFL[6:0], have been processed and the read interval state (RINTVL) is entered.
  The number of written data is RTOVFL[6:0] × 2.
- The idle time has passed without any write request and the INIT state is re-entered.
  The idle time is determined by the number of C_ISO_XCCLK clock cycles defined in VRAMnVRMTRCTL.IDLTIM[7:0].

Read interval state (RINTVL)

In the RINTVL state a single read transaction can take place, if a read request has occurred during RSUSP state.
Completion of a single read transaction causes an immediate return to the RSUSP state.
If no read request has occurred during RSUSP state the RINTVL state is left after the interval time has

passed and the RSUSP state is entered again.
The interval time is determined by the number of C_ISO_XCCLK clock cycles defined in VRAMnVRMTRINTVL.ITVLTIM[7:0].
After return to the RSUSP state the next write requests can be served.

54.3.3 Normal mode (VRMTRCTL.RCTL[1:0] = 00_B)

In normal mode all read and write requests are handled with same priority.

54.3.4 VRAM Transaction Restrictor Registers

The VRAM transaction restrictor is controlled and operated by means of the following registers

Table 54.5 VRAM transaction restrictor registers overview

Register Name	Shortcut	Address
VRAMn transaction restrictor control register	VRAMnVRMTRCTL	<VRAMn_base> + 40_H
VRAMn transaction restrictor interval time register	VRAMnVRMTRINTVL	<VRAMn_base> + 44_H

<VRAMn_base>

The base addresses <VRAMn_base> of the VRAMn is defined in “Register base addresses <VRAMn_base>” in the section above.

54.3.4.1 VRAMnVRMTRCTL – VRAMn transaction restrictor control register

These registers control the transaction restrictor operation mode for VRAMn.

CAUTION

This register must not be changed while a VRAM access is ongoing.

Access: This register can be accessed in 32-bit units.

Address: <VRAMn_base> + 40_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	IDLTIM[7:0]								0	0	0	0	0	0	RCTL[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RTOVFL[6:0]							DHDTREN*1	WTOVFL[6:0]							0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Note 1. The GPU2D read transaction restrictor control bit DHDTREN is only available in the VRAM0VRMTRCTL register.

Table 54.6 VRAMnVRMTRCTL register contents

Bit position	Bit name	Function
31 to 24	IDLTIM[7:0]	IDLTIM[7:0] defines the idle time in number of C_ISO_XCCLK clock cycles. The idle time is the time to change from write suspend (WSUSP) or read suspend (RSUSP) state to initial state (INIT).
23 to 18	Reserved	When read, the value after reset is read. When written, write the value after reset.
17 to 16	RCTL[1:0]	Transaction restrictor mode 00_B: normal mode (transaction restrictor disabled) 01_B: write transaction restrictor mode 10_B: read transaction restrictor mode 11_B: setting prohibited
15 to 9	RTOVFL[6:0]	Read overflow count RTOVFL[6:0] determines the number of written data in read suspend state (RSUSP) before changing to read interval state (RINTVL). The actual number of written data is RTOVFL[6:0] × 2. RTOVFL[6:0] is only valid in read transaction restrictor mode, i.e. RTCL[1:0] = 10_B.
8	DHDTREN	2D Graphics Processing Unit (GPU2D) read transaction restrictor control 0 : GPU2D read transaction restrictor disable 1 : GPU2D read transaction restrictor enable **Note:** DHDTREN is only available in the VRAM0VRMTRCTL register.
7 to 1	WTOVFL[6:0]	Write overflow count WTOVFL[6:0] determines the number of read data in write suspend state (WSUSP) before changing to write interval state (WINTVL). The actual number of read data is WTOVFL[6:0] × 2. WTOVFL[6:0] is only valid in write transaction restrictor mode, i.e. RTCL[1:0] = 01_B.
0	Reserved	When read, the value after reset is read. When written, write the value after reset.

54.3.4.2 VRAMnVRMTRINTVL – VRAMn transaction restrictor interval time register

These registers define the state change timing of from interval states to suspend states.

CAUTION

This register must not be changed while a VRAM access is ongoing.

Access: This register can be accessed in 32-bit units.

Address: <VRAMn_base> + 44_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	ITVLTIM[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 54.7 VRAMnVRMTRINTVL register contents

Bit position	Bit name	Function
7 to 0	ITVLTIM[7:0]	ITVLTIM[7:0] defines the interval time in number of C_ISO_XCCLK clock cycles. The interval time is the time to change from write or read interval state (WINTVL or RINTVL) to write or read suspend state (WSUSP or RSUSP). ITVLTIM[7:0] is only valid in write or read transaction restrictor mode, i.e. RTCL[1:0] = 01_B or 10_B.

54.3.5 2D Graphics Processing Unit (GPU2D) read transaction restrictor

The GPU2D read transaction restrictor restricts read transactions, requested by the GPU2D.

By enabling this restrictor more transfer bandwidth is granted to other masters, which are accessing the Video RAMs.

NOTES

1. In particular, when the Video Output Interfaces require more bandwidth to sustain a continuous video output data flow, enable the GPU2D read transaction restrictor.
2. The GPU2D transaction restrictor does not affect GPU2D write transactions.

The GPU2D read transaction restrictor operation is controlled via the VRAM0 transaction restrictor control register VRAM0VRMTRCTL:

- DHDTREN = 0: GPU2D read transaction restrictor disabled
- DHDTREN = 1: GPU2D read transaction restrictor enabled
  GPU2D read transactions are limited to a single read process in order to write accesses.

NOTE

VRAM0VRMTRCTL.DHDTREN controls the GPU2D read transaction restrictor for both Video RAMs VRAM0 and VRAM1.

In D1M1(H), D1M1-V2 and D1M1A the GPU2D read transaction restrictor effects the read transaction to all XC Memory slaves.

54.3.6 2D Graphics Processing Unit (GPU2D) write transaction restrictor (D1M1-V2, D1M1A only)

The GPU2D write transaction restrictor restricts write transactions, requested by the GPU2D.
By enabling this restrictor more transfer bandwidth is granted to other masters, which are accessing the Video RAMs.

NOTES

1. In particular, when the Video Output Interfaces require more bandwidth to sustain a continuous video output data flow, enable the GPU2D write transaction restrictor.
2. The GPU2D write transaction restrictor does not affect GPU2D read transactions.

The GPU2D write transaction restrictor operation is controlled via the GPU2D transaction restrictor control register GPU2DTRCTL:

- DHDTREN = 0: GPU2D write transaction restrictor disabled
- DHDTREN = 1: GPU2D write transaction restrictor enabled
  GPU2D write transactions are limited to a single write process in order to read accesses.

NOTE

GPU2DTRCTL.DHDTREN controls the GPU2D write transaction restrictor for all XC Memory slaves.

54.3.6.1 GPU2DTRCTL – GPU2D transaction restrictor control register

This register controls the transaction restrictor operation mode for GPU2D.

CAUTION

This register must not be changed while a GPU2D access is ongoing.

Access: This register can be accessed in 32-bit units.

Address: FFC0 605C$_H$

Initial value: 0000 0000$_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	DHD TREN	0	0	0	0	0	0	0	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R

Table 54.8 GPU2DTRCTL register contents

Bit position	Bit name	Function
8	DHDTREN	2D Graphics Processing Unit (GPU2D) write transaction restrictor control 0: GPU2D write transaction restrictor disable 1: GPU2D write transaction restrictor enable

NOTE

In the header files the module name of the above register is defined as:

SELB.

54.4 VRAM Wrapper data handling

54.4.1 Direct modes

The direct modes do not manipulate the data and thus can be used for storage of any kind of data, not just for 32 bpp αRGB8888 color data.

64-, 32,-, 16- and 8-bit access modes are supported.

54.4.1.1 Direct 32-bit mode without ECC

9 32-bit words / 288-bit block

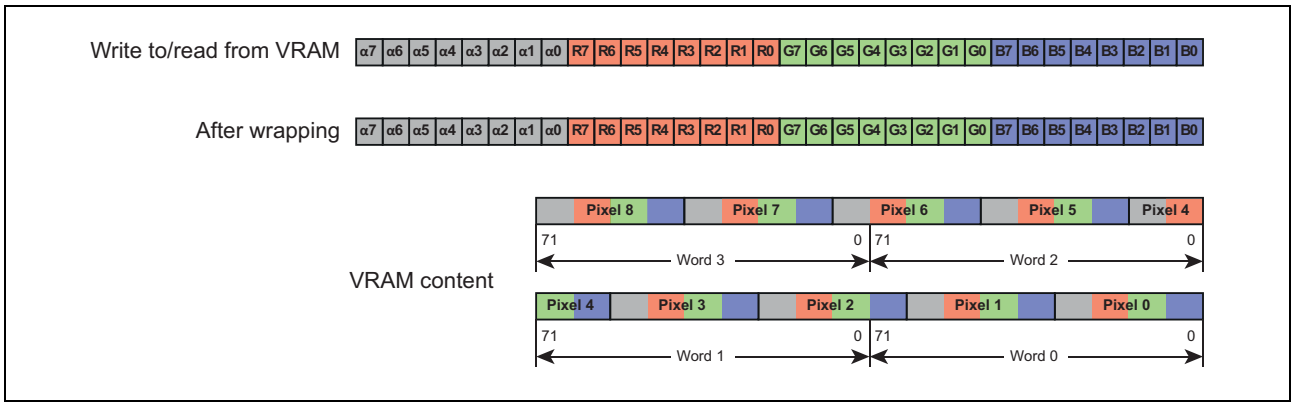


Figure 54.3 Direct 32-bit mode without ECC

54.4.1.2 Direct 32-bit mode with ECC

8 32-bit words/ 288-bit block

Due to ECC secured data storage this mode can also be used to store safety relevant data, e.g. used by the CPU.

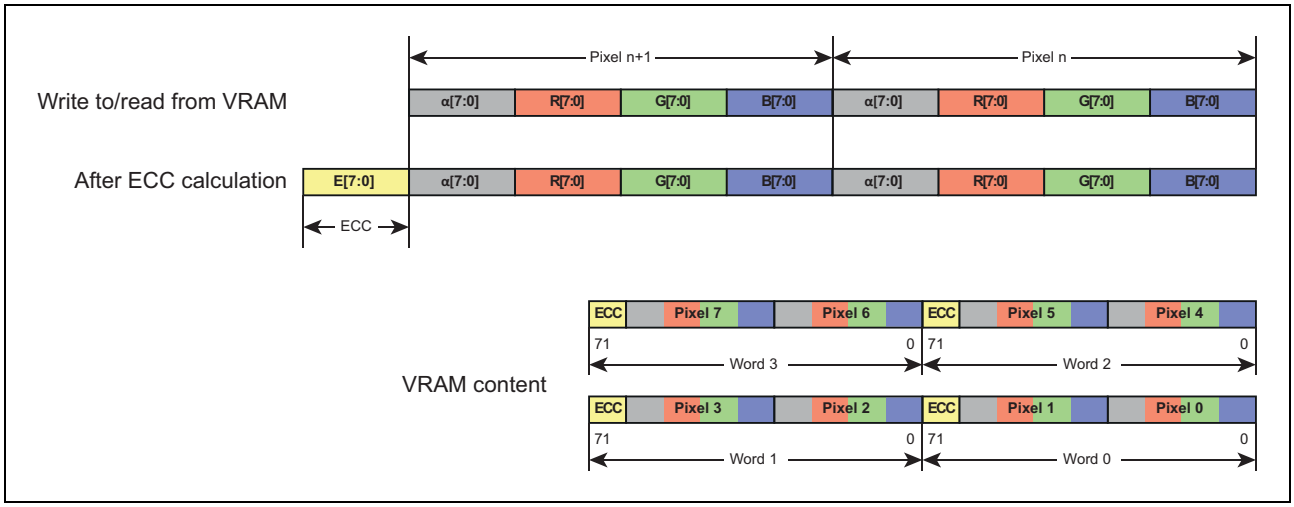


Figure 54.4 Direct 32-bit mode with ECC

54.4.2 Modes with color conversion

The color conversion modes wrap and unwrap the color data in order to utilize the video RAM capacity efficiently.

54.4.2.1 Wrapped 24 bpp RGB888

3 pixel / 72-bit word, 12 pixel / 288-bit

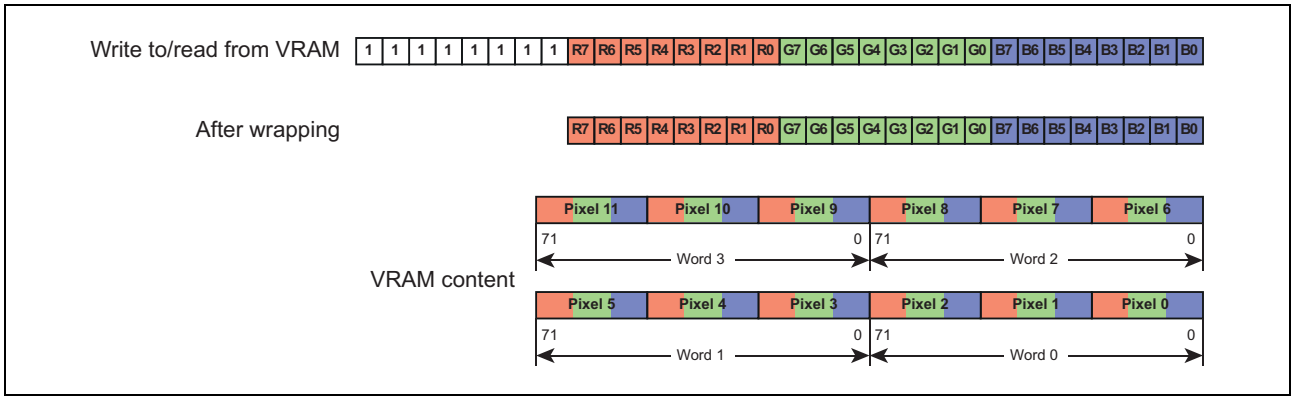


Figure 54.5 Wrapped 24 bpp RGB888

NOTE

The upper 8 bit of the 32-bit word, written to the VRAM, are ignored.
At read these bits are set to FF_H, which can be interpreted as α = 1.

54.4.2.2 Wrapped 24 bpp αRGB6666

3 pixel / 72-bit word, 12 pixel / 288-bit block

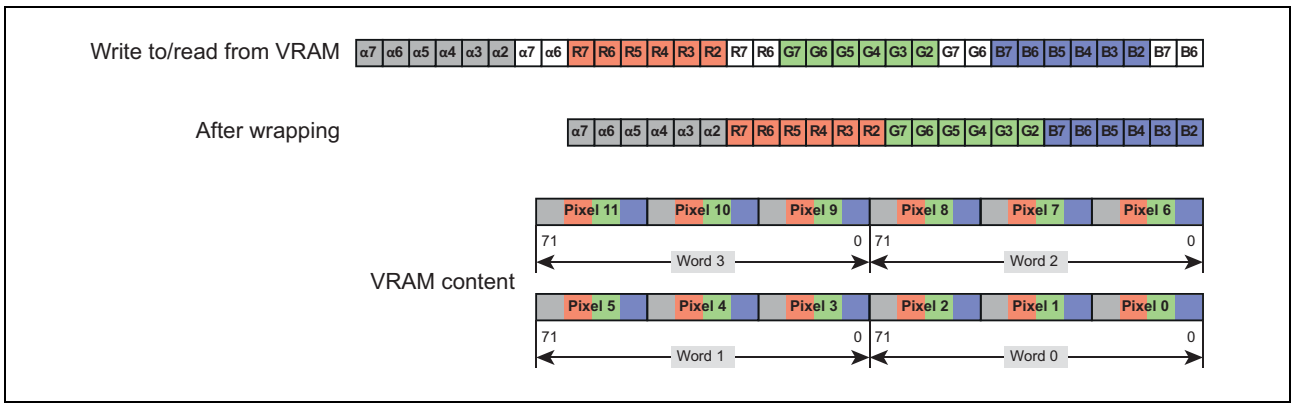


Figure 54.6 Wrapped 18 bpp αRGB6666

54.4.2.3 Wrapped 18 bpp RGB666

4 pixel / 72-bit word, 16 pixel / 288-bit block

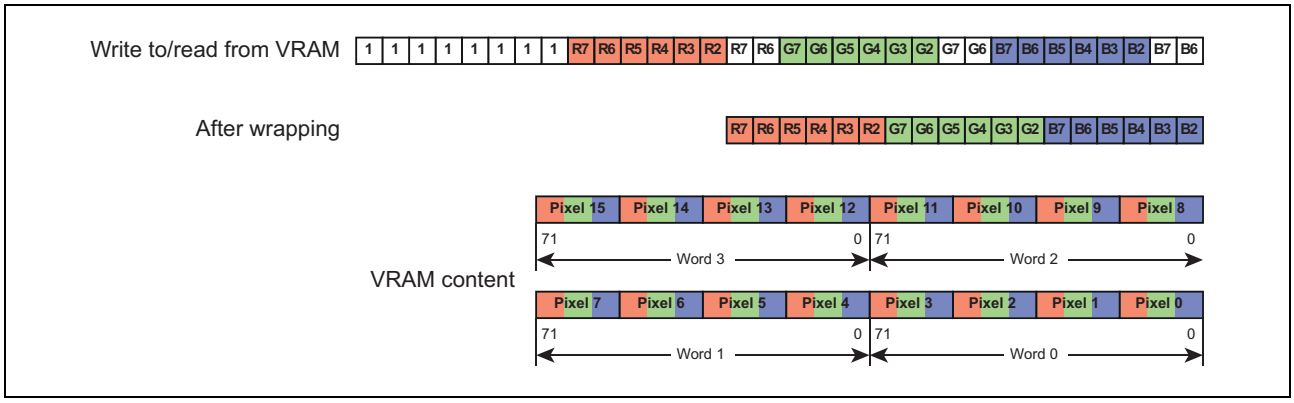


Figure 54.7 Wrapped 24 bpp RGB666

NOTES

1. The upper 8 bit of the 32-bit word, written to the VRAM, are ignored.
   At read these bits are set to FF_H, which can be interpreted as α = 1.
2. The LSBs of the R, G and B value of the 32-bit word, written to the VRAM, are ignored.
   At read these bits are stuffed with the two MSBs [7:6] of the respective color value.

54.5 Video RAM and SDRAM memory map

The direct and wrap/unwrap modes for different color formats are selected by accessing the Video RAM via different address areas through the Mirror/VRAM Wrapper window in the cross-connect address space.

When using the VRAM in wrap mode the lower 128 MB portion of the external SDRAM and the 512 MB Serial Flash Memory area is mirrored contiguously mapped above the VRAM, so a larger continuous address space can be utilized without any address gaps.

54.5.1 D1L2(H) Mirror/VRAM Wrapper window memory map

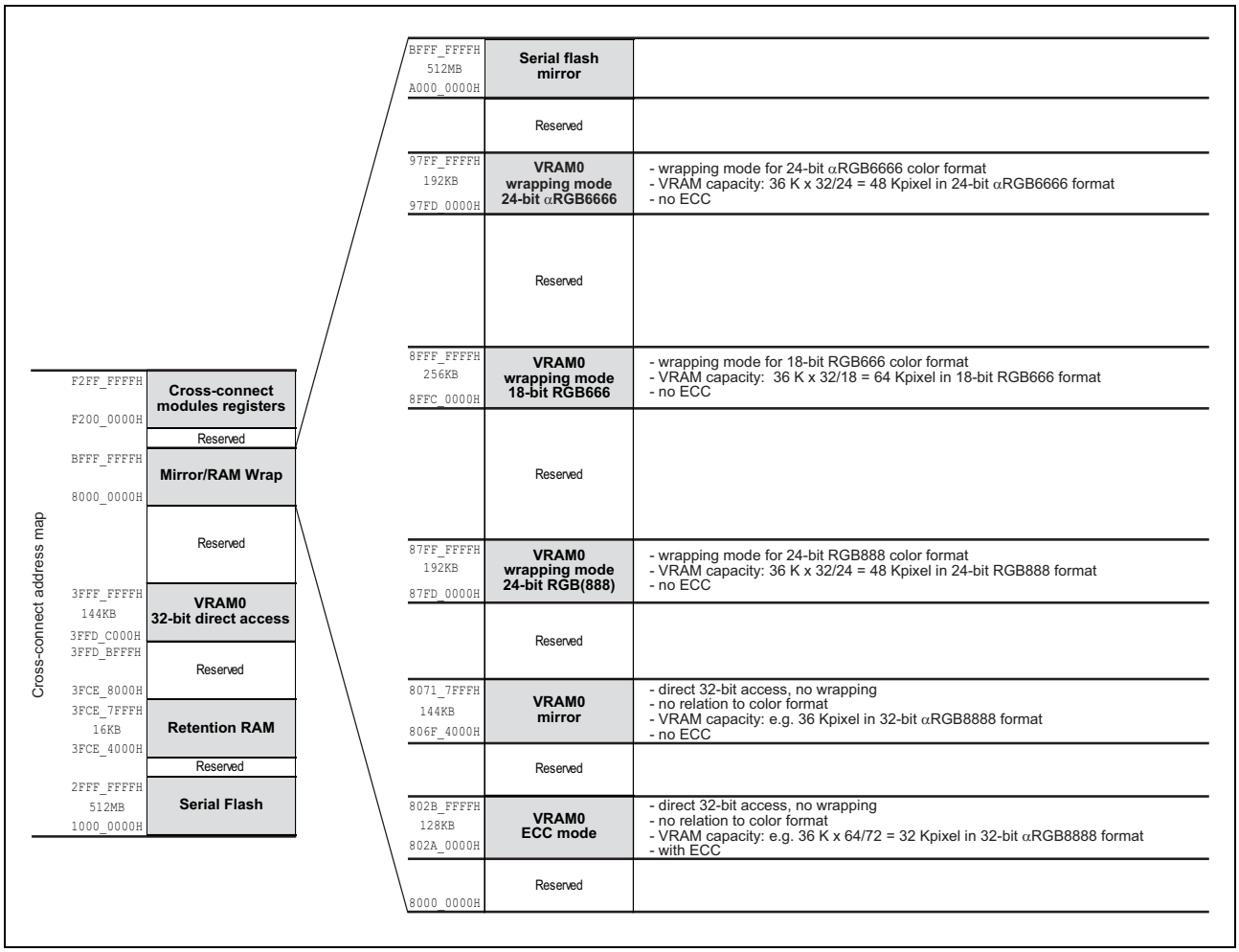


Figure 54.8 D1L2(H) Mirror/RAM Wrap mapping

54.5.2 D1M1(H) Mirror/VRAM Wrapper window memory map

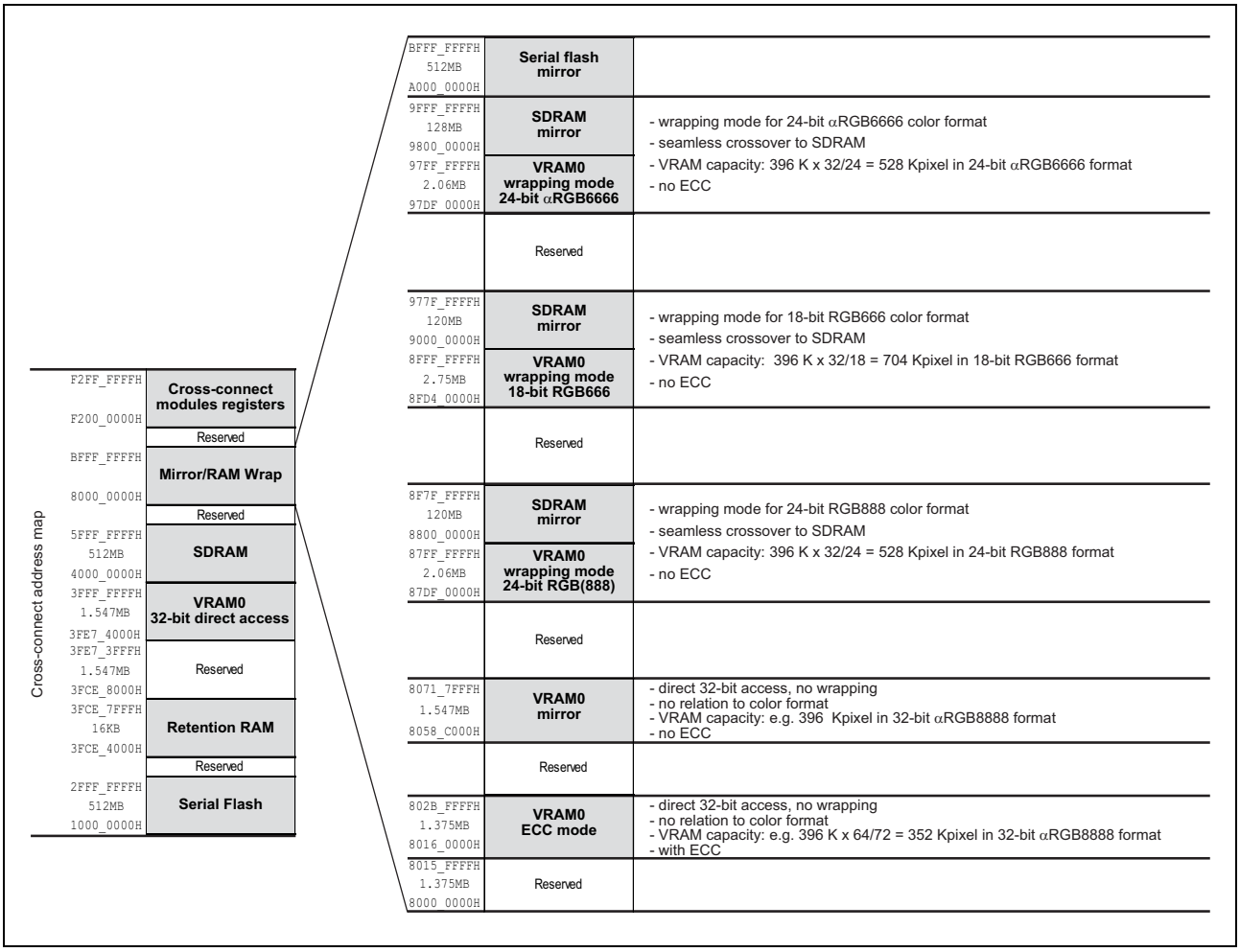


Figure 54.9 D1M1(H) Mirror/RAM Wrap mapping

54.5.3 D1M1-V2 Mirror/VRAM Wrapper window memory map

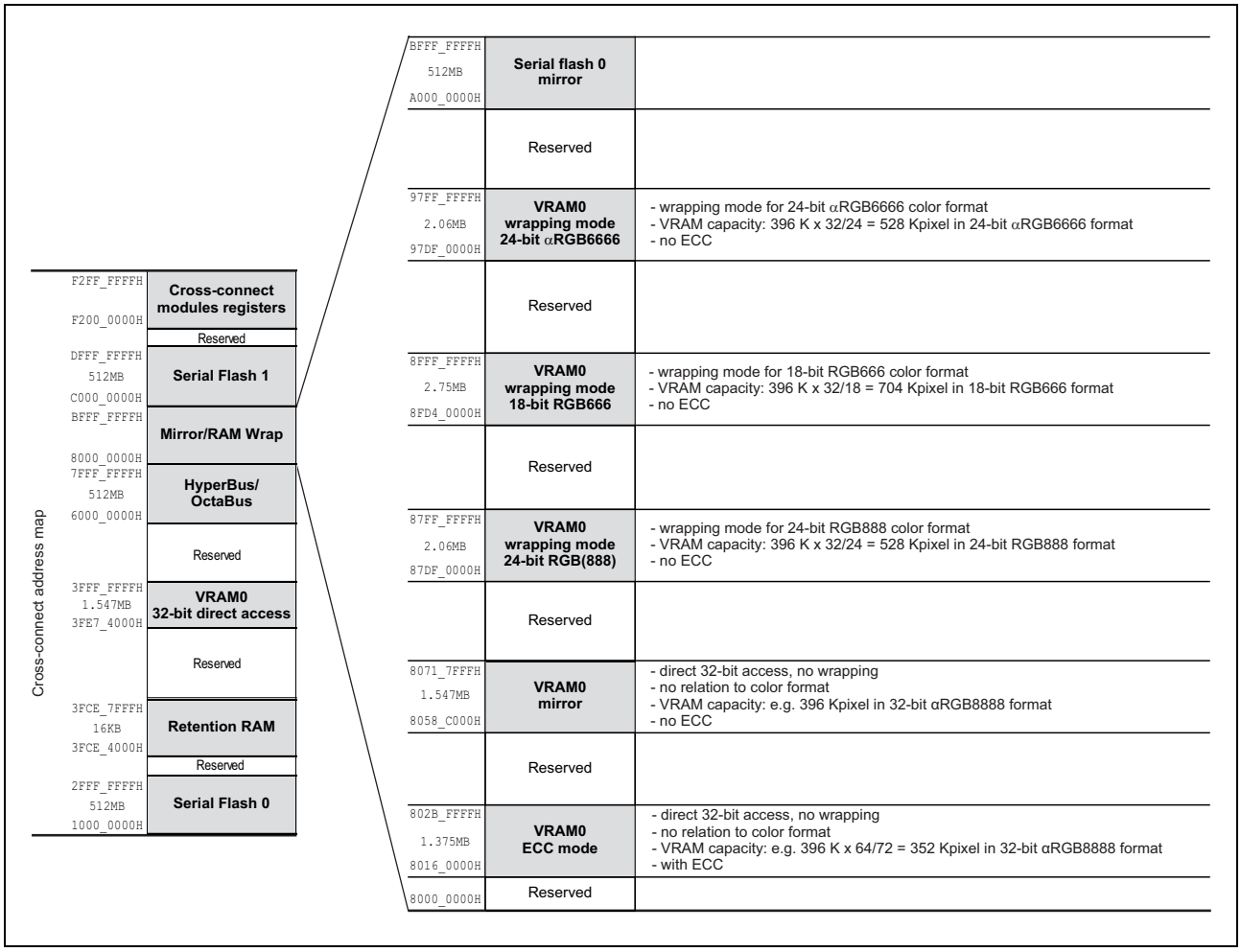


Figure 54.10 D1M1-V2 Mirror/RAM Wrap mapping

54.5.4 D1M1A Mirror/VRAM Wrapper window memory map

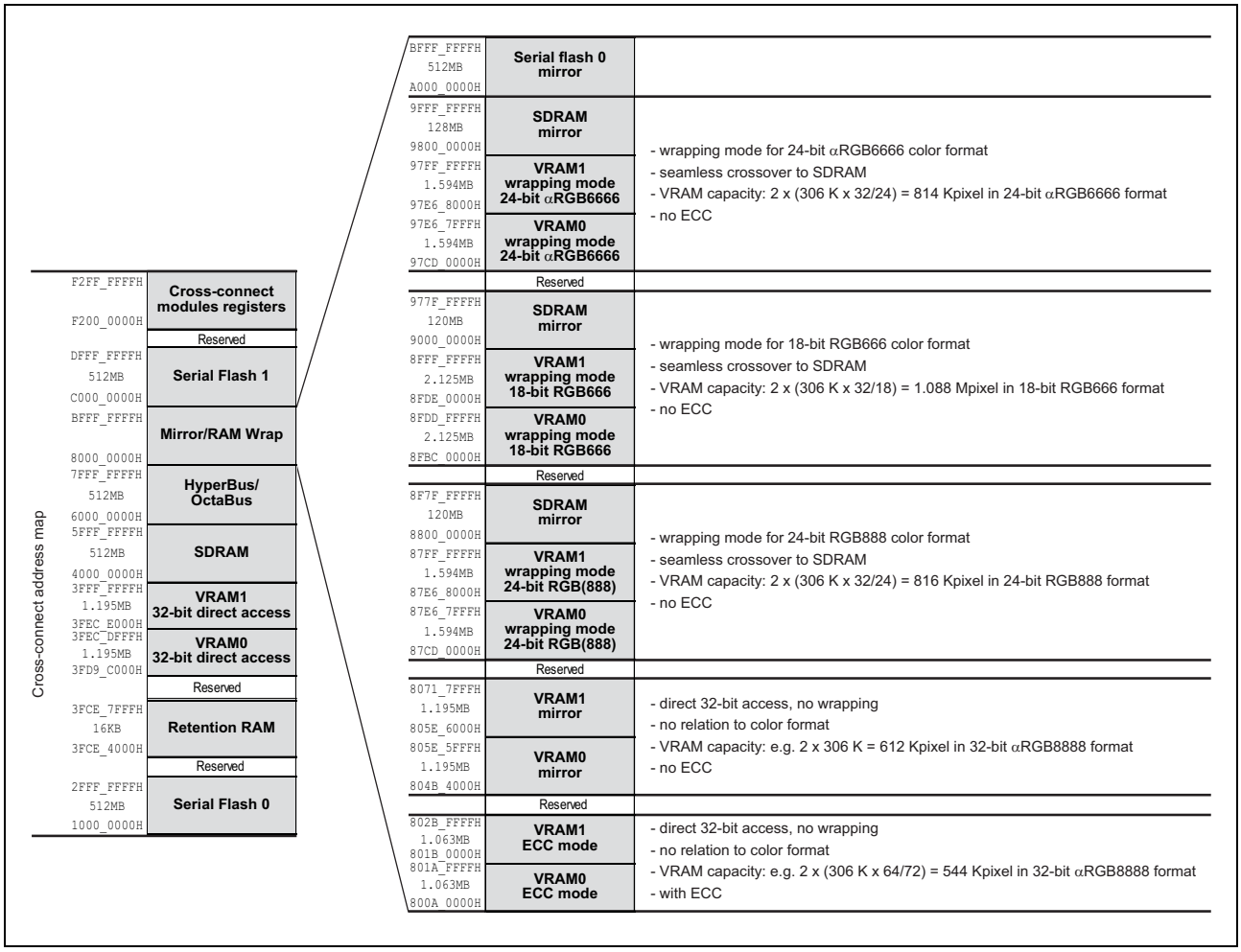


Figure 54.11 D1M1A Mirror/RAM Wrap mapping

54.5.5 D1M2(H) Mirror/RAM Wrapper window memory map

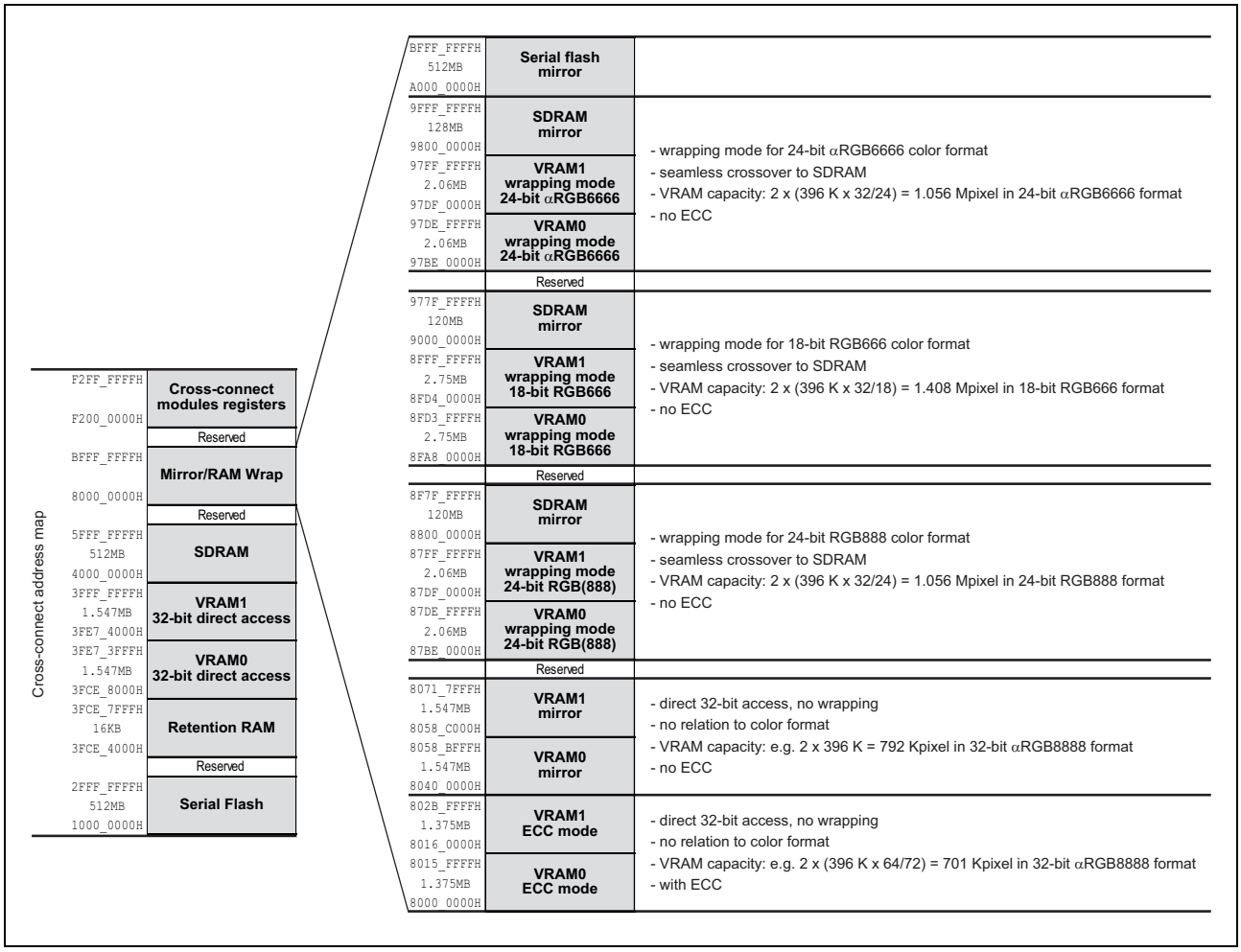


Figure 54.12 D1M2(H) Mirror/RAM Wrap mapping

NOTE

If D1M2(H) uses the DDR2-SDRAM bigger than 128 Mbyte, the D1M2(H) mirror address is offset as follows:

- 256 Mbyte of DDR2-SDRAM
  - D1M2(H) address = 8800 0000_H to 8F7F $FFFF_H$ →
    DDR2-SDRAM memory address = 0800 0000H to 0F7F FFFFH
  - D1M2(H) address = 9000_0000_H to 977F $FFFF_H$ →
    DDR2-SDRAM memory address = 0000 0000_HH to 077F $FFFF_H$
  - D1M2(H) address = 9800 0000_H to 9FFF $FFFF_H$ →
    DDR2-SDRAM memory address = 0800 0000_H to 0FFF $FFFF_H$

- 512 Mbyte of DDR2-SDRAM
  - D1M2(H) address = 8800 0000_H to 8F7F $FFFF_H$ →
    DDR2-SDRAM memory address = 0800 0000_H to 0F7F $FFFF_H$
  - D1M2(H) address = 9000 0000_H to 977F $FFFF_H$ →
    DDR2-SDRAM memory address = 1000 0000_H to 177F $FFFF_H$
  - D1M2(H) address = 9800 0000_H to 9FFF $FFFF_H$ →
    DDR2-SDRAM memory address = 1800 0000_H to 1FFF $FFFF_H$

54.6 VRAM Error Correction Coding (VRMECC)

54.6.1 Functional Description

The VRAM Error Correction Coding (VRMECC) module generates and tests Error Correction Coding (ECC) information for 32-bit Video RAM data.

8 bit ECC information is calculated and added to each 64-bit data unit, that is written to the Video RAM through the area for direct 32-bit access with ECC.

During read-back of the data through this area the ECC information is checked against the read data.

Upon detection of a single erroneous bit in a 64-bit data unit

- the erroneous bit of the read data is corrected (but the erroneous memory data is retained)
- a Single Error Detection (SED) flag VRAMnCTL.ECSEDFm is set
- the read access address is stored in the VRAMnEADm register
- an INTECCSED interrupt is generated

Upon detection of two erroneous bits in a 64-bit data unit

- a Double Error Detection (DED) flag VRAMnCTL.ECDEDFm is set
- the read access address is stored in the VRAMnEADm register
- an INTECCDED interrupt is generated

NOTE

The write access with less than 64 bit to the ECC area is prohibited.

For ECC support in case of access size other than 64 bit it is mandatory to use cached or buffered access to the VRAM.

54.6.1.1 VRAM initialization

Before reading from the VRAM, while the ECC error detection and correction is enabled, the entire VRAM must be initialized.

CAUTIONS

1. **VRAM initialization must be performed by 64-bit write accesses in order to ensure that all VRAM data is initialized and the respective ECC value fits to the VRAM content.**
2. **Reading from un-initialized VRAM may lead to the detection of ECC errors**

54.6.1.2 Error detection

(1) Error detection stage

Information about up to eight error detections are stored, i.e.

- up to eight read addresses with errors are stored in the error address registers VRAMnEADm, with m = 0 to 7
- single or double error detection indication is stored in the ECSEDFm and ECDEDFm of the ECC control register VRAMnCTL.

The error detection flags ECSEDFm and ECDEDFm are related to the error address stored in VRAMnEADm. I.e. each error detection address has separate assigned error flags.

The address of the detected errors are stored in ascending order of m, starting from VRAMnEAD0.

(2) Error information clear

Setting either VRAMnCTL.ECER1C = 1 or VRAMnCTL.ECER2C = 1 clears all error detection flags ECER1F and ECER2F in the ECC control register VRAMnCTL.

Additionally all ECC error address registers VRAMnEADm are set to their default value $0000\ 0000_H$.

(3) Error detection stages overflow

An error detection while all error detection stages hold valid error indications (i.e. any of the error flags ECSEDFm or ECDEDFm of stage m is active), are treated as follows:

- If the current error detection reveals a single bit error, all error flags ECSEDFm or ECDEDFm and error addresses VRAMnEADm are not changed.
- If the current error detection reveals a double bit error and
  - all error detection stages m = 0 to 7 indicate single bit errors (i.e. ECSEDF[7:0] = 1 and ECDEDF[7:0] = 0),
    - the current error address overwrites the previous error address in VRAMnEAD7
    - the double bit error flag is set, i.e. ECDEDF7 = 1. The double bit error interrupt INTECCDEDVRAM is asserted, provide this is permitted by VRAMnCTL.EC2EDIC = 1.
  - if any of the error detection stages m = 0 to 7 indicate a double bit errors (i.e. any ECDEDF[7:0] = 1) all error flags ECSEDFm or ECDEDFm and error addresses VRAMnEADm are not changed.

In any of the above cases the overflow flag VRAMnCTL.ECOVFF is set.

(4) Additional error flags

- VRAMnCTL.ECER2F: Double bit error detection flag
  This flag is set with the 1st occurrence of a double bit error. It is cleared by setting VRAMnCTL.ECER2C = 1.
- VRAMnCTL.ECER1F: Single bit error detection flag
  This flag is set with the 1st occurrence of a single bit error. It is cleared by setting VRAMnCTL.ECER1C = 1.
- VRAMnCTL.ECEMF: ECC error has occurred during the last read
  This bit is cleared by the next read if the ECC value is correct.

The detailed description is given in Section 54.6.2.1, VRAMnCTL – ECC control register.

(5) Error interrupt control

The generation of error detection interrupts can be controlled by control bits of the VRAMnCTL register:

- EC2EDIC = 0/1 disables/enables generation of the double error detection interrupt INTECCDED
- EC1EDIC = 0/1 disables/enables generation of the single error detection interrupt INTECCSED

54.6.1.3 ECC mode control

The ECC module can be set in two different modes:

- VRAMnCTL.ECTHM = 0: normal operation mode
  Error judgement can be enabled/disabled by VRAMnCTL.ECERVF
- VRAMnCTL.ECTHM = 1: pass through mode
  ECC calculation and storage of the ECC bits in the RAM is performed even in pass through mode. ECC decoding upon RAM data read is not executed.

The following table gives an overview about the different modes:

Generation of the 8-bit ECC data during write of data to the VRAM through the direct access 32-bit window with ECC check is not affected by VRAMnCTL.ECTHM.

Table 54.9 ECC modes overview

ECC mode setting		Error correction	Error address storage	Interrupts assertion	Error flags settings
ECTHM*1	ECERVF*1	Single bit errors	VRAMnEADm*2, ECDEDFm*1, ECSEDFm*1	INTECCDED, INTECCSED	ECER2F, ECER1F, ECEMF*1
0: normal operation	0: error judgement disabled	yes	no	no	no
	1: error judgement enabled	yes	yes	yes	yes
1: pass through	x	no	no	no	no

Note 1. Bits in the ECC control register VRAMnCTL.
Note 2. ECC error address registers.

ECC mode setting protection

Modification of the ECC mode control bits ECTHM and ECERVF in the VRAMnCTL register is depends on two protection bits in the VRAMnCTL register:

- ECMA[1:0] = 01_B: modification of ECTHM and ECERVF is possible
- All other ECMA[1:0] settings: modification of ECTHM and ECERVF is not possible

CAUTION

Since an 8-bit ECC code is first calculated during the write to the respective 64-bit data, the ECC code does most likely not fit to the data unit after power-up of the VRAM.

Consequently a read of uninitialized VRAM content will most likely lead to ECC error detections.

54.6.1.4 ECC module test

The functionality of the ECC module can be tested by provoking ECC errors:

1. Write data to the VRAM through the direct 32-bit access window without ECC coding.
2. Read the data through the direct 32-bit access window with ECC coding.

Step 2 will result in an ECC error detection, if the 8 bit, treated as ECC code in the next step, do not hold a correct ECC code to the respective 64-bit data unit.

54.6.2 Registers

This section contains a description of all registers of the VRAM ECC.

Table 54.10 VRAM ECC registers overview

Register Name	Shortcut	Address
ECC control register	VRAMnCTL	<VRAMn_base> + 00_H
ECC error address register 0	VRAMnEAD0	<VRAMn_base> + 14_H
ECC error address register 1	VRAMnEAD1	<VRAMn_base> + 18_H
ECC error address register 2	VRAMnEAD2	<VRAMn_base> + $1C_H$
ECC error address register 3	VRAMnEAD3	<VRAMn_base> + 20_H
ECC error address register 4	VRAMnEAD4	<VRAMn_base> + 24_H
ECC error address register 5	VRAMnEAD5	<VRAMn_base> + 28_H
ECC error address register 6	VRAMnEAD6	<VRAMn_base> + $2C_H$
ECC error address register 7	VRAMnEAD7	<VRAMn_base> + 30_H
ECC test mode control register	VRAMnTMC	<VRAMn_base> + 04_H
ECC encode/decode input/output substitution register 1	VRAMnTED1	<VRAMn_base> + 08_H
ECC encode/decode input/output substitution register 2	VRAMnTED2	<VRAMn_base> + $0C_H$
ECC test redundant bit data control register	VRAMnTRC	<VRAMn_base> + 10_H

<VRAMn_base>

The base addresses <VRAMn_base> of the VRAMn is defined in “Register base addresses <VRAMn_base>” in the section above.

NOTE

Memory access right after register access has possibility to cause the racing between two accesses via different routes.
Therefore, the last accessed register value must be confirmed at least before memory access. Refer to Section 3.4.1, Synchronization of Store Instruction Completion and Subsequent Instruction Generation for details.

54.6.2.1 VRAMnCTL – ECC control register

This register holds various status and control bits.

Access: This register can be accessed in 32-bit units.

Address: <VRAMn_base> + 00_H

Initial value: 0000 0000 0000 0000 0000 0000 0001 $000x_B$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECDED F7	ECSED F7	ECDED F6	ECSED F6	ECDED F5	ECSED F5	ECDED F4	ECSED F4	ECDED F3	ECSED F3	ECDED F2	ECSED F2	ECDED F1	ECSED F1	ECDED F0	ECSED F0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	EMCA[1:0]		0	0	EC OVFF	EC ER2C	EC ER1C	0	EC THM	EC ERVF	EC 1ECP	EC 2EDIC	EC 1EDIC	EC ER2F	EC ER1F	EC EMF
Initial value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	x
R/W	R/W	R/W	R	R	R	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R	R

Table 54.11 VRAMnCTL register contents (1/3)

Bit position	Bit name	Function
31 to 16	ECDEDFm ECSEDFm	These bits indicate the type of error, that is related to the stored error address of stage m in the VRAMnEADm register. • ECDEDFm = 0 and ECSEDFm = 0 No error address stored in VRAMnEADm. • ECDEDFm = 1 and ECSEDFm = 0 VRAMnEADm address of double error detection • ECDEDFm = 0 and ECSEDFm = 1 VRAMnEADm address of single error detection • ECDEDFm = 1 and ECSEDFm = 1 Does not occur. These bits are cleared under following conditions: • While ECER2F bit is set, writing VRAMnCTL.ECER2C = 1 clears ECDEDF7-0 and ECSEDF7-0 While ECER1F bit is set, writing VRAMnCTL.ECER1C = 1 clears ECDEDF7-0 and ECSEDF7-0 • ECC pass through mode (VRAMnCTL.ECTHM = 1) • ECC error judgment disabled (VRAMnCTL.ECERVF = 0)
15 to 14	EMCA[1:0]	ECC mode protection • EMCA[1:0] = 01_B: modification of the ECC mode control bits ECTHM and ECERVF is permitted • All other EMCA[1:0] setting: modification of the ECC mode control bits ECTHM and ECERVF is not possible
13 to 12	Reserved	When read, the value after reset is read. When written, write the value after reset.

Table 54.11 VRAMnCTL register contents (2/3)

Bit position	Bit name	Function
11	ECOVFF	Overflow detection flag In case of an error detection, while all error address registers VRAMnEADm hold already a valid error address, this overflow flag is set. 0: no overflow detected 1: overflow detected Writing to this bit has no effect. These bits are cleared under following conditions: • Writing VRAMnCTL.ECER2C = 1 while ECER2F bit is set • Writing VRAMnCTL.ECER1C = 1 while ECER1F bit is set • ECC pass through mode (VRAMnCTL.ECTHM = 1) • ECC error judgment disabled (VRAMnCTL.ECERVF = 0)
10	ECER2C	Clear double bit error detection flag ECER2F 0: no function 1: clears ECER2F Reading this bits returns always 0.
9	ECER1C	Clear single bit error detection flag ECER1F 0: no function 1: clears ECER1F Reading this bits returns always 0.
8	Reserved	When read, the value after reset is read. When written, write the value after reset.
7	ECTHM	Pass through mode enable bit. 0: Normal operation mode 1: Pass through mode enable. ECC decoding upon RAM data read is not executed, i.e. no error judgment or the bit correction is performed. ECC calculation and storage of the ECC bits in the RAM is performed even in pass through mode. Modification of this bit is only possible, if EMCA[1:0] = 01_B. Otherwise any write to this bit is ignored.
6	ECERVF	ECC error judgment enable 0: error judgement disabled 1: error judgement enabled Error judgement is only effective in normal operation mode, i.e. if VRAMnCTL.ECTHM = 0. If error judgement is disabled no interrupts are asserted in case of a single or double bit error detection. However detected single bit errors are corrected. Modification of this bit is only possible, if EMCA[1:0] = 01_B. Otherwise any write to this bit is ignored.
5	EC1ECP	Single bit error correction permission 0: Single bit error is corrected 1: Single bit error is not corrected
4	EC2EDIC	Double bit error detection interrupt INTVRAMnDED control 0: INTVRAMnDED is not generated 1: INTVRAMnDED is generated upon double bit error detection
3	EC1EDIC	Single bit error detection interrupt INTVRAMnSED control 0: INTVRAMnSED is not generated 1: INTVRAMnSED is generated upon single bit error detection
2	ECER2F	Double bit error detection flag 0: no double bit error detected 1: double bit error detected Writing to this bit has no affect. This bit is cleared under following conditions: • writing VRAMnCTL.ECER2C = 1 • ECC pass through mode (VRAMnCTL.ECTHM = 1) • ECC error judgment disabled (VRAMnCTL.ECERVF = 0)

Table 54.11 VRAMnCTL register contents (3/3)

Bit position	Bit name	Function
1	ECER1F	Single bit error detection flag 0: no single bit error detected 1: single bit error detected and corrected Writing to this bit has no affect. This bit is cleared under following conditions: • writing VRAMnCTL.ECER1C = 1 • ECC pass through mode (VRAMnCTL.ECTHM = 1) • ECC error judgment disabled (VRAMnCTL.ECERVF = 0)
0	ECEMF	ECC error indication of current VRAM data read 0: no ECC error detected at current read 1: ECC error detected at current read This bit is updated by every RAM output data. This bit is cleared under following conditions: • current VRAM data read without ECC error • ECC pass through mode (VRAMnCTL.ECTHM = 1) • ECC error judgment disabled (VRAMnCTL.ECERVF = 0)

54.6.2.2 VRAMnEADm – ECC error address register m (m = 0 to 7)

These registers store the read address, when an ECC error was detected.

Access: This register can be accessed in 32-bit units.

Address: <VRAMn_base> + 14_H + m x 04_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEAD[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEAD[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 54.12 VRAMnEADm register contents

Bit position	Bit name	Function
31 to 0	ECEAD[31:0]	Address of read access, when an ECC error was detected. The type of error, i.e. single or double bit error, is indicated by VRAMnCTL.ECDEDm and VRAMnCTL.ECSEDm. ECEAD[31:0] is the address of the erroneous 64-bit data unit. Base address is not included. The complete read address is calculated by use of this formula: read address = VRAMn ECC mode start address + ECEAD[31:0] * 8 Under certain conditions the address of the VRAMnEAD7 register is overwritten upon detection of an ECC error. Refer to Section (3), Error detection stages overflow for details. VRAMnEADm is set to its default value by clearing the error detection flag ECER1C or ECER2C in the ECC control register VRAMnCTL. For VRAMn ECC mode start address of each product, refer to **Section 54.5, Video RAM and SDRAM memory map**.

54.6.2.3 VRAMnTMC – ECC test mode control register

The VRAMnTMC register switches to and controls the test mode.

This register can be used when VRAM is not accessed.

Access: This register can be accessed in 16-bit units.

Address: <VRAMn_base> + 04_H

Initial value: 0000 0000_H

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ETMA[1:0]		0	0	0	0	0	0	EC TMCE	0	0	EC TRRS	EC REOS	EC ENS	EC DCS	EC REIS
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R	R	R	R	R	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 54.13 VRAMnTMC register contents (1/2)

Bit position	Bit name	Function
15, 14	ETMA[1:0]	Access control bits 1 and 0 to ECC test mode bit These two bits specify whether updating the ECTMCE bit (bit 7) is disabled or enabled. The value written to these bits is not retained. When these bits are read, the read value is always 0. When these bits are 10_B, writing to bit 7 is enabled.
13 to 8	Reserved	When read, the value after reset is read. When written, write the value after reset.
7	ECTMCE	ECC test mode enable bit This bit specifies whether to enable access to the test registers and test control bits. When writing to this bit, 10_B should be written to ETMA[1:0] at the same time. 0: Access to the test mode registers and bits is disabled. 1: Access to the test mode registers and bits is enabled. Test registers: VRAMnTED, VRAMnTRC, VRAMnHORD, VRAMnECRD, VRAMnERDB Register test control bit: ECTRRS, ECREOS, ECENS, ECDCS, ECREIS
6 to 5	Reserved	When read, the value after reset is read. When written, write the value after reset.
4	ECTRRS	ECC RAM read test mode selection bit Select reading destination when reading VRAMnTED register and reading destination when reading VRAMnERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: The read value of the VRAMnTED register is the write value of the VRAMnTED register. The read value of the VRAMnERDB register is the write value of the VRAMnERDB register. 1: The read value of the VRAMnTED register can read RAM data. The read value of the VRAMnERDB register is the ECC data to be written to RAM.
3	ECREOS	ECC redundant bit output data selection bit This bit is used to identify and select the ECC data to be stored in RAM as ECC data generated for write data or the value of the VRAMnERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Store ECC data generated for write data to RAM. 1: Store the value of VRAMnERDB register to RAM.

Table 54.13 VRAMnTMC register contents (2/2)

Bit position	Bit name	Function
2	ECENS	ECC encoder input selection bit This bit is used to identify the targeted data for ECC data generation as write data to RAM or the value of the VRAMnTED register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate ECC data from the write data to RAM. 1: Generate ECC data from the value of the VRAMnTED register.
1	ECDCS	ECC decoder input selection bit This bit is used to identify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the VRAMnTED register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate syndrome code from RAM data and detect errors. 1: Generate syndrome code from the value of the VRAMnTED register and detect errors.
0	ECREIS	ECC redundant bit input data selection bit This bit is used to identify and select the targeted data for syndrome code generation and error detection as RAM data or the value of the VRAMnERDB register. Writing to this bit is enabled only when ECTMCE = 1 (can be set simultaneously). This bit is cleared when ECTMCE = 0 (cleared synchronously). 0: Generate syndrome code from ECC data stored in RAM and detect errors. 1: Generate syndrome code from the value of the VRAMnERDB register and detect errors.

54.6.2.4 VRAMnTED1 – ECC encode/decode input/output substitution register 1

In ECC test mode, this register handles upper 32 bit data of ECC test data.
The value of the register can be used to generate ECC data or syndrome code.
It is accessible when ECC test mode is enabled (VRAMnTMC.ECTMCE = 1)
When VRAMnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when VRAM is not accessed.

Access: This register can be accessed in 32-bit units.

Address: <VRAMn_base> + 08_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[63:48]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[47:32]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 54.14 VRAMnTED1 register contents

Bit position	Bit name	Function
31 to 0	ECEDB[63:32]	When VRAMnTMC.ECENS = 1, the value of this register is used to generate ECC data and the value this register to RAM is stored to RAM. When VRAMnTMC.ECDCS = 1, the value of this register is used to generate syndrome code. In addition, when VRAMnTMC.ECTRRS = 1, RAM data [63:32], instead of written data, is read for the value of this register.

54.6.2.5 VRAMnTED2 – ECC encode/decode input/output substitution register 2

In ECC test mode, this register handles lower 32 bit data of ECC test data.
The value of the register can be used to generate ECC data or syndrome code.
It is accessible when ECC test mode is enabled (VRAMnTMC.ECTMCE = 1).
When VRAMnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when VRAM is not accessed.

Access: This register can be accessed in 32-bit units.

Address: <VRAMn_base> + $0C_H$

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ECEDB[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
							R									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECEDB[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 54.15 VRAMnTED2 register contents

Bit position	Bit name	Function
31 to 0	ECEDB[31:0]	When VRAMnTMC.ECENS = 1, the value of this register is used to generate ECC data and the value this register to RAM is stored to RAM. When VRAMnTMC.ECDCS = 1, the value of this register is used to generate syndrome code. In addition, when VRAMnTMC.ECTRRS = 1, RAM data [31:0], instead of written data, is read for the value of this register.

54.6.2.6 VRAMnTRC – ECC test redundant bit data control register

In ECC test mode, this test register, for ECC data, consists of three 8-bit registers, VRAMnHORD, VRAMnECRD, and VRAMnERDB.
It is accessible when ECC test mode is enabled (VRAMnTMC.ECTMCE = 1).
When VRAMnTMC.ECTMCE = 0, writing to this register is ignored and 0000 0000_H is read.

This register can be used when VRAM is not accessed.

Access: This register can be accessed in 32-bit units.

Address: <VRAMn_base> + 10_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	HORD[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECRD[7:0]								ERDB[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 54.16 VRAMnTRC register contents (1/2)

Bit position	Bit name	Function
31 to 24	Reserved	Writing to this register is ignored. When ECC test mode is enabled (VRAMnTMC.ECTMCE = 1), the read values of these bits are undefined. When ECC test mode is disabled (VRAMnTMC.ECTMCE = 0), 00_H is read.
23 to 16	HORD[7:0]	In ECC test mode, this register is used to store ECC data for read RAM data. Writing to this register is ignored. When ECC test mode is enabled (VRAMnTMC.ECTMCE = 1), this register can be accessed. When ECC test mode is disabled (VRAMnTMC.ECTMCE = 0), 00_H is read. HORD[7:0] ECC code for read RAM data is stored as needed. When VRAMnTMC.ECTRRS = 1 and VRAMnTED register is read, ECC code is stored.
15 to 8	ECRD[7:0]	In ECC test mode, this is a read-only register for storing ECC data generated for write RAM data. Writing to this register is ignored. When ECC test mode is enabled (VRAMnTMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (VRAMnTMC.ECTMCE = 0), 00_H is read. ECRD[7:0] These bits can read ECC data generated at the time of RAM data writing, and can read ECC data for the data written to the VRAMnTED register when VRAMnTMC.ECENS = 1.

Table 54.16 VRAMnTRC register contents (2/2)

Bit position	Bit name	Function
7 to 0	ERDB[7:0]	In ECC test mode, this register handles ECC data. The value of this register can be used as ECC data generated at the time of RAM writing or ECC data read at the time of RAM data reading. When ECC test mode is enabled (VRAMnTMC.ECTMCE = 1), this register is accessible. When ECC test mode is disabled (VRAMnTMC.ECTMCE = 0), 00_H is read. ERDB[7:0] When VRAMnTMC.ECREOS = 1, these bits store the value of this register as ECC data to RAM. When VRAMnTMC.ECREIS = 1, the value of this register is read as ECC data read from RAM. When VRAMnTMC.ECTRRS = 1, ECC data to be stored in RAM is read instead of the write data for the read value of this register.

Section 55 Boundary Scan

55.1 Outline

Boundary Scan is a test method defined by IEEE Std 1149.1.

It is used for testing interconnections between devices mounted on printed circuits boards.

RH850/D1L/D1M complies with the IEEE Std 1149.1-2001.

NOTE

Boundary Scan generic issues (e.g. explanation of each instruction, state transition diagram of TAP controller, etc.) are not described in this document.

55.2 JTAG interface

Communication with a Boundary Scan host test system is executed via the JTAG interface, as described in the following table:

Table 55.1 JTAG interface

Port	Signal name	Recommendation if unused
JP0_0	DCUTDI	Open (if $\overline{\text{DCUTRST}}$ at low level)
JP0_1	DCUTDO	Open
JP0_2	DCUTCK	Open
JP0_3	DCUTMS	Open
JP0_4	$\overline{\text{DCUTRST}}$	Connect to VSS via an appropriate resistor

55.3 Entering Boundary Scan mode

Boundary Scan mode is entered by setting following pins at reset release:

- FLMD0 = VCC
- FLMD1 (P0_1) = VCC
- MODE0 (P0_0) = VSS
- MODE1 (P0_2) = VCC

Refer also to Section 6, Operating Modes for details.

55.4 Boundary scan features

The table below shows the IEEE1149.1-2001 instructions and instruction codes for RH850/D1L/D1M.

Table 55.2 Boundary scan instructions

Instruction code								Description
0	0	0	0	0	0	0	0	JTAG EXTEST
0	1	0	0	0	0	0	0	JTAG SAMPLE/PRELOAD
0	1	0	1	0	1	0	1	JTAG IDCODE (initial value)
1	1	1	1	1	1	1	1	JTAG BYPASS
Other than above								Reserved

55.5 Boundary Scan applicable pins

NOTE

The ports of the Boundary Scan applicable pins don't need to be configured by use of any port configuration register during Boundary Scan.

Boundary Scan (EXTEST) is applicable to all pins except pins indicated in the following table.

Table 55.3 Boundary Scan inapplicable pins

Type	Pin name
Reset	$\overline{\text{RESET}}$
MODE	FLMD0
JTAG interface	JP0_0 to JP0-4
External power supply control	PWRGD, PWRCTL
Voltage references	A0VREF, ZPDVREF
Clock inputs/outputs	X1, X2, XT1, XT2
Power supply	all power supply pins
Ground	all ground pins

Following pins are shared with analog buffer, differential buffer, etc. So, boundary scan is applied only to general purpose I/O part.

Table 55.4 Boundary Scan (only general purpose I/O part) applicable pins

Type	Pin name
ADCE0 inputs	P10_0 to P10_12, P11_0 to P11_7
MIPI video input I/F	P40_0 to P40_5
RSDS video output	P44_0 to P44_11, P45_0 to P45_13

55.6 Device ID register (DID)

The RH850/D1L/D1M specific ID code is provide by the device identification register DID. The value of DID is read with IDCODE instruction and cannot be accessed by CPU.

55.6.1 DID - Boundary scan ID register

This register stores the boundary scan ID.

Access: This register can not be read by the CPU.

Address: –

Initial value:
- D1L1: 0822 $C447_H$
- D1L2(H): 0822 $B447_H$
- D1M1(H): 0822 $A447_H$
- D1M1-V2: 1835 1447H
- D1M1A: 1835 0447_H
- D1M2(H): 0822 9447_H

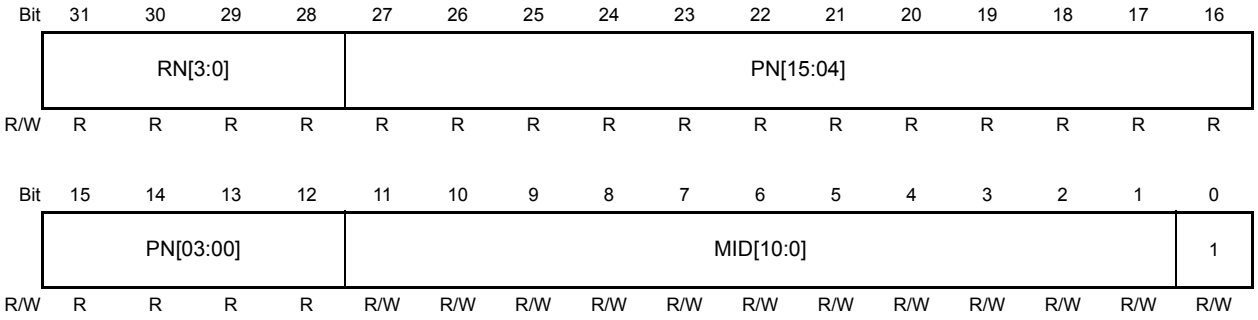


Table 55.5 DID register contents

Bit position	Bit name	Function
31 to 28	RN[3:0]	Revision number
27 to 12	PN[15:00]	Product number: D1L1: $822C_H$ D1L2(H): $822B_H$ D1M1(H): $822A_H$ D1M1-V2: 8351_H D1M1A: 8350_H D1M2(H): 8229_H
11 to 1	MID[10:0]	Manufacturer ID: 447_H = {MID[10:0], 1}

Section 56 HyperBus Controller (HYPB)

This section contains a generic description of the HyperBus Controller.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

56.1 Overview of RH850/D1L/D1M HyperBus Controller

56.1.1 Units

This microcontroller has the following number of units of the HyperBus Controller.

Table 56.1 Units

HyperBus Controller	D1L1	D1L2(H)	D1M1	D1M1H	D1M1-V2 D1M1A	D1M2(H)
Units	–	–	–	–	1	–
Names	–	–	–	–	HYPB0	–

56.1.2 Indices

Following indices are used in this section:

Table 56.2 Indices

Index	Meaning
n	The individual HYPB units are generically indicated by the index “n”.

56.1.3 Register addresses

All HyperBus Controller register addresses are given as address offsets from the individual base addresses <HYPBn_base>.
The <HYPBn_base> addresses of each HYPBn are listed in the following table:

Table 56.3 Register base addresses <HYPBn_base>

HYPBn unit	<HYPBn_base> address
HYPB0	F2FF 8000_H

56.1.4 Clock supply

All HyperBus Controller provide two clock inputs.

Table 56.4 Clock supply

HYPBn unit	HYPBn clock	Connected to
HYPB0	RPC_CLK	Clock Controller CKSC_ISFMAD_CTL clock divider output
	RPC_CLK2	Clock Controller CKSC_ISFMAD_CTL clock divider output divided by 2

56.1.5 Reset sources

The HyperBus Controller and their registers are initialized by the following reset signal:

Table 56.5 Reset sources

HYPBn unit	Reset signal
HYPBn	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

56.1.6 I/O signals

The following table shows the I/O signals of the HyperBus Controller.

Table 56.6 HYPBn I/O signals

HYPBn signals	Function	Connected to
HYPB0		
MCLK	Clock output	Port MCK
MCS	Chip select	Port MCS
MDQ[7:0]	Data[7:0]	Port MDQI[7:0] / MDQO[7:0]
MDQS	Read data strobe / write data mask	Port MDQS

CAUTION

The HyperBus and the OctaBus interface share the same ports P21_[9:0] and P22_10. The HyperBus or OctaBus selection via the bus switch control bit XCRAMCFG0.HYPBSEL must be done before the XC0 cross-connect is released from reset, i.e. before MRSTC.XC0RES = 1 is set.

Refer to Section 14.4, Bus Switch for external memory interfaces (D1M1-V2, D1M1A only) for details.

56.2 Overview

The HyperBus controller enables direct connection of the HyperRAM or HyperFlash memory to this LSI.

56.2.1 Features

This module allows the HyperRAM or HyperFlash memory connected to the MCU to be accessed by reading the external address space, or using manual mode to transmit and receive data.

HyperBus Interface

- One HyperFlash or one HyperRAM can be connected.
- Maximum flash/RAM size is 16 Mbyte.
- The data bus size is fixed to 8 bits.
- HyperRAM can be used with fixed latency (variable latency is not supported)
- 3.3 V Interface voltage level (1.8 V I/O connection is not supported)
- The burst type support "linear mode" only.

NOTE

Although D1M1A supports flash/RAM sizes up to 16 Mbytes, it does not support access to dual-die HyperFlash/RAM devices.

External Address Space Read/Write Mode

- A read access from an MCU on-chip module to the HyperBus memory area is automatically converted into a read command to the device and the read data is returned to the on-chip module.
- Efficient data reception is possible by utilizing the on-chip read cache memory (64-bit line × 32 entries) during burst read operation.
- A write access from an MCU on-chip module to the HyperRAM memory area is automatically converted into a write command to the device and the write data is stored to the device.

Manual Mode

- Desired command and read accesses to the HyperRAM or HyperFlash memory are possible. Manual mode does not support write access to HyperRAM.
- Efficient data writing is possible by using write buffer for HyperFlash programming.

NOTE

If the HyperBus Controller is operated in Manual Mode, access of the HyperBus memory area by other modules is not possible.

56.3 Block Diagram

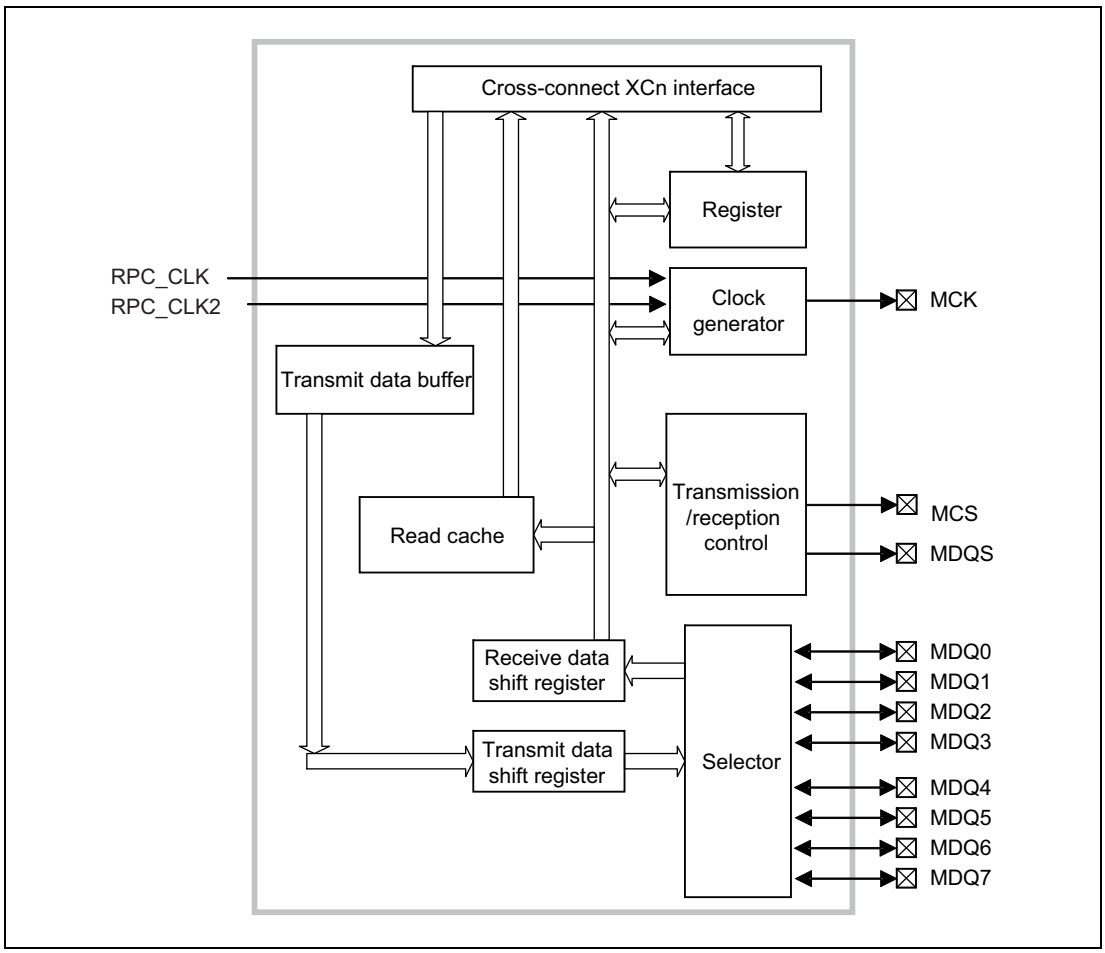


Figure 56.1 HyperBus Controller block diagram

56.4 Register Configurations

The following table shows the register configuration of the HyperBus Controller.

Table 56.7 Register configuration

Register name	Symbol	Address
Common control register	CMNCR	<HYPBn_base> + 00_H
SSL delay register	SSLDR	<HYPBn_base> + 04_H
Data read control register	DRCR	<HYPBn_base> + $0C_H$
Data read command setting register	DRCMR	<HYPBn_base> + 10_H
Data read option setting register	DROPR	<HYPBn_base> + 18_H
Data read enable setting register	DRENR	<HYPBn_base> + $1C_H$
Manual mode control register	SMCR	<HYPBn_base> + 20_H
Manual mode command setting register	SMCMR	<HYPBn_base> + 24_H
Manual mode address setting register	SMADR	<HYPBn_base> + 28_H
Manual mode option setting register	SMOPR	<HYPBn_base> + $2C_H$
Manual mode enable setting register	SMENR	<HYPBn_base> + 30_H
Manual mode read data register 0	SMRDR0	<HYPBn_base> + 38_H
Manual mode read data register 1	SMRDR1	<HYPBn_base> + $3C_H$
Manual mode write data register 0	SMWDR0	<HYPBn_base> + 40_H
Manual mode write data register 1	SMWDR1	<HYPBn_base> + 44_H
Common status register	CMNSR	<HYPBn_base> + 48_H
Data read dummy cycle setting register	DRDMCR	<HYPBn_base> + 58_H
Manual mode dummy cycle setting register	SMDMCR	<HYPBn_base> + 60_H
Manual mode DDR enable register	SMDRENR	<HYPBn_base> + 64_H
PHY control register	PHYCNT	<HYPBn_base> + $7C_H$

<HYPBn_base>

The base addresses <HYPBn_base> of the HYPBn is defined in the first subsection of this section under the key word “Register addresses”.

NOTES

1. In the header files the names of the above registers are defined in the following format: <ModuleName> + <Symbol>.
   <ModuleName> and <Symbol> are defined in the above table.
2. Do not write to any addresses other than listed above. Operations cannot be guaranteed if writing is attempted. Values read from addresses other than those listed above are undefined.

56.5 Register Descriptions

The registers of the HyperBus are assigned to and located in the address space of the internal bus.

[Legend for Register Descriptions]

Initial value: Register value after a reset.

—: Undefined value

R/W: Readable/writable. The written value can be read.

R: Read-only. The write value should always be 0.

—/WB: Write-only. The read value is undefined.

56.5.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the HyperBus controller. The settings of this register are reflected both in external address space read mode and manual operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MD	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 56.8 CMNCR register contents

Bit Position	Bit Name	Function
31	MD	Operating Mode Switch 0: External address space read mode 1: Manual mode
30 to 0	Reserved	Reading these bits returns always the initial value. The write value should always be the initial value.

56.5.2 SSL Delay Register (SSLDR)

SSLDR is a 32-bit register that adjusts the timing between the MCS signal and the MCK signal.

The settings of this register are reflected both in external address space read mode and manual operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	SPNDL[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	SLNDL[2:0]			—	—	—	—	—	SCKDL[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 56.9 SSLDR register contents

Bit Position	Bit Name	Function
31 to 19	Reserved	These bits are always read as 0. The write value should always be 0.
18 to 16	SPNDL[2:0]	Next Access Delay Sets the period from transfer end to next transfer start (next access). 000: 1 cycle of MCK 001: 2 cycles of MCK 010: 3 cycles of MCK 011: 4 cycles of MCK 100: 5 cycles of MCK 101: 6 cycles of MCK 110: 7 cycles of MCK 111: 8 cycles of MCK
15 to 11	Reserved	These bits are always read as 0. The write value should always be 0.
10 to 8	SLNDL[2:0]	MCS Negation Delay Sets the period from the time the last MCK edge is sent of a transfer to MCS pin negation (MCS negation delay). 000: 1 cycle of MCK 001: 2 cycles of MCK 010: 3 cycles of MCK 011: 4 cycles of MCK 100: 5 cycles of MCK 101: 6 cycles of MCK 110: 7 cycles of MCK 111: 8 cycles of MCK
7 to 3	Reserved	These bits are always read as 0. The write value should always be 0.
2 to 0	SCKDL[2:0]	Clock Delay Sets the period from MCS pin assertion to MCK oscillation (clock delay). 000: 1.5 cycles of MCK 001: 2.5 cycles of MCK 010: 3.5 cycles of MCK 011: 4.5 cycles of MCK 100: 5.5 cycles of MCK 101: 6.5 cycles of MCK 110: 7.5 cycles of MCK 111: 8.5 cycles of MCK

56.5.3 Data Read Control Register (DRCR)

DRCR is a 32-bit register that sets the operation in the external address space read mode.

For other than SSLN bit, the settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	SSLN	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
R/W	R	R	R	R	R	R	R	W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	RCF	BST[3:0]				—	—	—	—	SSLE
Initial value	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	W	R/W	R/W	R/W	R/W	R	R	R	R	R/W

Table 56.10 DRCR register contents

Bit Position	Bit Name	Function
31 to 25	Reserved	These bits are always read as 0. The write value should always be 0.
24	SSLN	MCS Negation This bit is always read as 0. To start next access after MCS negation using this bit, read SSLF in CMNSR = 0 to confirm that the MCS has been negated.
23 to 10	Reserved	Reading these bits returns always the initial value. The write value should always be the initial value.
9	RCF	Read Cache Flush When 1 is written to this bit, all the entries in the read cache are cleared. This bit is always read as 0. After flushing the read cache by writing 1 to the RCF bit, read the DRCR before proceeding to read from the external address space.
8 to 5	BST[3:0]	Burst Control 1111:Enable burst read/write HyperBus read/write transfer length according to Cross-connect XCn access transfer length. Other: prohibit
4 to 1	Reserved	These bits are always read as 0. The write value should always be 0.
0	SSLE	MCS Negation Setting Sets the conditions for MCS negation during read burst. MCS is negated for each access during normal read. 0: MCS is negated after transfer of data set in burst length. 1: MCS is negated when the accessed address is not continuous with the previously transferred address. **Note:** For HyperFlash/HyperRAM SSLE must be set to 0.

56.5.4 Data Read Command Setting Register (DRCMR)

DRCMR is a 32-bit register that sets the commands issued in external address space read mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CMD[7:0]							
Initial value	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OCMD[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 56.11 DRCMR register contents

Bit Position	Bit Name	Function
31 to 24	Reserved	These bits are always read as 0. The write value should always be 0.
23 to 16	CMD[7:0]	Command Sets the command.
15 to 8	Reserved	These bits are always read as 0. The write value should always be 0.
7 to 0	OCMD[7:0]	Optional Command Sets the optional command.

56.5.5 Data Read Option Setting Register (DROPR)

DROPR is a 32-bit register that sets the option data in external address space read mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OPD3[7:0]								OPD2[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPD1[7:0]								OPD0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 56.12 DROPR register contents

Bit Position	Bit Name	Function
31 to 24	OPD3[7:0]	Option Data 3 Sets the option data 3.
23 to 16	OPD2[7:0]	Option Data 2 Sets the option data 2.
15 to 8	OPD1[7:0]	Option Data 1 Sets the option data 1.
7 to 0	OPD0[7:0]	Option Data 0 Sets the option data 0.

NOTE

OPD3[7:0], OPD2[7:0], OPD1[7:0], and OPD0[7:0] are output in this order.

56.5.6 Data Read Enable Setting Register (DRENR)

DRENR is a 32-bit register that sets the read data in external address space read mode and enables output of data other than read data.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DME	CDE	—	OCDE	—	—	—	—	OPDE[3:0]				—	—	—	—
Initial value	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R

Table 56.13 DRENR register contents

Bit Position	Bit Name	Function
31 to 16	Reserved	Reading these bits returns always the initial value. The write value should always be the initial value.
15	DME	Dummy Cycle Enable Enables or disables insertion of a dummy cycle. The dummy cycle is inserted before the read data. Setting of a transfer starting with a dummy cycle is prohibited. 0: Insertion of a dummy cycle is disabled. 1: Insertion of a dummy cycle is enabled.
14	CDE	Command Enable Enables or disables output of commands. 0: Output disabled 1: Output enabled
13	Reserved	This bit is always read as 0. The write value should always be 0.
12	OCDE	Optional Command Enable Sets the optional command to be output. 0: Output disabled 1: Optional command output enabled
11 to 8	Reserved	Reading these bits returns always the initial value. The write value should always be the initial value.
7 to 4	OPDE[3:0]	Option Data Enable Sets the option data to be output. Be sure to use the following setting. Otherwise, the operation is not guaranteed. 0000: Output disabled 1000: OPD3[7:0] is output. 1100: OPD3[7:0] and OPD2[7:0] are output. 1110: OPD3[7:0], OPD2[7:0], and OPD1[7:0] are output. 1111: OPD3[7:0], OPD2[7:0], OPD1[7:0], and OPD0[7:0] are output. All others: Setting prohibited
3 to 0	Reserved	These bits are always read as 0. The write value should always be 0.

56.5.7 Manual Mode Control Register (SMCR)

SMCR is a 32-bit register that sets the operation in manual mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	SSLKP	—	—	—	—	—	SPIRE	SPIWE	SPIE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R/W	R/W	W

Table 56.14 SMCR register contents

Bit Position	Bit Name	Function
31 to 9	Reserved	These bits are always read as 0. The write value should always be 0.
8	SSLKP	MCS Signal Level Determines the MCS status after the end of transfer. 0: MCS signal is negated at the end of transfer. 1: MCS signal level is maintained from the end of transfer to the start of next access.
7 to 3	Reserved	These bits are always read as 0. The write value should always be 0.
2	SPIRE	Data Read Enable Enables reading in manual mode. 0: Data reading disabled 1: Data reading enabled The SPIRE and SPIWE bits should not be set to 1 at the same time.
1	SPIWE	Data Write Enable Enables writing in manual mode. 0: Data writing disabled 1: Data writing enabled The SPIRE and SPIWE bits should not be set to 1 at the same time.
0	SPIE	SPI Data Transfer Enable Data is transferred by setting this bit to 1. This bit is enabled only when the TEND bit in CMNSR is set to 1. The operation cannot be guaranteed when this bit is set to 1 with the TEND bit set to 0. This bit is always read as 0. **Note:** When the MCS pin is negated, the command, optional command, address, and option data that are output enabled are output even if the SPIRE and SPIWE bits are set to 0.

56.5.8 Manual Mode Command Setting Register (SMCMR)

SMCMR is a 32-bit register that sets the commands issued in manual mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CMD[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	OCMD[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 56.15 SMCMR register contents

Bit Position	Bit Name	Function
31 to 24	Reserved	These bits are always read as 0. The write value should always be 0.
23 to 16	CMD[7:0]	Command Sets the command.
15 to 8	Reserved	These bits are always read as 0. The write value should always be 0.
7 to 0	OCMD[7:0]	Optional Command Sets the optional command.

56.5.9 Manual Mode Address Setting Register (SMADR)

SMADR is a 32-bit register that sets the addresses in manual mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADR[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 56.16 SMADR register contents

Bit Position	Bit Name	Function
31 to 0	ADR[31:0]	Address Sets the address.

56.5.10 Manual Mode Option Setting Register (SMOPR)

SMOPR is a 32-bit register that sets the option data in manual mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	OPD3[7:0]								OPD2[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OPD1[7:0]								OPD0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 56.17 SMOPR register contents

Bit Position	Bit Name	Function
31 to 24	OPD3[7:0]	Option Data 3 Sets the option data 3.
23 to 16	OPD2[7:0]	Option Data 2 Sets the option data 2.
15 to 8	OPD1[7:0]	Option Data 1 Sets the option data 1.
7 to 0	OPD0[7:0]	Option Data 0 Sets the option data 0.

NOTE

OPD3[7:0], OPD2[7:0], OPD1[7:0], and OPD0[7:0] are output in this order.

56.5.11 Manual Mode Enable Setting Register (SMENR)

SMENR is a 32-bit register that sets the bit size of the command, optional command, address, option data, and transfer data in manual mode and enables their output. SMENR also enables dummy cycle insertion. Disabling all of the command, optional command, address, option data, dummy cycle, and transfer data is prohibited. At least one of them except dummy cycle must be enabled.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CDB[1:0]		OCDB[1:0]		—	—	ADB[1:0]		—	—	OPDB[1:0]		—	—	SPIDB[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DME	CDE	—	OCDE	ADE[3:0]				OPDE[3:0]				SPIDE[3:0]			
Initial value	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 56.18 SMENR register contents (1/2)

Bit Position	Bit Name	Function
31, 30	CDB[1:0]	Command Bit Size Sets the command size in bit units. 00_B: reserved 01_B: reserved 10_B: 8 bits 11_B: Setting prohibited
29, 28	OCDB[1:0]	Optional Command Bit Size Sets the optional command size in bit units. 00_B: reserved 01_B: reserved 10_B: 8 bits 11_B: Setting prohibited
27, 26	Reserved	This bit is always read as 0. The write value should always be 0.
25, 24	ADB[1:0]	Address Bit Size Sets the address size in bit units. 00_B: reserved 01_B: reserved 10_B: 8 bits 11_B: Setting prohibited
23, 22	Reserved	This bit is always read as 0. The write value should always be 0.
21, 20	OPDB[1:0]	Option Data Bit Size Sets the option data size in bit units. 00_B: reserved 01_B: reserved 10_B: 8 bits 11_B: Setting prohibited
19, 18	Reserved	This bit is always read as 0. The write value should always be 0.
17, 16	SPIDB[1:0]	Transfer Data Bit Size Sets the transfer data size in bit units. 00_B: reserved 01_B: reserved 10_B: 8 bits 11_B: Setting prohibited

Table 56.18 SMENR register contents (2/2)

Bit Position	Bit Name	Function
15	DME	Dummy Cycle Enable Enables or disables insertion of the dummy cycle before the read data. **Note:** Dummy cycle insertion is prohibited for write in manual mode, including the case in which a transfer ends with a dummy cycle. A setting is prohibited for a transfer starting with a dummy cycle. 0: Dummy cycle insertion disabled 1: Dummy cycle insertion enabled
14	CDE	Command Enable Sets the command to be output. 0: Output disabled 1: Output enabled
13	Reserved	This bit is always read as 0. The write value should always be 0.
12	OCDE	Optional Command Enable Sets the optional command to be output. 0: Optional command output disabled 1: Optional command output enabled
11 to 8	ADE[3:0]	Address Enable Sets the address to be output. The following settings must be used. Otherwise, the operation cannot be guaranteed. 0000_B: Output disabled 0100_B: ADR[31:0] is output All others: Setting prohibited
7 to 4	OPDE[3:0]	Option Data Enable Sets the option data to be output. The following settings must be used. Otherwise, the operation cannot be guaranteed. 0000_B: Output disabled 1000_B: OPD3[7:0] is output. 1100_B: OPD3[7:0] and OPD2[7:0] are output. 1110_B: OPD3[7:0], OPD2[7:0], and OPD1[7:0] are output. 1111_B: OPD3[7:0], OPD2[7:0], OPD1[7:0], and OPD0[7:0] are output. All others: Setting prohibited
3 to 0	SPIDE[3:0]	Transfer Data Enable Sets valid transfer data. The following settings must be used. Otherwise, the operation cannot be guaranteed. 0000_B: Not transferred 1000_B: 16 bits transferred (enables data at addresses 0 to 1 of the manual mode read/write data registers 0) 1100_B: 32 bits transferred (enables data at addresses 0 to 3 of the manual mode read/write data registers 0) 1111_B: 64 bits transferred (enables data at addresses 0 to 3 of the manual mode read/write data registers 0 and data at addresses 0 to 3 of the manual mode read/write data registers 1) All others: Setting prohibited

56.5.12 Manual Mode Read Data Register 0 (SMRDR0)

SMRDR1 is a 32-bit register that holds the read data in manual mode.

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the manual mode enable setting register (SMENR). Be sure to access from LSB.

The data should be read when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

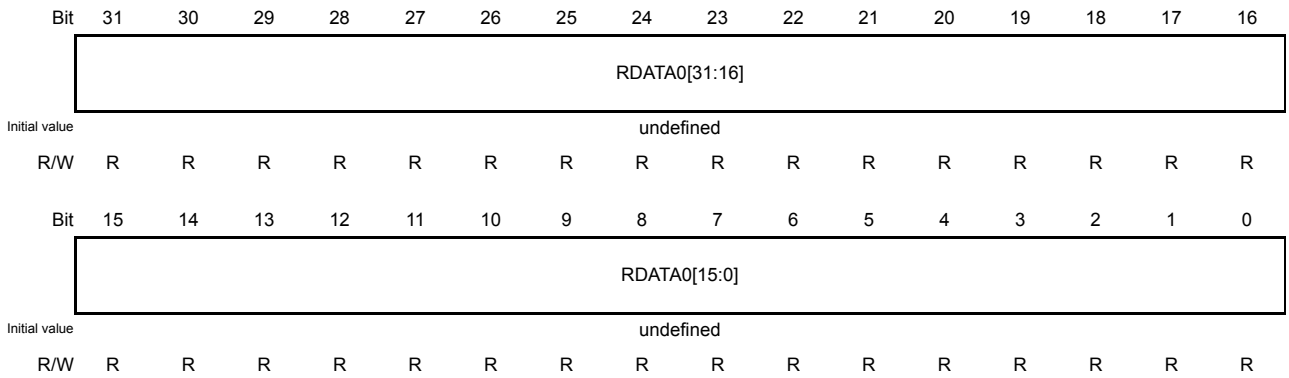


Table 56.19 SMRDR0 register contents

Bit Position	Bit Name	Function
31 to 0	RDATA0[31:0]	Read Data Holds the data read in manual mode. Read data[63:32] is stored to this register.

NOTE

The contents of this register and SMRDR1 are modified upon completion of reception in manual mode. Be sure to read data when reception in manual mode is completed.

56.5.13 Manual Mode Read Data Register 1 (SMRDR1)

SMRDR1 is a 32-bit register that holds the read data in manual mode.

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the manual mode enable setting register (SMENR). Be sure to access from LSB.

The data should be read when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	RDATA1[31:16]															
Initial value	undefined															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RDATA1[15:0]															
Initial value	undefined															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 56.20 SMRDR1 register contents

Bit Position	Bit Name	Function
31 to 0	RDATA1[31:0]	Read Data Holds the data read in manual mode. Read data[31:0] is stored to this register.

56.5.14 Manual Mode Write Data Register 0 (SMWDR0)

SMWDR0 is a 32-bit register that sets the write data in manual mode.

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the manual mode enable setting register (SMENR). Be sure to access from address 0.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDATA0[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDATA0[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 56.21 SMRWR0 register contents

Bit Position	Bit Name	Function
31 to 0	WDATA0[31:0]	Write Data Holds the data to be written in manual mode. This register is assigned to Write data[63:32].

56.5.15 Manual Mode Write Data Register 1 (SMWDR1)

SMWDR1 is a 32-bit register that sets the write data in manual mode.

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the manual mode enable setting register (SMENR). Be sure to access from address 0.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	WDATA1[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	WDATA1[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 56.22 SMRWDR1 register contents

Bit Position	Bit Name	Function
31 to 0	WDATA1[31:0]	Write Data Holds the data to be written in manual mode. This register is assigned to Write data[31:0].

56.5.16 Common Status Register (CMNSR)

CMNSR is a 32-bit register that holds flags indicating the operating state.

The settings of this register are reflected both in external address space read mode and manual operating mode.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSLF	TEND
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 56.23 CMNSR register contents

Bit Position	Bit Name	Function
31 to 2	Reserved	These bits are always read as 0. The write value should always be 0.
1	SSLF	MCS Pin State Monitor 0: MCS pin is negated. 1: MCS pin is asserted.
2	TEND	Transfer End Flag Indicates whether the data transfer has ended. 0: Indicates that data transfer is in progress. 1: Indicates that data transfer has ended.

56.5.17 Data Read Dummy Cycle Setting Register (DRDMCR)

DRDMCR is a 32-bit register that sets the size and number of dummy cycles to be inserted in external address space read mode.

The settings of this register are enabled when the DME bit in the data read enable setting register (DRENR) is 1.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	DMCYC[2:0]			—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R

Table 56.24 DRDMCR register contents

Bit Position	Bit Name	Function
31 to 4	Reserved	These bits are always read as 0. The write value should always be 0.
3 to 1	DMCYC[2:0]	Number of Dummy Cycles Setting Sets the number of dummy cycles to be inserted when the DME bit in the data read enable setting register (DRENR) is 1. 011: 8 cycles 100: 10 cycles 101: 12 cycles 110: 14 cycles 111: 16 cycles All others: Setting prohibited
0	Reserved	These bits are always read as 0. The write value should always be 0.

56.5.18 Manual Mode Dummy Cycle Setting Register (SMDMCR)

SMDMCR is a 32-bit register that sets the size and number of dummy cycles to be inserted in manual mode.

The settings of this register are enabled when the DME bit in the manual mode enable setting register (SMENR) is 1.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	DMCYC[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W

Table 56.25 SMDMCR register contents

Bit Position	Bit Name	Function
31 to 4	Reserved	These bits are always read as 0. The write value should always be 0.
3 to 0	DMCYC[3:0]	Number of Dummy Cycles Sets the number of dummy cycles to be inserted when the DME bit in the manual mode enable setting register (SMENR) is 1. 0000: 1 cycle 0001: 2 cycles 0010: 3 cycles .... 1110: 15 cycles 1111: 16 cycles All others: Setting prohibited

56.5.19 Manual Mode DDR Enable Register (SMDRENR)

SMDRENR is a 32-bit register that specifies the SDR or DDR transfer of the address, option data, and data for transfer in manual mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	HYPE[2:0]			—	—	—	ADDRE	—	—	—	OPDRE	—	—	—	SPIDRE
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R	R	R	R/W	R	R	R	R/W	R	R	R	R/W

Table 56.26 SMDRENR register contents

Bit Position	Bit Name	Function
31 to 15	Reserved	These bits are always read as 0. The write value should always be 0.
14 to 12	HYPE[2:0]	HyperFlash or HyperRAM in DDR mode enable 101: HyperFlash or HyperRAM in DDR mode All others: Setting prohibited
11 to 9	Reserved	These bits are always read as 0. The write value should always be 0.
8	ADDRE	Address DDR enable Specifies the SDR or DDR transfer of the address. 0: SDR transfer 1: DDR transfer **Note:** When HyperFlash/HyperRAM is connected, please specify 1 in this bit
7 to 5	Reserved	These bits are always read as 0. The write value should always be 0.
4	OPDRE	Option Data DDR enable Specifies the SDR or DDR transfer of the option data. 0: SDR transfer 1: DDR transfer **Note:** When HyperFlash/HyperRAM is connected, please specify 1 in this bit
3 to 1	Reserved	These bits are always read as 0. The write value should always be 0.
0	SPIDRE	Transfer Data DDR enable Specifies the SDR or DDR transfer of the data for transfer. 0: SDR transfer 1: DDR transfer **Note:** When HyperFlash/HyperRAM is connected, please specify 1 in this bit

56.5.20 PHY Control Register (PHYCNT)

PHYCNT is a 32-bit register that sets the PHY operating mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CAL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	WBUF2	—	WBUF	PHYMEM[1:0]	
Initial value	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W

Table 56.27 PHYCNT register contents

Bit Position	Bit Name	Function
31	CAL	PHY Calibration Executes calibration of the PHY. Be sure to execute calibration before access in manual mode. 1: Calibration is executed. 0: Calibration is not executed
30 to 5	Reserved	These bits are always read as 0. The write value should always be 0.
4	WBUF2	Write Buffer Enable2 The write buffer is used when the flash memory is written to. Refer to Section 56.6.9, Write Buffer Operation for usage of the write buffer. 1: The write buffer is used to write data to the flash memory. 0: The write buffer is not used.
3	Reserved	This bit is always read as 0. The write value should always be 0.
2	WBUF	Write Buffer Enable The write buffer is used when the flash memory is written to. Refer to Section 56.6.9, Write Buffer Operation for usage of the write buffer. 1: The write buffer is used to write data to the flash memory. 0: The write buffer is not used.
1, 0	PHYMEM[1:0]	Device Selection Selects a device to connect. 11: HyperFlash All others: Setting prohibited

56.6 Operation

56.6.1 System Configuration

With this module, one HyperFlash/HyperBus memory device can be connected.

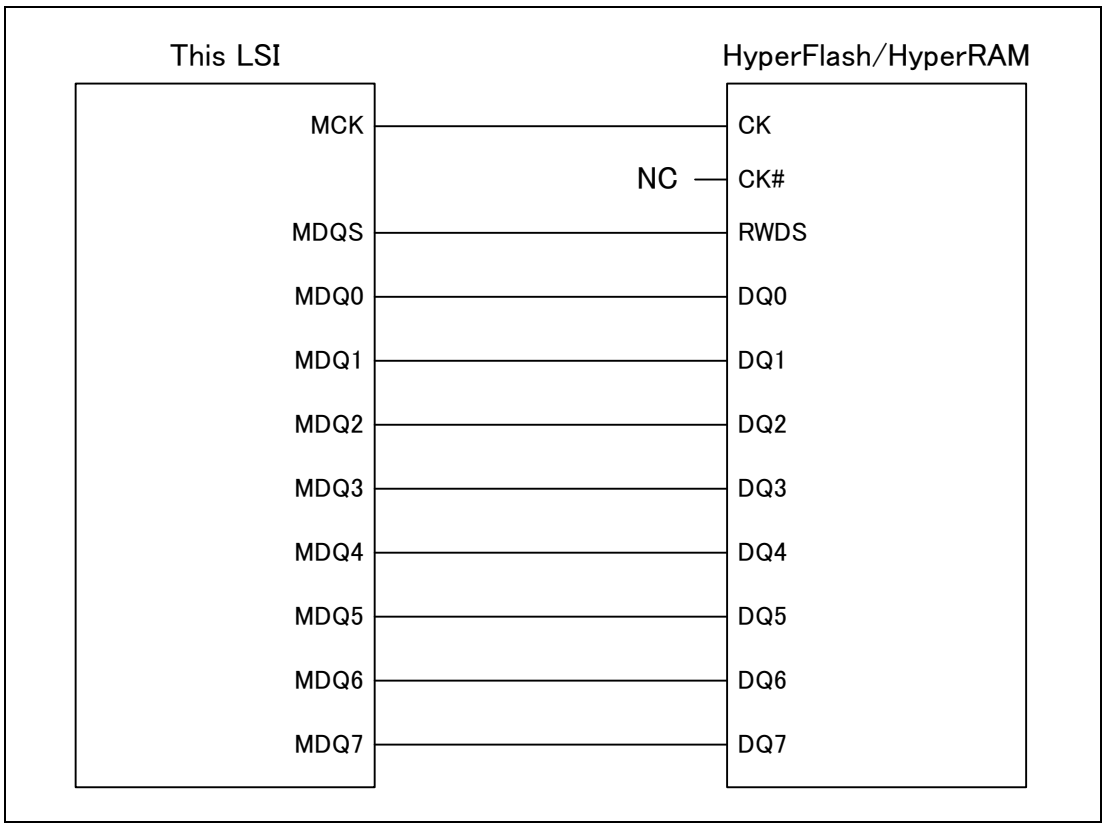


Figure 56.2 Example of system configuration with connected HyperFlash

56.6.2 Address Map

In external address space read mode, the HyperBus to be connected are assigned in the multi I/O bus space. The maximum accessible address space differs depending on the number of HyperBus memory devices connected.

Table 56.28 Address Map

Internal Address	Maximum Access Area
6000 0000$_H$ to 7FFF FFFF	HyperFlash x 1: 512 Mbytes

56.6.3 Operating Modes

This module has two operating modes: external address space read/write mode and manual mode.

In external address space read/write mode, a read access to the HyperBus space is converted into the data read protocol and data is received from the external device. After data acquisition from the external device, data is returned to the bus master that is the issuing source of read access. A write access to the HyperBus space is converted into the write protocol and data is transferred to the external device. HYPB generates write command automatically from DRCMR register value. For details,see Section 56.6.4, External Address Space Read/Write Mode.

The manual mode carries out an arbitrary protocol by using the register settings. For details,see Section 56.6.6, Manual Mode.

56.6.4 External Address Space Read/Write Mode

By external address space read/write mode, a read/write access to the HyperBus space is converted into the data read protocol. Further, the commands, optional commands, option data, and dummy cycle issued for reading external device can be modified using registers.

(1) Initial Setting Flow

An example of an initial setting flow in external address space read/write mode is shown in figure below.

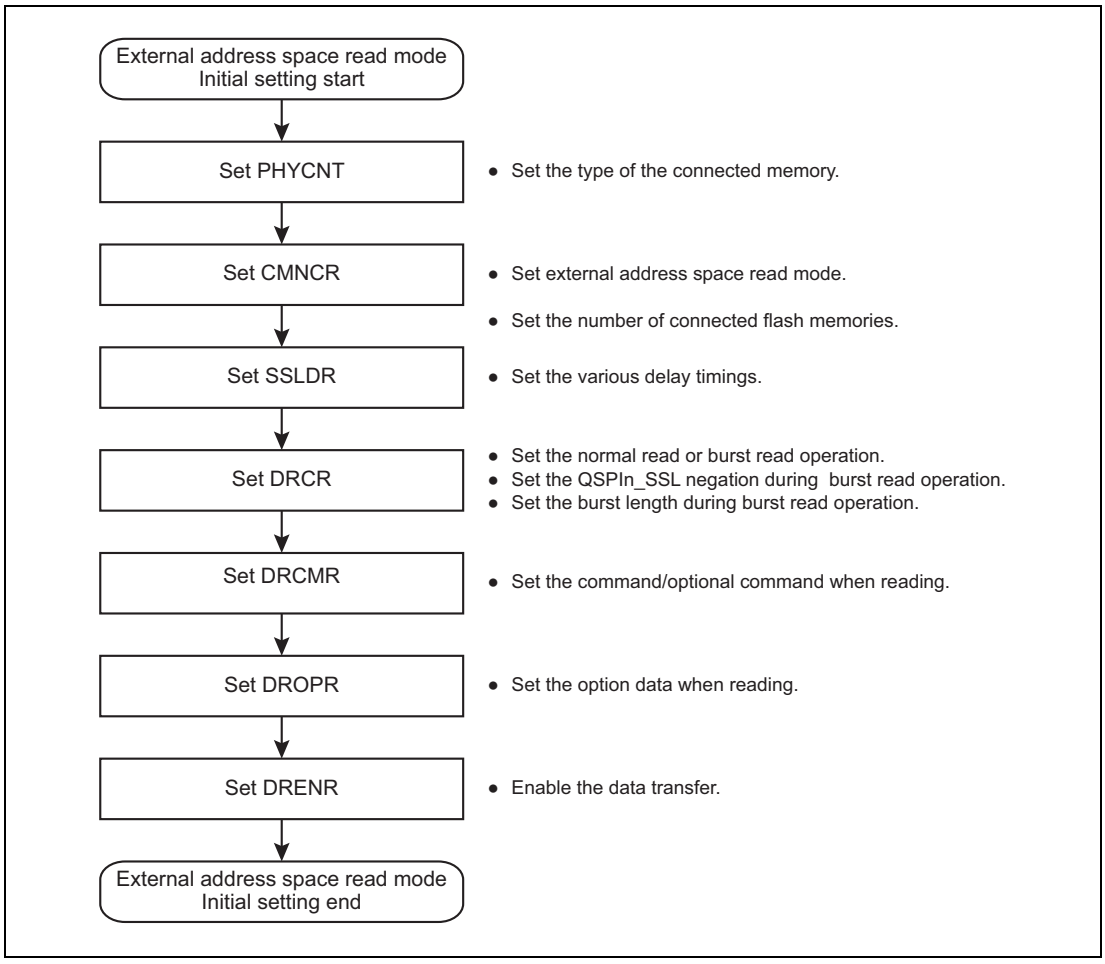


Figure 56.3 Example of initial setting flow in external address space read mode

(2) Initial sequence example for HyperFlash

This example sets latency 8 to the HyperFlash device (S26KLxxxx)

1. Set the configuration register access1 data
   - CMNCR = 0x81ff_f301
   - SSLDR = 0x0000_0000
   - SMCMR = 0x0020_0000
   - SMADR = 0x0000_0555
   - SMOPR = 0x0000_0000
   - SMENR = 0xa222_5408
   - SMWDR0 = 0xaa00_0000
   - SMDRENR = 0x0000_5111
2. Issue the configuration register access1
   - SMCR = 0x0000_0003
3. wait TEND assert 1.
   - wait CMNSR = 0x0000_0001
4. Set the configuration register access2 data
   - SMADR = 0x0000_02aa
   - SMWDR0 = 0x5500_0000
5. Issue the configuration register access2
   - SMCR = 0x0000_0003
6. wait TEND assert 1.
   - wait CMNSR = 0x0000_0001
7. Set the configuration register access3 data
   - SMADR = 0x0000_0555
   - SMWDR0 = 0x3800_0000
8. Issue the configuration register access3
   - SMCR = 0x0000_0003
9. wait TEND assert 1.
   - wait CMNSR = 0x0000_0001
10. Set the non-volatime configuration register data
    - SMADR = 0x0000_0555
    - SMWDR0 = 0x398f_0000
11. Issue the non-volatime configuration register access
    - SMCR = 0x0000_0003
12. wait TEND assert 1.

- wait CMNSR = 0x0000_0001

13. setting internal register
    - CMNCR = 0x01ff_f301
    - DRCMR = 0x00a0_0000
    - DRDMCR = 0x0000_0006
    - DRCR = 0x000f_03e1
14. Start external address space read to HyperFlash

(3) Initial sequence example for HyperRAM

This example sets latency 4 (fixed latency) to the HyperRAM device (S27KLxxxx)

1. Set configuration register access data
    - CMNCR = 0x81ff_f301
    - SSLDR = 0x0000_0000
    - SMCMR = 0x0060_0000
    - SMADR = 0x0000_0800
    - SMOPR = 0x0000_0000
    - SMENR = 0xa222_5408
    - SMWDR0 = 0xff8f_0000
    - SMDRENR = 0x0000_5111
2. Issue the configuration register access1
    - SMCR = 0x0000_0003
3. wait TEND assert 1.
    - wait CMNSR = 0x0000_0001
4. setting internal register
    - CMNCR = 0x01ff_f301
    - DRCMR = 0x00a0_0000
    - DRDMCR = 0x0000_0006
    - DRCR = 0x000f_03e0
5. Start external address space read/write to HyperRAM

56.6.5 Read Cache

This module has a simple built-in read (write-through) cache. The read cache can be used during external address space operation. The read cache is configured with a line size of 64 bits and 32 entries. If this module receives less than 8 bytes data read request from Bus master, it reads 32 bytes data from HyperBus.

Read cache configuration is shown in the figure below.

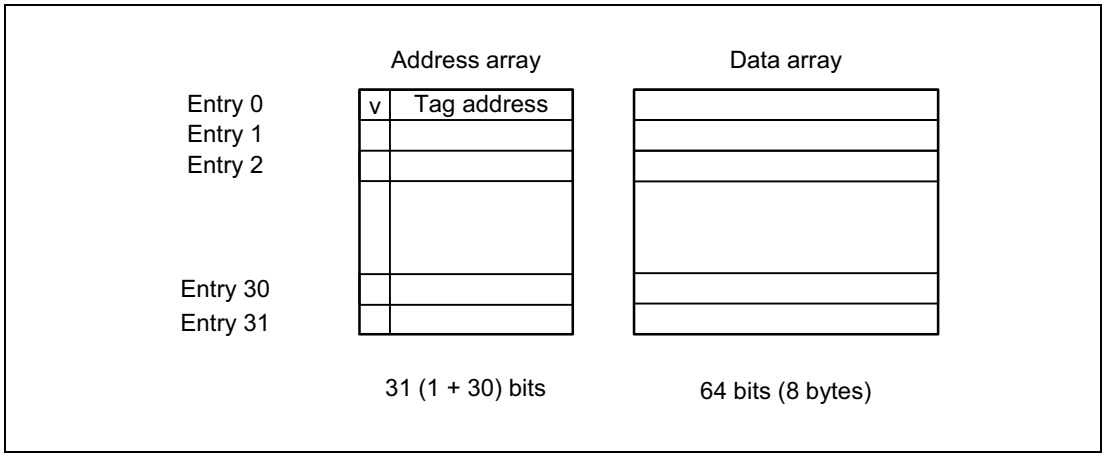


Figure 56.4 Read cache configuration

Address Array

The V bit indicates whether the entry data is valid. When the V bit is 1, the data is valid and when V bit is 0, the data is invalid.

The tag address bits hold the address used for the HyperFlash/HyperRAM memory. Address bits 32 to 3 are used for the purpose.

Data Array

It retains the 64-bit read data. Registration in the read cache is performed in line units.

Read Operation

In case of read-hit, data is read from the read cache. In case of read-miss, after the 64 bits × 4 data is read from the HyperBus memory and the read cache is updated, the data is returned to the bus master.

Data Replacement

The write pointer is used to update data. In case of read-miss, the data is replaced starting at the entry specified by the write pointer. In other words, the data is replaced in the storage order of the data. Whether data is referred to or not will not affect the replacement order of data.

Invalidate Data

In case of a write address hit to cache tag address in side of 1 Kbyte boundary, this module clears the V flag and invalidate cache content.

56.6.6 Manual Mode

This module can carry out an arbitrary serial transfer operation by using the register settings.

Manual mode can be used for reading the status of the HyperBus memory and writing to the HyperBus memory.

In this mode, one transfer refers to the operation from when the SPIE bit in SMCR is set to 1 to when the TEND bit is set to 1.

Initial Setting Flow

An example of an initial setting flow in manual mode is shown in the figure below.

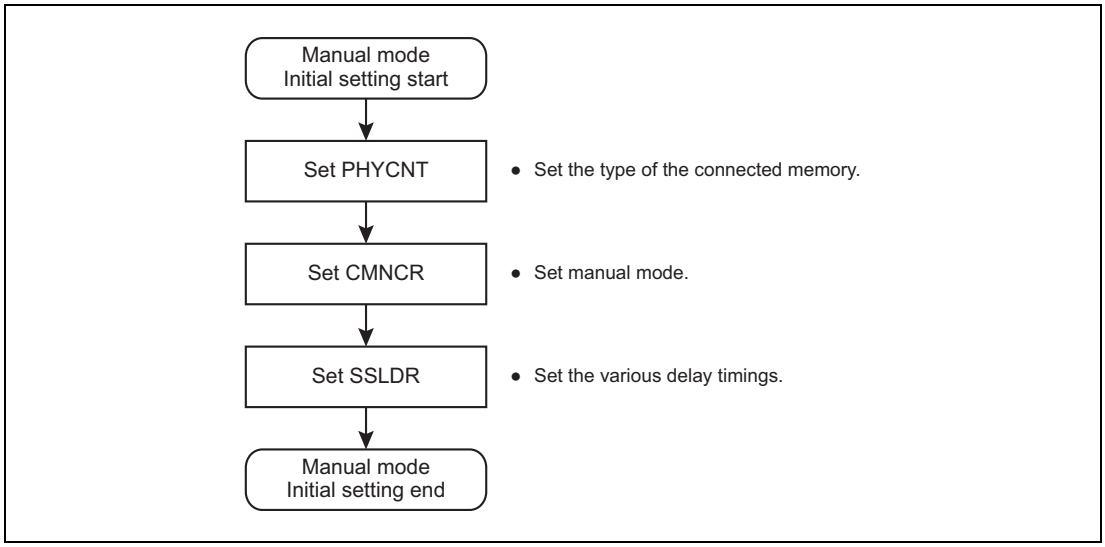


Figure 56.5 Example of initial setting flow in manual operating mode

Data Transfer Setting Flow

An example of a data transfer setting flow in manual mode is shown in the figure below.

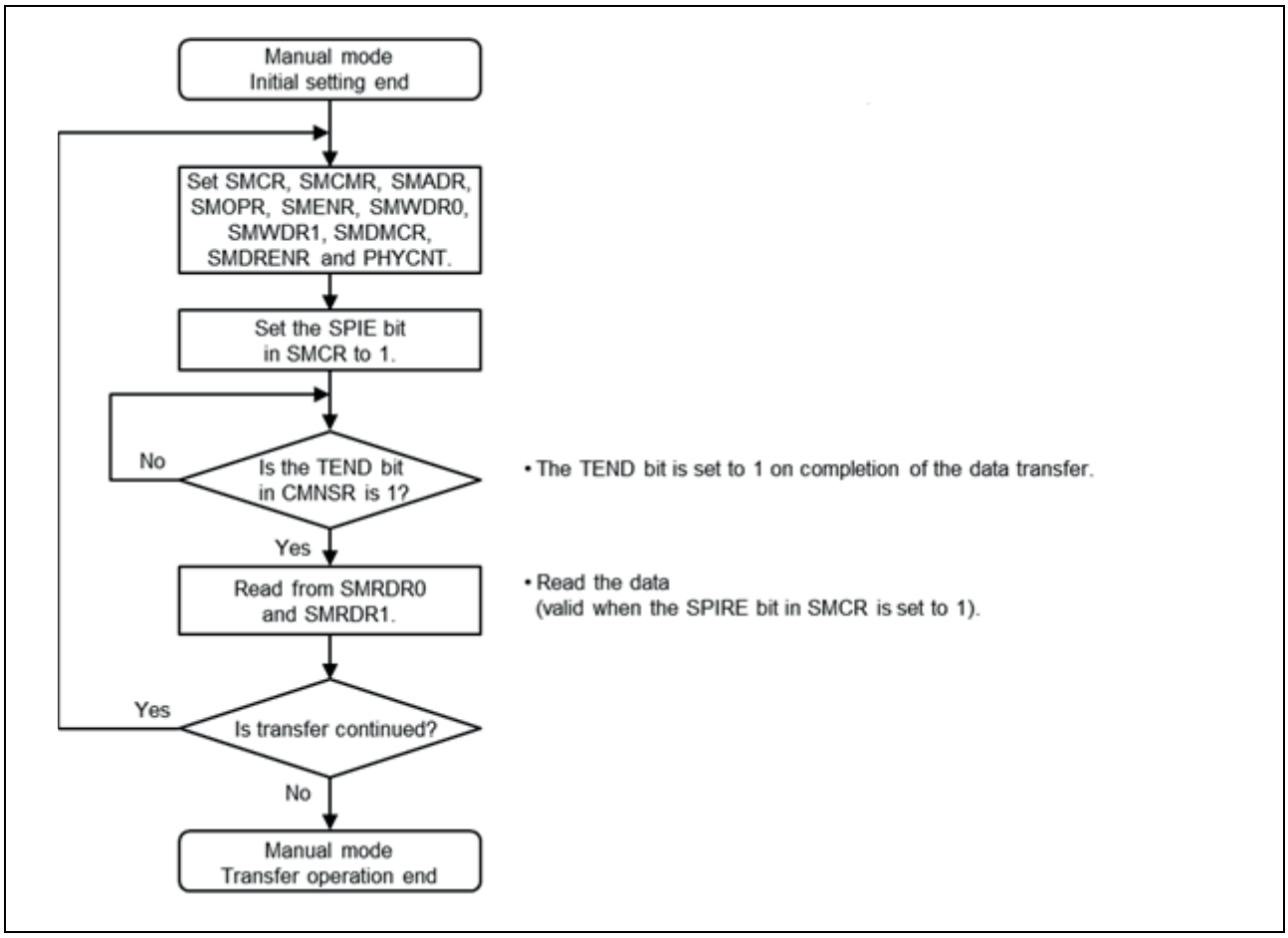


Figure 56.6 Example of data transfer setting flow in manual operating mode

56.6.7 Command Sequence

This module handles input and output data in the order of command, optional command, address, option data, dummy cycle, and data.

(a) 4-bit size (Hyperflash)

The following figures show examples of the transfer format.

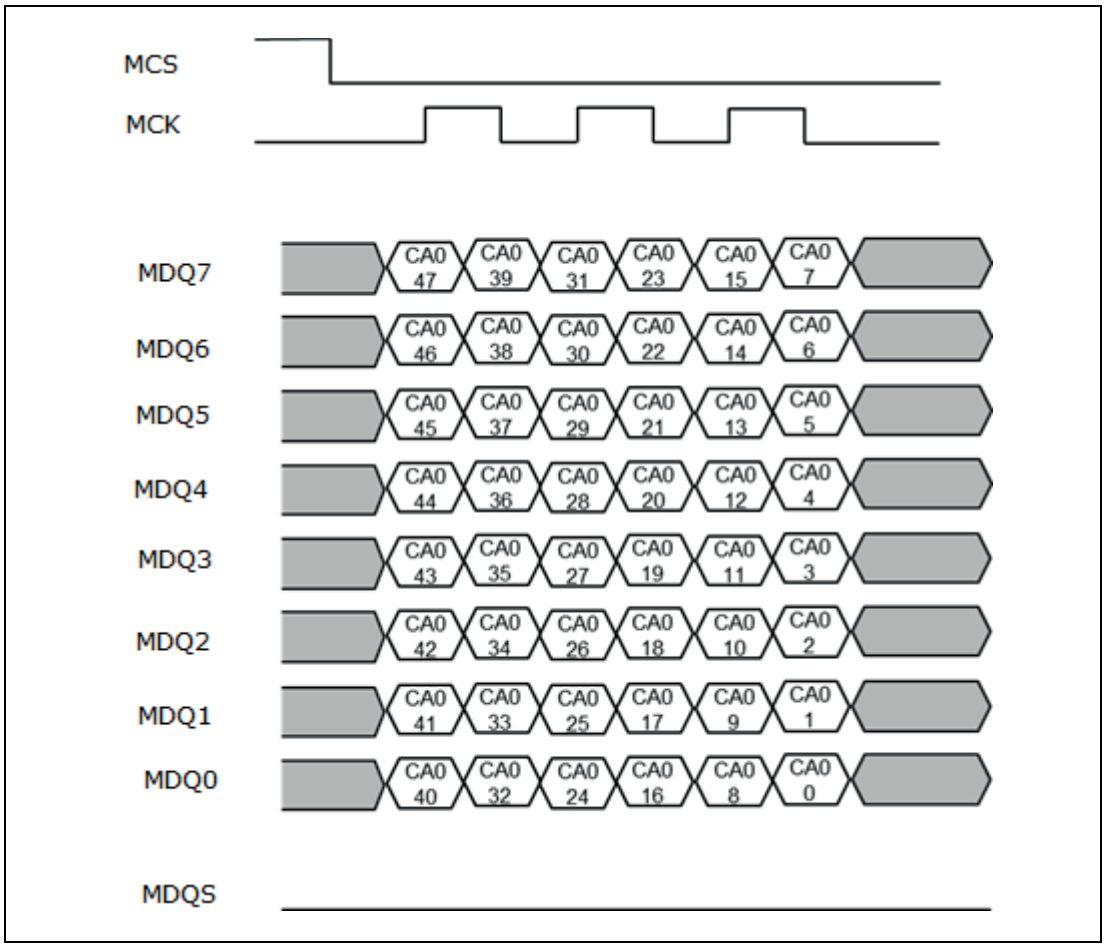


Figure 56.7 Example of transfer format in command/address phase with HyperFlash

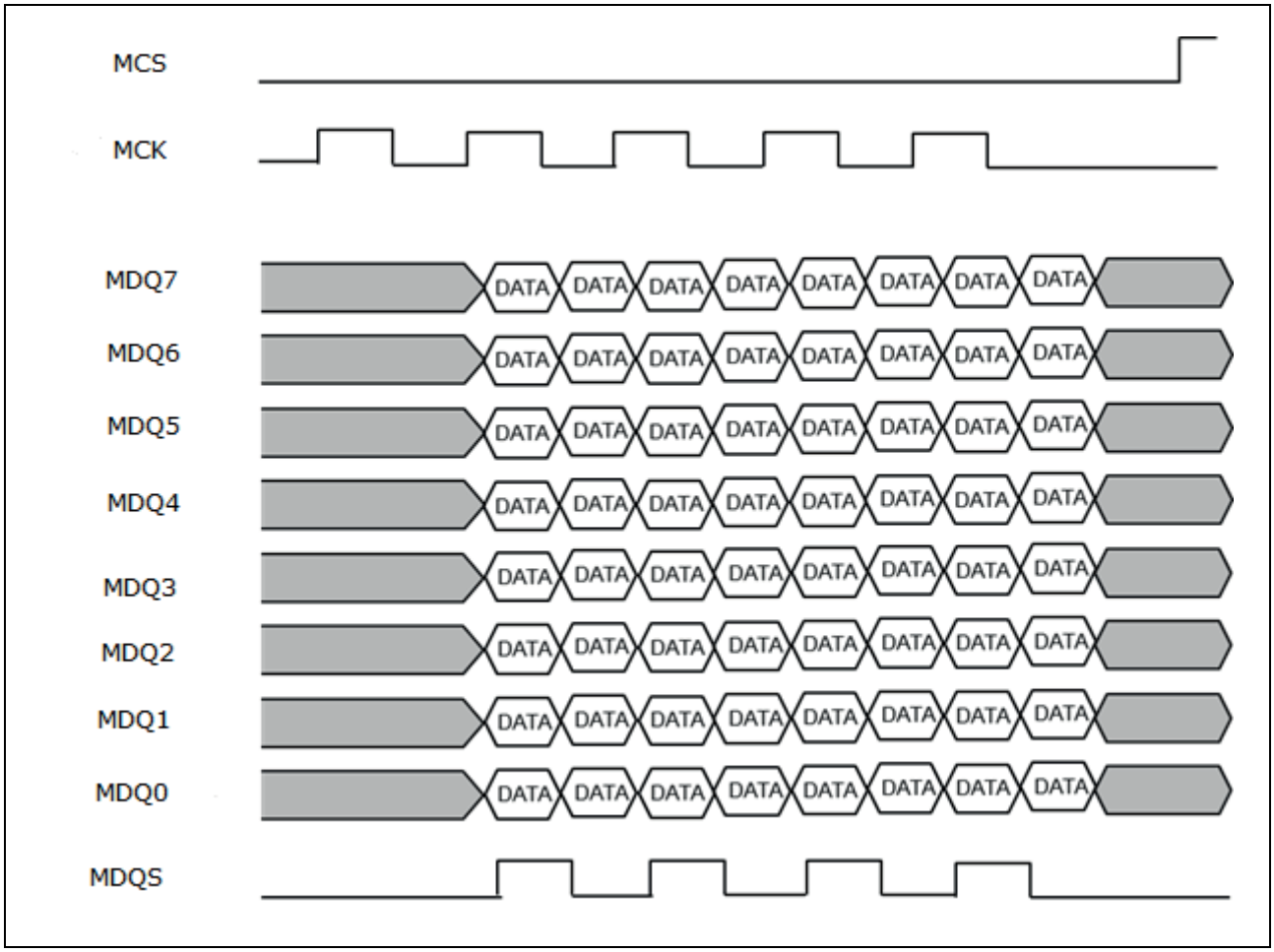


Figure 56.8 Example of transfer format in read phase with HyperFlash

Table 56.29 Command/Address bit assignment of HyperBus

Data	External address space read operation	Manual operation
CA0 (47-40)	CMD[7:5] bits in DRCMR + [31:27] bits of the read address	CMD[7:5] bits in SMCMR + ADR[31:27] bits in SMADR
CA0 (39-32)	[26:19] bits of the read address	ADR[26:19] bits in SMADR
CA1 (31-24)	[18:11] bits of the read address	ADR[18:11] bits in SMADR
CA1 (23-16)	[10:3] bits of the read address	ADR[10:3] bits in SMADR
CA2 (15-8)	DROPR[15:8]	SMOPR[15:8]
CA2 (7-0)	DROPR[7:3] + [2:0] bits of the read address	SMOPR[7:3] + ADR[2:0] bits in SMADR
Dummy cycle	DRDMCR register	SMDMCR register
Transfer data	Normal read: 8, 16, and 32 bits, Burst read: 64 x RBURST bits	Read: SMRDR0 and SMRDR1 Write: SMWDR0 and SMWDR1

56.6.8 Flags

This module has two flag bits SSLF and TEND in CMNSR. These bits are read-only bits.

SSLF Bit

This bit indicates the MCS pin status. The status is 1 when the MCS is asserted, and the status is 0 when the MCS is negated.

TEND Bit

This bit indicates whether transfer of data is in progress or the transfer of data has ended.

During waiting for read access by burst read and MCS automatic negation, the TEND bit is read as 0 to indicate that the transfer of data is in progress.

When other than the above, the TEND bit is read as 1 to indicate that transfer of data has ended.

Register Re-writing Timing

The status of the TEND bit determines the rewritable registers.

The registers which can be written to, except the SSLN bit in DRCR, should be modified when TEND = 1.

The data should be read when the TEND flag in SMRDR0 or SMRDR1 is 1; CMNSR can always be read.

56.6.9 Write Buffer Operation

This module uses the read cache as write buffer in programming operation. This write buffer improves the programming performance.

Table 56.30 Write Buffer Area

Address	Access Size
F2FF 8800_H to F2FF $88FF_H$	8 Byte

NOTE

This area should be accessed sequentially from the start address and transfer size to the device is 8-byte unit.

When accessing non-sequentially or at random, the operation is not guaranteed.

The following figure shows the sequence to use write buffer. About the initial setting of Manual mode, refer to **Figure 56.5**.

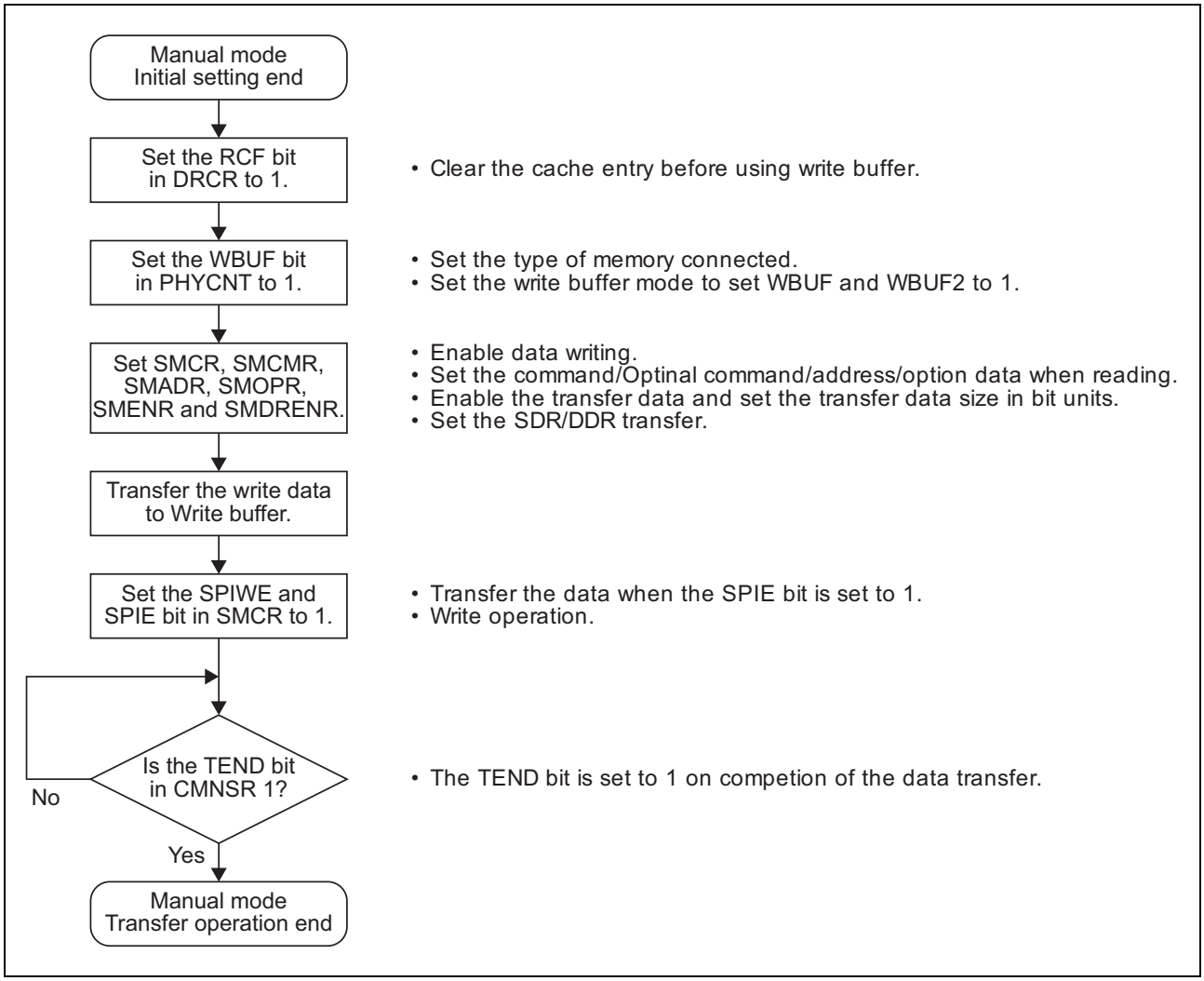


Figure 56.9 Write buffer usage sequence

Section 57 OctaBus Controller (OCTA)

This section contains a generic description of the OctaBus Controller.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

57.1 Overview of RH850/D1L/D1M OctaBus Controller

57.1.1 Units

This microcontroller has the following number of units of the OctaBus Controller.

Table 57.1 Units

OctaBus Controller	D1L1	D1L2(H)	D1M1	D1M1H	D1M1-V2	D1M1-V2 D1M1A	D1M2(H)
Units	–	–	–	–	–	1	–
Names	–	–	–	–	–	OCTA0	–

57.1.2 Indices

Following indices are used in this section:

Table 57.2 Indices

Index	Meaning
n	The individual OCTA units are generically indicated by the index “n”.

57.1.3 Register addresses

All OctaBus Controller register addresses are given as address offsets from the individual base addresses <OCTAn_base>.
The <OCTAn_base> addresses of each OCTAn are listed in the following table:

Table 57.3 Register base addresses <OCTAn_base>

OCTAn unit	<OCTAn_base> address
OCTA0	F2FF 8000$_H$

57.1.4 Clock supply

All OctaBus Controllers provide two clock inputs.

Table 57.4 Clock supply

OCTAn unit	OCTAn clock	Connected to
OCTA0	SCLK	Clock Controller C_ISFMAD_CTL clock divider output
	BUS_CLK	Clock Controller C_ISO_XCCLK

57.1.5 Reset sources

The OctaBus Controllers and their registers are initialized by the following reset signal:

Table 57.5 Reset sources

OCTAn unit	Reset signal
OCTAn	• Reset Controller SYSRES • reset upon wake-up from DEEPSTOP mode

57.1.6 I/O signals

The following table shows the I/O signals of the OctaBus Controller.

Table 57.6 OCTAn I/O signals

OCTAn signals	Function	Connected to
OCTA0		
CLK_O	Clock output	Port MCK
CS_0	Chip select	Port MCS
CS_1*[1]	Chip select 1	Port MCS1
SIO[7:0]_IN / SIO[7:0]_OUT	Data[7:0]	Port MDQI[7:0] / MDQO[7:0]
DQS_IN / DQS_OUT	Read data strobe / write data mask	Port MDQS

Note 1. The OctaBus has a second chip select pin on P45_13. This pin is shared with the SFMA2 interface. To configure this pin, please check the register bit XCRAMCFG0.CSSEL as described in Section 14.4, Bus Switch for external memory interfaces (D1M1-V2, D1M1A only).

CAUTION

The OctaBus and the HyperBus interface share the same ports P21_[9:0] and P22_10. The HyperBus or OctaBus selection via the bus switch control bit XCRAMCFG0.HYPBSEL must be done before the XC0 cross-connect is released from reset, i.e. before MRSTC.XC0RES = 1 is set.

Refer to Section 14.4, Bus Switch for external memory interfaces (D1M1-V2, D1M1A only) for details.

57.2 Features

- Support Macronix Serial Multi I/O (MXSMIO®) Octa Peripheral Interface (OPI) interface for high-end consumer applications
- Compatible with Macronix OctaFlash/OctaRAM family
- Support 3-byte/4-byte OctaFlash address command
- Support 4-byte OctaRAM address command
- Support memory-mapped read/write feature with independent Flash(RW)/RAM(RW) address space
- Configurable address space of configuration/flash/RAM address
- Two device channel can configured as Flash only, RAM only, Flash/Flash, Flash/RAM, RAM/RAM
- Shared bus architecture with only split CS pin could reduce system pin counts
- Support Read-While-Write function (RWW) (please reference OctaRAM data sheet)
- Support 1LC/2LC OctaRAM (Please reference Macronix OctaFlash Automotive data sheet)

57.3 General Description

The reference design introduces an Octa NOR Flash/RAM controller for Macronix Octa NOR Flash/RAM products.
Octa I/O read/program mode is supported for high performance applications. The controller supports several types of I/O mode of OctaFlash memory products, OctaRAM only support DTOIO mode:

- SIO (OctaFlash, Single I/O STR, 1 bit per cycle)
- OIO (OctaFlash, Octa I/O STR, 8 bit per cycle)
- DTOIO (OctaFlash/OctaRAM, Octa I/O DTR, 16 bit per cycle)

STR (Single Transfer Rate) is basic flash access mode for flash device access. The data are triggered on single edge of clock as shown in Figure 57.1, STR Read Sequence (Flash Only).
The DTR (Double Transfer Rate) mode is for doubling the frequency of reading data. The data are triggered on both rising and falling edge of clock, as shown in Figure 57.2, DTR Read Sequence (Flash/RAM).

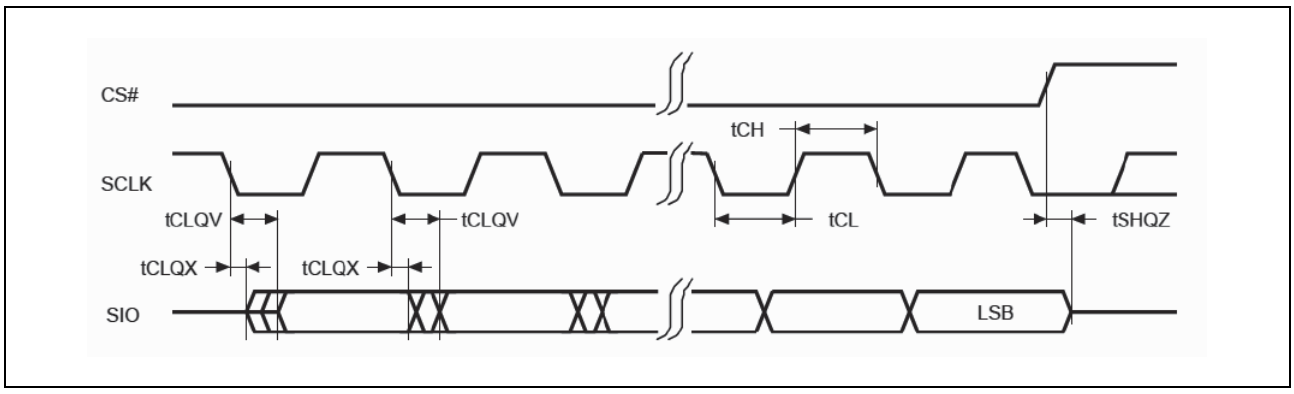


Figure 57.1 STR Read Sequence (Flash Only)

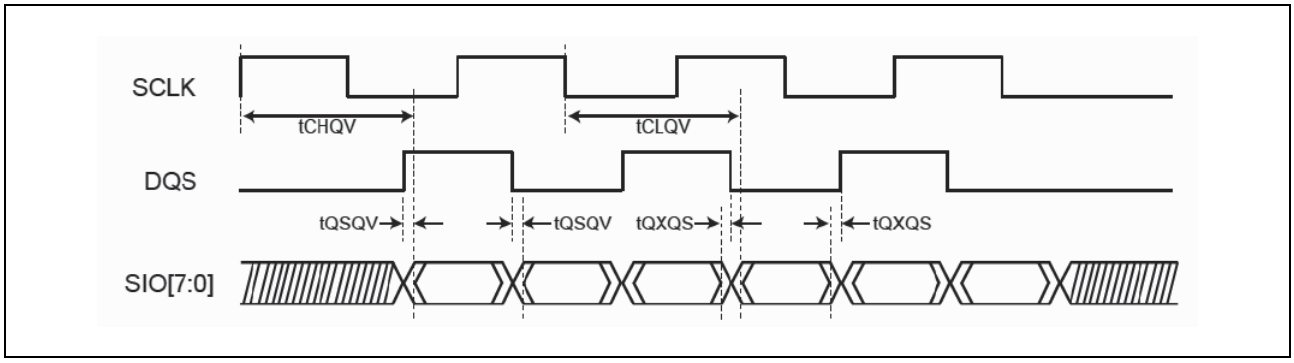


Figure 57.2 DTR Read Sequence (Flash/RAM)

57.4 Configuration Registers

57.4.1 Register configuration overview

The following table shows the register configuration of the OctaBus Controller.

Table 57.7 Register configuration (1/2)

Register name	Shortcut	Address
Device command register	DCR	<OCATn_base> + 00_H
Device address register	DAR	<OCATn_base> + 04
Device command setting register	DCSR	<OCATn_base> + 08
Device size register 0	DSR0	<OCATn_base> + 0C
Device size register 1	DSR1	<OCATn_base> + 10
Memory delay trim register	MDTR	<OCATn_base> + 14
Auto-calibration timer register	ACTR	<OCATn_base> + 18
Auto-Calibration Address Register 0	ACAR0	<OCATn_base> + 1C
Auto-Calibration Address Register 1	ACAR1	<OCATn_base> + 20
Device Memory Map Read chip select timing setting register	DRCSTR	<OCATn_base> + 34
Device Memory Map Write chip select timing setting register	DWCSTR	<OCATn_base> + 38
Device chip select timing setting register	DCSTR	<OCATn_base> + 3C
Controller and device setting register	CDSR	<OCATn_base> + 40
Memory Map dummy length register	MDLR	<OCATn_base> + 44
Memory Map read/write command register 0	MRWCR0	<OCATn_base> + 48
Memory Map read/write command register 1	MRWCR1	<OCATn_base> + 4C
Memory Map read/write command setting register	MRWCSR	<OCATn_base> + 50
Error Status Register	ESR	<OCATn_base> + 54
Configure write without data register	CWNDR	<OCATn_base> + 58
Configure write data register	CWDR	<OCATn_base> + 5C
Configure read register	CRR	<OCATn_base> + 60
Auto-calibration status register	ACSR	<OCATn_base> + 64
Auto-calibration result register 0	ACRR0	<OCATn_base> + 68

Table 57.7 Register configuration (2/2)

Register name	Shortcut	Address
Auto-calibration result register 1	ACRR1	<OCATn_base> + 6C
Auto-calibration all scan result register 0	ACASRR0	<OCATn_base> + 70
Auto-calibration all scan result register 1	ACASRR1	<OCATn_base> + 74

<OCTAn_base>

The base addresses <OCTAn_base> of the OCTAn is defined in the first subsection of this section under the key word “Register addresses”.

57.4.1.1 Device command register (DCR)

The device command register is used to set the device commands for controller operation.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 00_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DVC3[7:0]								DVC2[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DVC1[7:0]								DVC0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.8 DCR Register Contents

Bit Position	Bit Name	Function
31 to 24	DVC3[7:0]	Set device command After determine the command length (DCSR[23:20]), the controller will send the corresponding device command to devices. • If DCSR[22:20] = 1, the controller will send DCR[7:0] to memory. • If DCSR[22:20] = 2, the controller will send DCR[15:8] as the 1st byte, and DCR[7:0] as the 2nd byte to memory.
23 to 16	DVC2[7:0]	
15 to 8	DVC1[7:0]	
7 to 0	DVC0[7:0]	

57.4.1.2 Device address register (DAR)

The device address register is used to set the device address for controller operation.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 04_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DVA3[7:0]								DVA2[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DVA1[7:0]								DVA0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.9 DAR Register Contents

Bit Position	Bit Name	Function
31 to 24	DVA3[7:0]	Set device address After determine the address length (DCSR[27:24]), the controller will send the corresponding device address to devices. • If DCSR[27:24] == 3, the controller will send DAR[23:16] as the 1st byte, DAR[15:8] as the 2nd byte, and DAR[7:0] as the 3rd byte to memory. • If DCSR[27:24] == 4, the controller will send DAR[31:24] as the 1st byte, DAR[23:16] as the 2nd byte, DAR[15:8] as the 3rd byte, and DAR[7:0] as the 4th byte to memory.
23 to 16	DVA2[7:0]	
15 to 8	DVA1[7:0]	
7 to 0	DVA0[7:0]	

57.4.1.3 Device command setting register (DCSR)

The device command setting register is used to set the command setting.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 08_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	PRBE	ACMA	DOPI SB	ADL[2:0]			DTO	CML[2:0]			ACD	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DML[7:0]								DTL[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.10 DCSR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 30	Reserved	Reading these bits return the initial value.
29	PRBE	Preamble bit enable for configuration read
28	ACMA	Access Memory Array 1: access memory array 0: configuration command • When ACMA = 1, (1) In DOPI mode, the address bit0 will be forced to 0. (2) In DOPI mode, if a read type command's DAR[0] is 1, the byte0 from device will be dropped. (3) The AXI RDATA will be arranged according to DAR. For example, when DAR[1:0]=1, the 1st valid data will be put at RDATA[15:8]. (4) In DOPI mode, the AXI WDATA will be used according to DAR. For example: when the target device is flash, and when DAR[1:0] = 1, the 1st byte sent to flash will be forced to 0xff. when the target device is RAM and when DAR[1:0] = 1, the 1st byte sent to ram will be masked by DQSM. (6) When the target device is RAM, the address sent to memory will be rearranged according to RAM spec. • When ACMA = 0, the address will be sent to memory without any modification, and the output data from memory will be sent out without any modification.
27	DOPISB	DOPI Single Byte is used to indicate that although the memory is in DOPI mode, the read out data has only 1 byte in each cycle. For example, the RDID, RDSR command of OctaFlash. 0: normal DOPI mode (each cycle has 2 bytes data, one at positive edge, the other one at negative edge.) 1: single byte mode (each cycle has only one byte data; the positive edge and negative edge data is the same)
26 to 24	ADL[2:0]	Address Length is used to set the transferring address bytes for current command. Its unit is byte.

Table 57.10 DCSR Register Contents (2/2)

Bit Position	Bit Name	Function
23	DTO	Data Order is used to set the read/write data transfer from/to memory in DOPI mode. In SOPI/SPI mode, Data Order is don't care. 0: byte0, byte1, byte2, byte3 1: byte1, byte0, byte3, byte2 If user sets the DTO to 1, the data length should be set greater than 2 while executing WRCR, WRCR2, WRSR, WRLR, and WRDPB commands in DOPI mode.
22 to 20	CML[2:0]	Command Length is used to set the transferring command bytes for current command. Its unit is byte. Command Length equals to 0 means no command will be transferred.
19	ACD	Access Device 0: Send commands to device 0 1: Send commands to device 1
18 to 16	Reserved	Reading these bits return the initial value.
15 to 8	DML[7:0]	Dummy Length – set the dummy length of the command. Its unit is SCLK cycle.
7 to 0	DTL[7:0]	Data Length – set the data length of the command. Its unit is byte. If DCSR[7:0] is 0, there will be no data for sending to memory or receiving from memory.

57.4.1.4 Device size register 0 (DSR0)

The device size register 0 is used to set the device type and size (can be flash or RAM) to access for device 0.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + $0C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DV0T[1:0]		DV0S[29:16]													
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DV0S[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.11 DSR0 Register Contents

Bit Position	Bit Name	Function
31 to 30	DV0T[1:0]	Set device info 00_B: flash on device 0 01_B: RAM on device 0 10_B: no connection on device 0 11_B: forbidden
29 to 0	DV0S[29:0]	Set 30-bit device size mounted on system. Default size is 256 MByte. For example: 0x1000 0000: 256 MByte Flash 0x0800 0000: 128 MByte Flash 0x4100 0000: 16 MByte RAM 0x4080 0000: 8 MByte RAM The actual address window for device 0 is 0x6000_0000 to 0x6000_0000 + DSR0. Note that address window of device 0 and device 1 can't be overlapping. If they are overlapping, they will be set abutted.

57.4.1.5 Device size register 1 (DSR1)

The device size register 1 is used to set the device type and size (can be flash or RAM) to access for device 1.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 10_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DV1T[1:0]		DV1S[29:16]													
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DV1S[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.12 DSR1 Register Contents

Bit Position	Bit Name	Function
31 to 30	DV1T[1:0]	Set device info 00_B: flash on device 1 01_B: RAM on device 1 10_B: no connection on device 1 11_B: forbidden
29 to 0	DV1S[29:0]	Set 30-bit device size mounted on system. Default size is 256 MByte. For example: 0x1000 0000: 256 MByte Flash 0x0800 0000: 128 MByte Flash 0x4100 0000: 16 MByte RAM 0x4080 0000: 8 MByte RAM The actual address window for device 1 is 0x7000_0000 to 0x7000_0000 + DSR1. Note that address window of device 0 and device 1 can't be overlapping. If they are overlapping, they will be set abutted.

57.4.1.6 Memory delay trim register (MDTR)

Set the delay line control value for device 0 and device 1.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 14_H

Initial value: 0600 9400_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	DQED[3:0]				DV1D[7:0]							
Initial value	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DQES[3:0]				DQER[3:0]				DV0D[7:0]							
Initial value	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.13 MDTR Register Contents

Bit Position	Bit Name	Function
31 to 28	Reserved	Reading these bits return the initial value.
27 to 24	DQED[3:0]	DQS enable counter in DOPI mode During read operation with DQS clock input (RAM/SOPI/DOPI mode), DQS clock will transition from high-impedance to zero after command/address phase is finished. To prevent using such invalid DQS clock, user can adjust these configuration to guarantee data correctness. Configuration: 0000_B: Enable DQS clock input at 1st SCLK 0001_B: Enable DQS clock input at 2nd SCLK 0010_B: Enable DQS clock input at 3rd SCLK 0011_B: Enable DQS clock input at 4th SCLK ... 1111_B: Enable DQS clock input at 16th SCLK Reference DQS enable counter settings (when SCLK frequency < 120 MHz)

DQS delay [SCLK]	RAM		Flash SOPI	Flash DOPI	
	Pre-cycle On	Pre-cycle Off	Pre-cycle On	Pre-cycle On	Pre-cycle Off
0 - 0.5	3	3 - 4	7 - 8	5	5 - 6
0.5 - 1.5	4	5	8	6	6 - 7
1.5 - 2.5	5	5 - 6	9	7	7 - 8

		DQS delay value: Delay value = SCLK output pad delay + SCLK wire bonding + DQS wire bonding + DQS input pad delay (Memory device delay has considered in this table, maximum 2 SCLK cycle)
23 to 16	DV1D[7:0]	Device 1 delay (refer to DQED[3:0] description)
15 to 12	DQES[3:0]	DQS enable counter in SOPI mode (refer to DQED[3:0] description)
11 to 8	DQER[3:0]	DQS enable counter in RAM mode (refer to DQED[3:0] description)
7 to 0	DV0D[7:0]	Device 0 delay (refer to DQED[3:0] description)

57.4.1.7 Auto-calibration timer register (ACTR)

The auto-calibration timer register sets the time period of the DQS delay-line for auto-calibration. The unit of timer is bus clock period.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 18_H

Initial value: $1000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	TMPD[31:16]															
Initial value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TMPD[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.14 ACTR Register Contents

Bit Position	Bit Name	Function
31 to 0	TMPD[31:0]	Set 32-bit time-out period of the DQS delay-line for auto-calibration. Re-calibration Time = AC_TIMER * T_{BUS_CLK} When auto-calibration mode is enabled (CDSR[11:10]) Auto-calibration sequence will start if internal timer of auto-calibration equal to this register

57.4.1.8 Auto-calibration address register 0 (ACAR0)

Set 32-bit auto-calibration address of device 0 for write/ read the auto-calibration pattern.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + $1C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.15 ACAR0 Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Set 32-bit auto-calibration address of device 0.

57.4.1.9 Auto-calibration address register 1 (ACAR1)

Set 32-bit auto-calibration address of device 0 for write/ read the auto-calibration pattern.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 20_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	ADDR[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.16 ACAR1 Register Contents

Bit Position	Bit Name	Function
31 to 0	ADDR[31:0]	Set 32-bit auto-calibration address of device 1.

57.4.1.10 Device memory map read chip select timing setting register (DRCSTR)

Set the device select timing value for memory map read.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 34_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCSLFC1[1:0]		DCSHAC1[2:0]			DCSBC1[2:0]			CRWE1	CRWC1[6:0]						
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCSLFC0[1:0]		DCSHAC0[2:0]			DCSBC0[2:0]			CRWE0	CRWC0[6:0]						
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.17 DRCSTR Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 30	DCSLFC1[1:0]	Device chip select low for command setting 1 For device 1, set the period for issuing command when the device chip select pull down. Its unit is MCLK cycle.
29 to 27	DCSHAC1[2:0]	Device chip select high after command setting 1 For device 1, set the period for the device chip select pull up after the command is finished. Its unit is MCLK cycle.
26 to 24	DCSBC1[2:0]	Device chip select between commands setting 1 For device 1, set the period between two commands. Its unit is MCLK cycle.
23	CRWE1	Continuous Read Waiting Enable 1 For device 1, set continuous read waiting enable. When it is set to 1, the continuous read waiting cycle will be used. The purpose of this function is designed for improve continuous read operation of OctaFlash DOPI mode. Please make sure the memory is OctaFlash and under DOPI-mode before running continuous read using this option 0: disable continuous read waiting 1: enable continuous read waiting
22 to 16	CRWC1[6:0]	Continuous Read Waiting Cycles 1 For device 1, set the waiting cycles for continuous read command. It will keep chip select of device low for waiting next continuous read transaction after finishing a continuous read transaction until the waiting cycle exceeds CRWC1[6:0]. Note that the waiting cycle is CRWC1[6:0] * 2, unit is MCLK cycle.
15 to 14	DCSLFC0[1:0]	Device chip select low for command setting 0 For device 0, set the period for issuing command when the device chip select pull down. Its unit is MCLK cycle.
13 to 11	DCSHAC0[2:0]	Device chip select high after command setting 0 For device 0, set the period for the device chip select pull up after the command is finished. Its unit is MCLK cycle.
10 to 8	DCSBC0[2:0]	Device chip select between commands setting 0 For device 0, set the period between two commands. Its unit is MCLK cycle.

Table 57.17 DRCSTR Register Contents (2/2)

Bit Position	Bit Name	Function
7	CRWE0	Continuous Read Waiting Enable 0 For device 0, set continuous read waiting enable. When it is set to 1, the continuous read waiting cycle will be used. The purpose of this function is designed for improve continuous read operation of OctaFlash DOPI mode. Please make sure the memory is OctaFlash and under DOPI-mode before running continuous read using this option 0: disable continuous read waiting 1: enable continuous read waiting
6 to 0	CRWC0[6:0]	Continuous Read Waiting Cycles 0 For device 0, set the waiting cycles for continuous read command. It will keep chip select of device low for waiting next continuous read transaction after finishing a continuous read transaction until the waiting cycle exceeds CRWC0[6:0]. Note that the waiting cycle is CRWC0[6:0] * 2, unit is MCLK cycle.

Timing definitions (x = 0, 1)

Table 57.18 Chip select low timing definition (from CS low to the first SCLK high)

DCSLFCx[3:0]	DOPI mode	Other modes
0	1.5 MCLK cycles	2 MCLK cycles
1	2.5 MCLK cycles	3 MCLK cycles
2	3.5 MCLK cycles	4 MCLK cycles
3	4.5 MCLK cycles	5 MCLK cycles

Table 57.19 Chip select high timing definition (from the last SCLK low to CS high)

DCSHACx[1:0]	DOPI mode	Other modes
0	1.5 MCLK cycles	2 MCLK cycles
1	2.5 MCLK cycles	3 MCLK cycles
2	3.5 MCLK cycles	4 MCLK cycles
3	4.5 MCLK cycles	5 MCLK cycles
4	5.5 MCLK cycles	6 MCLK cycles
5	6.5 MCLK cycles	7 MCLK cycles
6	7.5 MCLK cycles	8 MCLK cycles
7	8.5 MCLK cycles	9 MCLK cycles

Table 57.20 Chip select between two commands definition (from CS high to the next CS low)

DCSBCx[3:0]	All modes
0	2 MCLK cycles
1	5 MCLK cycles
2	7 MCLK cycles
3	9 MCLK cycles
4	11 MCLK cycles
5	13 MCLK cycles
6	15 MCLK cycles
7	17 MCLK cycles

57.4.1.11 Device memory map write chip select timing setting register (DWCSTR)

Set the device select timing value for memory map write.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 38_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DCSLFC1[1:0]		DCSHAC1[2:0]			DCSBC1[2:0]			—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCSLFC0[1:0]		DCSHAC0[2:0]			DCSBC0[2:0]			—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 57.21 DWCSTR Register Contents

Bit Position	Bit Name	Function
31 to 30	DCSLFC1[1:0]	Device chip select low for command setting 1 For device 1, set the period for issuing command when the device chip select pull down. Its unit is MCLK cycle.
29 to 27	DCSHAC1[2:0]	Device chip select high after command setting 1 For device 1, set the period for the device chip select pull up after the command is finished. Its unit is MCLK cycle.
26 to 24	DCSBC1[2:0]	Device chip select between commands setting 1 For device 1, set the period between two commands. Its unit is MCLK cycle.
23 to 16	Reserved	Reading these bits return the initial value.
15 to 14	DCSLFC0[1:0]	Device chip select low for command setting 0 For device 0, set the period for issuing command when the device chip select pull down. Its unit is MCLK cycle.
13 to 11	DCSHAC0[2:0]	Device chip select high after command setting 0 For device 0, set the period for the device chip select pull up after the command is finished. Its unit is MCLK cycle.
10 to 8	DCSBC0[2:0]	Device chip select between commands setting 0 For device 0, set the period between two commands. Its unit is MCLK cycle.
7 to 0	Reserved	Reading these bits return the initial value.

Timing definitions (x = 0, 1)

Table 57.22 Chip select low timing definition (from CS low to the first SCLK high)

DCSLFCx[3:0]	DOPI mode	Other modes
0	1.5 MCLK cycles	2 MCLK cycles
1	2.5 MCLK cycles	3 MCLK cycles
2	3.5 MCLK cycles	4 MCLK cycles
3	4.5 MCLK cycles	5 MCLK cycles

Table 57.23 Chip select high timing definition (from the last SCLK low to CS high)

DCSHAFCx[1:0]	DOPI mode	Other modes
0	1.5 MCLK cycles	2 MCLK cycles
1	2.5 MCLK cycles	3 MCLK cycles
2	3.5 MCLK cycles	4 MCLK cycles
3	4.5 MCLK cycles	5 MCLK cycles
4	5.5 MCLK cycles	6 MCLK cycles
5	6.5 MCLK cycles	7 MCLK cycles
6	7.5 MCLK cycles	8 MCLK cycles
7	8.5 MCLK cycles	9 MCLK cycles

Table 57.24 Chip select between two commands definition (from CS high to the next CS low)

DCSBCx[3:0]	All modes
0	2 MCLK cycles
1	5 MCLK cycles
2	7 MCLK cycles
3	9 MCLK cycles
4	11 MCLK cycles
5	13 MCLK cycles
6	15 MCLK cycles
7	17 MCLK cycles

57.4.1.12 Device chip select timing setting register (DCSTR)

Set the device select timing value.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + $3C_H$

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DCSLFC[1:0]		DCSHAC[2:0]			DCSBC[2:0]			—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R

Table 57.25 DCSTR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	Reading these bits return the initial value.
15 to 14	DCSLFC[1:0]	Device chip select low for command setting Set the period for issuing command when the device chip select pull down. Its unit is MCLK cycle.
13 to 11	DCSHAC[2:0]	Device chip select high after command setting Set the period for the device chip select pull up after the command is finished. Its unit is MCLK cycle.
10 to 8	DCSBC[2:0]	Device chip select between commands setting Set the period between two commands. Its unit is MCLK cycle.
7 to 0	Reserved	Reading these bits return the initial value.

Timing definitions

Table 57.26 Chip select low timing definition (from CS low to the first SCLK high)

DCSLFC[3:0]	DOPI mode	Other modes
0	1.5 MCLK cycles	2 MCLK cycles
1	2.5 MCLK cycles	3 MCLK cycles
2	3.5 MCLK cycles	4 MCLK cycles
3	4.5 MCLK cycles	5 MCLK cycles

Table 57.27 Chip select high timing definition (from the last SCLK low to CS high)

DCSHACx[1:0]	DOPI mode - read/write command	Other modes
0	1.5 MCLK cycles	2 MCLK cycles
1	2.5 MCLK cycles	3 MCLK cycles
2	3.5 MCLK cycles	4 MCLK cycles
3	4.5 MCLK cycles	5 MCLK cycles
4	5.5 MCLK cycles	6 MCLK cycles
5	6.5 MCLK cycles	7 MCLK cycles
6	7.5 MCLK cycles	8 MCLK cycles
7	8.5 MCLK cycles	9 MCLK cycles

Table 57.28 Chip select between two commands definition (from CS high to the next CS low)

DCSBCx[3:0]	All modes
0	2 MCLK cycles
1	5 MCLK cycles
2	7 MCLK cycles
3	9 MCLK cycles
4	11 MCLK cycles
5	13 MCLK cycles
6	15 MCLK cycles
7	17 MCLK cycles

57.4.1.13 Controller and device setting register (CDSR)

Controller and device setting register is used to set the auto-calibration enable, device output byte order, and device transfer type.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 40_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DLFTE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	ACMD[1:0]		ACME[1:0]		—	—	—	—	D1PC	D0PC	D1TT[1:0]		D0TT[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.29 CDSTR Register Contents (1/2)

Bit Position	Bit Name	Function
31	DLFTE	Deadlock-free timer enable enable timer to prevent deadlock case of controller. After time out occurred, controller will response BUSERR to XC connect bus. 0: Enable timer (default) 1: Disable timer
30 to 14	Reserved	Reading these bits return the initial value.
13 to 12	ACMD[1:0]	ac_mode ac_mode can't be enabled before finishing programing flash. 00_B: auto-calibration is not enable 01_B: enable executing auto-calibration periodically by timer and modify MDTR automatically 10_B: execute auto-calibration once for all trim code scan immediately, but it will not modify MDTR. It will clear to 00_B automatically after finishing auto-calibration. 11_B: forbidden
11 to 10	ACME[1:0]	ac_mem_en memory enable auto-calibration Auto-calibration only supports DOPI mode. If exiting DOPI mode, please turn-off auto-calibration.
9 to 6	Reserved	Reading these bits return the initial value.
5	D1PC	Device 1 pre-cycle D1PC is used to set the pre-cycle mode of memory 0: pre-cycle mode off 1: pre-cycle mode on
4	D0PC	Device 0 pre-cycle D0PC is used to set the pre-cycle mode of memory 0: pre-cycle mode off 1: pre-cycle mode on
3 to 2	D1TT[1:0]	Device 1 transfer type D1TT[1:0] is used to set the transfer type of device 1 00_B: SPI mode 01_B: SOPI mode 10_B: DOPI mode 11_B: forbidden

Table 57.29 CDSTR Register Contents (2/2)

Bit Position	Bit Name	Function
1 to 0	D0TT[1:0]	Device 0 transfer type D1TT[1:0] is used to set the transfer type of device 0 00_B: SPI mode 01_B: SOPI mode 10_B: DOPI mode 11_B: forbidden

57.4.1.14 Memory map dummy length register (MDLR)

Set the dummy length for memory map read/write.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 44_H

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	D1WDL[7:0]								D1RDL[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	D0WDL[7:0]								D0RDL[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.30 MDLR Register Contents

Bit Position	Bit Name	Function
31 to 24	D1WDL[7:0]	Device 1 write dummy length Set the dummy length to device 1 (unit = SCLK cycle)
23 to 16	D1RDL[7:0]	Device 1 read dummy length Set the dummy length to device 1 (unit = SCLK cycle)
15 to 8	D0WDL[7:0]	Device 0 write dummy length Set the dummy length to device 1 (unit = SCLK cycle)
7 to 0	D0RDL[7:0]	Device 0 read dummy length Set the dummy length to device 1 (unit = SCLK cycle)

57.4.1.15 Memory map read write command register 0 (MRWCR0)

Set memory map read/write command for device 0.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 48_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MMWC1[7:0]								MMWC0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MMRC1[7:0]								MMRC0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.31 MRWCR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	MMWC1[7:0]	Set the memory map write command for device 0
23 to 16	MMWC0[7:0]	
15 to 8	MMRC1[7:0]	Set the memory map read command for device 0
7 to 0	MMRC0[7:0]	

According to flash memory specification, there are SPI, SOPI, and DOPI mode.

In SPI mode, controller will send only command 0 to memory.

In SOPI and DOPI mode, controller will send command 1 first and then send command 0 to memory.

The mode will be defined in CDSR[1:0] for device0, Controller will send the commands to device0 according to CDSR[1:0] (device0_transfer_type).

57.4.1.16 Memory map read write command register 1 (MRWCR1)

Set memory map read/write command for device 1.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + $4C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	MMWC1[7:0]								MMWC0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MMRC1[7:0]								MMRC0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.32 MRWCR1 Register Contents

Bit Position	Bit Name	Function
31 to 24	MMWC1[7:0]	Set the memory map write command for device 1
23 to 16	MMWC0[7:0]	
15 to 8	MMRC1[7:0]	Set the memory map read command for device 1
7 to 0	MMRC0[7:0]	

According to flash memory specification, there are SPI, SOPI, and DOPI mode.

In SPI mode, controller will send only command 0 to memory.

In SOPI and DOPI mode, controller will send command 1 first and then send command 0 to memory.

The mode will be defined in CDSR[1:0] for device0, Controller will send the commands to device0 according to CDSR[1:0] (device0_transfer_type).

57.4.1.17 Memory map read write command setting register (MRWCSR)

Set memory map read/write command setting.

Access: This register can be read/written in 32-bit units.

Address: <OCTAn_base> + 50_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	WDO1	MMWCL1[2:0]			MMWAL1[2:0]			PRBE1	RDO1	MMRCL1[2:0]			MMRAL1[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	WDO0	MMWCL0[2:0]			MMWAL0[2:0]			PRBE0	RDO0	MMRCL0[2:0]			MMRAL0[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 57.33 MRWCSR Register Contents

Bit Position	Bit Name	Function
31	Reserved	Reading these bits return the initial value.
30	WDO1	Device 1 write data order for memory map write in DOPI mode. In SOPI/SPI mode, write data order WDO1 is don't care. 0: data order receive from/send to device 1 is byte0, byte1, byte2, byte3 1: data order receive from/send to device 1 is byte1, byte0, byte3, byte2
29 to 27	MMWCL1[2:0]	Set the memory map write command length for device 1
26 to 24	MMWAL1[2:0]	Set the memory map write address length for device 1
23	PRBE1	Preamble bit enable for mem1 memory-map read
22	RDO1	Device 1 read data order for memory map read in DOPI mode. It also determines the device 1 auto-calibration read data order. In SOPI/SPI mode, read data order RDO1 is don't care. 0: data order receive from/send to device 1 is byte0, byte1, byte2, byte3 1: data order receive from/send to device 1 is byte1, byte0, byte3, byte2
21 to 19	MMRCL1[2:0]	Set the memory map read command length for device 1
18 to 16	MMRAL1[2:0]	Set the memory map read address length for device 1
15	Reserved	Reading these bits return the initial value.
14	WDO0	Device 0 write data order for memory map write in DOPI mode. In SOPI/SPI mode, write data order WDO0 is don't care. 0: data order receive from/send to device 1 is byte0, byte1, byte2, byte3 1: data order receive from/send to device 1 is byte1, byte0, byte3, byte2
13 to 11	MMWCL0[2:0]	Set the memory map write command length for device 0
10 to 8	MMWAL0[2:0]	Set the memory map write address length for device 0
7	PRBE0	Preamble bit enable for mem1 memory-map read
6	RDO0	Device 0 read data order for memory map read in DOPI mode. It also determines the device 0 auto-calibration read data order. In SOPI/SPI mode, read data order RDO0 is don't care. 0: data order receive from/send to device 1 is byte0, byte1, byte2, byte3 1: data order receive from/send to device 1 is byte1, byte0, byte3, byte2
5 to 3	MMRCL0[2:0]	Set the memory map read command length for device 0
2 to 0	MMRAL0[2:0]	Set the memory map read address length for device 0

57.4.1.18 Error status register (ESR)

Show the memory-map write and memory-map read error status if there is any error while executing memory-map write or memory-map read command.

Access: This register can be read in 32-bit units.

Address: <OCTAn_base> + 54_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MWES[7:0]								MRES[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 57.34 ESR Register Contents

Bit Position	Bit Name	Function
31 to 16	Reserved	Reading these bits return the initial value.
15 to 8	MWES[7:0]	Memory-mapped write error status 80_H: invalid command 01_H: write data length error
7 to 0	MRES[7:0]	Memory-mapped read error status 80_H: invalid command 01_H: ECC error 02_H: preamble error 03_H: wait DQS timeout

ESR content will be auto cleared to default value if user read any configuration register.

57.4.1.19 Configure write without data register (CWNDR)

Issue write command without data.

Access: This register can be written in 32-bit units.

Address: <OCTAn_base> + 58_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 57.35 CWNDR Register Contents

Bit Position	Bit Name	Function
31 to 0	Reserved	When user writes to this register, controller will issue a write command without data (data length = 0) to memory. The issued command bytes are defined in DCR. The issued address bytes are defined in DAR. The command settings are defined in DCSR. Reading these bits return the initial value.

57.4.1.20 Configure write data register (CWDR)

Configure write data and issue write command.

Access: This register can be written in 32-bit units.

Address: <OCTAn_base> + $5C_H$

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DB3[7:0]								DB2[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DB1[7:0]								DB0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 57.36 CWDR Register Contents

Bit Position	Bit Name	Function
31 to 24	DB3[7:0]	When user writes 4 bytes data to this address, controller will issue a write command to memory with the data. The issued command bytes are defined in DCR. The issued address bytes are defined in DAR. The command settings are defined in DCSR. According to CDSR[8] (Device 0 data order) and CDSR[9] (Device 1 data order), controller will decide the byte order sent to memory.
23 to 16	DB2[7:0]	
15 to 8	DB1[7:0]	
7 to 0	DB0[7:0]	

57.4.1.21 Configure read register (CRR)

Read register for configuration read.

Access: This register can be read in 32-bit units.

Address: <OCTAn_base> + 60_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DB3[7:0]								DB2[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DB1[7:0]								DB0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 57.37 CRR Register Contents

Bit Position	Bit Name	Function
31 to 24	DB3[7:0]	When user “read” this address, controller will issue a read command to memory and response the data to user. The issued command bytes are defined in DCR. The issued address bytes are defined in DAR. The command settings are defined in DCSR. According to CDSR[8] (Device 0 data order) and CDSR[9] (Device 1 data order), controller will decide the byte order received from memory. User needs to check the read data length configuration greater than zero. (DCSR[7:0]). If the read data length (DCSR[7:0]) equals to zero, controller will return BUSERR.
23 to 16	DB2[7:0]	
15 to 8	DB1[7:0]	
7 to 0	DB0[7:0]	

57.4.1.22 Auto-calibration status register (ACSR)

The auto-calibration status register provides the calibration status until next auto-calibration.

Access: This register can be read in 32-bit units.

Address: <OCTAn_base> + 64_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	ACSD1[2:0]			ACSD0[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 57.38 ACSR Register Contents

Bit Position	Bit Name	Function
31 to 6	Reserved	Reading these bits return the initial value.
5 to 3	ACSD1[2:0]	Keep information of auto-calibration status of device 1. This information is the status of last auto-calibration.
2 to 0	ACSD0[2:0]	Keep information of auto-calibration status of device 0. This information is the status of last auto-calibration.

Status information:

- 000_B: Fine normal ⇒ ACRR[15:0] represent fine tune result
- 001_B: Fine low bound ⇒ ACRR [31:0]: fine tune result for two fine tune code (Original, Low bound)
- 010_B: Fine high bound ⇒ ACRR [31:0]: fine tune result for two fine tune code (High bound, Original)
- 011_B: Coarse tune plus fine tune search ⇒ ACRR [15:0]: fine tune result, ACRR [31:16]: coarse tune result
- 100_B: No result ⇒ ACRR [31:0]: all zero or don't care

57.4.1.23 Auto-calibration result register 0 (ACRR0)

The auto-calibration result register provides the calibration result of device 0 with latest calibration. If user disables auto-calibration, this value also keeps unchanged until next auto-calibration.

Access: This register can be read in 32-bit units.

Address: <OCTAn_base> + 68_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCRD0[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFRD0[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 57.39 ACRR0 Register Contents

Bit Position	Bit Name	Function
31 to 16	CCRD0[31:16]	Auto-calibration fine result of device 0
15 to 0	CFRD0[15:0]	Auto-calibration coarse result of device 0. Coarse result only update if the fine calibration stage got all failed. (0000_H)

During auto-calibration, this register will be updated if any periodically auto-calibration sequence is finished. (CDSR[13:12] = 01_B)

Example:
When auto-calibration fine result is ACRR[15:0] = $3FFC_H$ = 0011 1111 1111 1100_B, user can know:

- ACRR[1:0] are 0: When set delay line configuration to 0/1, read result are failed.
- ACRR[13:2] are 1: When set delay line configuration to 2~13, read result are passed
- ACRR[15:14] are 0: When set delay line configuration to 14/15, read result are failed.

Therefore, the best configuration will be 7 ((2+13)/2 = 7, middle of passed range).
This configuration will also update to MDTR[7:0] after auto-calibration is finished.

When delay-line setting to stage 0/1, read result is failed. Stage 2~14, read passed and stage 15/16 is failed.

The best configuration will be stage 8 (middle of passed range)

57.4.1.24 Auto-calibration result register 1 (ACRR1)

The auto-calibration result register provides the calibration result of device 1 with latest calibration. If user disables auto-calibration, this value also keeps unchanged until next auto-calibration.

Access: This register can be read in 32-bit units.

Address: <OCTAn_base> + $6C_H$

Initial value: $0000\ 0000_H$

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CCRD1[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CFRD1[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 57.40 ACRR1 Register Contents

Bit Position	Bit Name	Function
31 to 16	CCRD1[31:16]	Auto-calibration fine result of device 1
15 to 0	CFRD1[15:0]	Auto-calibration coarse result of device 1. Coarse result only update if the fine calibration stage got all failed. (0000_H)

During auto-calibration, this register will be updated if any periodically auto-calibration sequence is finished. (CDSR[13:12] = 01_B)

Example

When auto-calibration fine result is ACRR[15:0] = $3FFC_H$ = 0011 1111 1111 1100-, user can know:

- ACRR[1:0] are 0: When set delay line configuration to 0/1, read result are failed.
- ACRR[13:2] are 1: When set delay line configuration to 2~13, read result are passed
- ACRR[15:14] are 0: When set delay line configuration to 14/15, read result are failed.

Therefore, the best configuration will be 7 ((2+13)/2 = 7, middle of passed range).
This configuration will also update to MDTR[7:0] after auto-calibration is finished.

When delay-line setting to stage 0/1, read result is failed. Stage 2~14, read passed and stage 15/16 is failed.

The best configuration will be stage 8 (middle of passed range)

57.4.1.25 Auto-calibration all scan result register 0 (ACASRR0)

The auto-calibration all scan result register 0 provides the auto-calibration scan result of device 0. Before reading ACASRR0, user should set the CDSR[13:12] to 10_B and wait for the CDSR[13:12] change to 00_B.

Access: This register can be read in 32-bit units.

Address: <OCTAn_base> + 70_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LPTCSD0[7:0]								FPTCSD0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LPTCD0[7:0]								FPTCD0[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 57.41 ACASRR0 Register Contents

Bit Position	Bit Name	Function
31 to 24	LPTCSD0[7:0]	After finishing auto-calibration scan (CDSR[13:12] = 10_B), the passed trim code period could be break into two parts. In this case, The last passed trim code of second consecutive passed trim code period will be stored in ACASRR0[31:24]. If the passed trim code period only have one part, ACASRR0[31:24] will be set to 0.
23 to 16	FPTCSD0[7:0]	After finishing auto-calibration scan (CDSR[13:12] = 10_B), the passed trim code period could be break into two parts. In this case, The first passed trim code of second consecutive passed trim code period will be stored in ACASRR0[23:16]. If the passed trim code period only have one part, ACASRR0[23:16] will be set to 0.
15 to 8	LPTCD0[7:0]	After finishing auto-calibration scan (CDSR[13:12] = 10_B), the passed trim code period could be break into two parts. In this case, The last passed trim code of first consecutive passed trim code period will be stored in ACASRR0[15:8]. If the passed trim code period only have one part, the last passed trim code will be stored in ACASRR0[15:8].
15 to 0	FPTCD0[7:0]	After finishing auto-calibration scan (CDSR[13:12] = 10_B), the passed trim code period could be break into two parts. In this case, The first passed trim code of first consecutive passed trim code period will be stored in ACASRR0[7:0]. If the passed trim code period only have one part, the first passed trim code will be stored in ACASRR0[7:0].

57.4.1.26 Auto-calibration all scan result register 1 (ACASRR1)

The auto-calibration all scan result register 1 provides the auto-calibration scan result of device 1. Before reading ACASRR0, user should set the CDSR[13:12] to 10_B and wait for the CDSR[13:12] change to 00_B.

Access: This register can be read in 32-bit units.

Address: <OCTAn_base> + 74_H

Initial value: 0000 0000_H

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	LPTCSD1[7:0]								FPTCSD1[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	LPTCD1[7:0]								FPTCD1[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 57.42 ACASRR1 Register Contents

Bit Position	Bit Name	Function
31 to 24	LPTCSD1[7:0]	After finishing auto-calibration scan (CDSR[13:12] = 10_B), the passed trim code period could be break into two parts. In this case, The last passed trim code of second consecutive passed trim code period will be stored in ACASRR1[31:24]. If the passed trim code period only have one part, ACASRR1[31:24] will be set to 0.
23 to 16	FPTCSD1[7:0]	After finishing auto-calibration scan (CDSR[13:12] = 10_B), the passed trim code period could be break into two parts. In this case, The first passed trim code of second consecutive passed trim code period will be stored in ACASRR1[23:16]. If the passed trim code period only have one part, ACASRR1[23:16] will be set to 0.
15 to 8	LPTCD1[7:0]	After finishing auto-calibration scan (CDSR[13:12] = 10_B), the passed trim code period could be break into two parts. In this case, The last passed trim code of first consecutive passed trim code period will be stored in ACASRR1[15:8]. If the passed trim code period only have one part, the last passed trim code will be stored in ACASRR1[15:8].
15 to 0	FPTCD1[7:0]	After finishing auto-calibration scan (CDSR[13:12] = 10_B), the passed trim code period could be break into two parts. In this case, The first passed trim code of first consecutive passed trim code period will be stored in ACASRR1[7:0]. If the passed trim code period only have one part, the first passed trim code will be stored in ACASRR1[7:0].

57.5 Reference operation flow

In the following content shows the reference operation flow for different purposes.

We set the following settings for the examples stated in this chapter.

- memory connection is device 0: flash; device 1: RAM.

57.5.1 Controller initialization

Step#	Read / Write	Address (Register)	Data	Data Length (byte)	Description
1	Write	DSR0	0x08000000	4	Set Device 0 Size
2	Write	DSR1	0x40800000	4	Set Device 1 Size
3	Write	MDTR	0x00320032	4	Set DQS delay trim value
4	Write	DRCSTR	0x11201120	4	Set CS pin timing for mem-map read
5	Write	DWCSTR	0x1a001a00	4	Set CS pin timing for mem-map write
6	Write	DCSTR	0x00001a00	4	Set CS pin timing
7	Write	CDSR	0x00000008	4	Set Controller Mode (1) Pre-cycle on/off (2) Transfer type selection SPI/STR/DTR
8	Write	MDLR	0x05050000	4	Set dummy cycle length
9	Write	MRWCR0	0x12edec13	4	Set command byte content for device0
10	Write	MRWCR1	0x2000a000	4	Set command byte content for device1
11	Write	MRWCSR	0x54545454	4	Set mem-map command mode (1) Command length (2) Address length (3) Data byte order

57.5.2 Basic operations

The following shows some basic operations which will be used later several times

57.5.2.1 For OctaFlash

(1) Turn on write enable (SPI)

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x00000006	4	
2	Write	DCSR	0x00100000	4	
3	Write	CWNDR	xx	4	

(2) Turn on write enable (OPI)

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x000006f9	4	
2	Write	DCSR	0x00200000	4	
3	Write	CWNDR	xx	4	

(3) Read status register (SPI)

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x00000005	4	
2	Write	DCSR	0x00100001	4	
3	Read	CRR	{23'b0, SR}	1	

(4) Read status register (OPI)

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x000005fa	4	
2	Write	DAR	0x00000000	4	
3	Write	DCSR	0x0c200401	4	
4	Read	CRR	{23'b0, SR}	1	

(5) Read configuration register (SPI)

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x00000015	4	
2	Write	DCSR	0x00100001	4	
3	Read	CRR	{23'b0, CR}	1	

(6) Read configuration register (OPI)

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x000015ea	4	
2	Write	DAR	0x00000001	4	
3	Write	DCSR	0x0c200401	4	
4	Read	CRR	{23'b0, CR}	1	

(7) Write configuration register (OPI)

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x000001fe	4	
2	Write	DAR	0x00000001	4	
3	Write	DCSR	0x04200401	4	
4	Write	CWDR	{23'b0, CR}	1	

(8) Read configuration register 2 (SPI)

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x00000071	4	
2	Write	DAR	Address to flash	4	
3	Write	DCSR	0x04100001	4	
4	Read	CRR	{23'b0, CR2}	1	

(9) Read configuration register 2 (OPI)

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x0000718e	4	
2	Write	DAR	Address to flash	4	
3	Write	DCSR	0x0c200401	4	
4	Read	CRR	{23'b0, CR2}	1	

(10) Write configuration register 2 (SPI)

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x00000072	4	
2	Write	DAR	Address to flash	4	
3	Write	DCSR	0x04100001	4	
4	Write	CWDR	{23'b0, CR2}	1	

(11) Write configuration register 2 (OPI)

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x0000728d	4	
2	Write	DAR	Address to flash	4	
3	Write	DCSR	0x04200001	4	
4	Write	CWDR	{23'b0, CR2}	1	

57.5.2.2 For OctaRAM

(1) Read configuration register

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x0000c000	4	
2	Write	DAR	0x00040000	4	
3	Write	DCSR	0x04a80502	4	
4	Read	CRR	{16'b0, CR}	2	

(2) Write configuration register

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x00004000	4	
2	Write	DAR	0x00040000	4	
3	Write	DCSR	0x04a80002	4	
4	Write	CWDR	{16'b0, CR}	2	

57.5.3 Set dummy cycle

57.5.3.1 For OctaFlash

1. Read Configuration Register 2 (SPI/OPI)
   (Step 1 can be ignored if you don't care about the original data in CR2. Therefore, you should read the content of CR2 first.)
2. Turn on Write Enable (SPI/OPI)
3. Read Status Register (repeat until SR[1]=1)
4. Write Configuration Register 2 (SPI/OPI): addr = 0x00000300, CR2[2:0] = 0x05
   (dummy_cycle = 10, CR2[2:0] = 20-dummy_cycle*2)
5. Read Status Register (repeat until SR[0 = 0)
6. Set MDLR accordingly

57.5.3.2 For OctaRAM

1. Read Configuration Register
   (Step 1 can be ignored if you don't care about the original data in CR. Therefore, you should read the content of CR first.)
2. Write Configuration Register: CR[7:4] = 0x02 (dummy_cycle = 5, CR[7:4] = dummy_cycle-3)
3. Set MDLR accordingly

57.5.4 Set Flash Transfer Type

1. Read Configuration Register 2 (SPI/OPI)
   (Step 0 can be ignored if you don't care about the original data in CR2. Therefore, you should read the content of CR2 first.)
2. Turn on Write Enable (SPI/OPI)
3. Read Status Register (repeat until SR[1] = 1)
4. Write Configuration Register 2 (SPI/OPI): addr = 0000 0000_H, CR2[1:0] = 02_H
   (00_H: SPI; 01_H: ST; 02_H: DTR)
5. Read Status Register (repeat until SR[0] = 0)
6. Set CDSR accordingly

57.5.5 Auto-calibration setting

1. Set ACTR (timer for controlling auto-calibration period)
2. Set ACAR0 (ac address for device0)
3. Set ACAR1 (ac address for device1)

57.5.5.1 For OctaFlash

1. Turn on Write Enable (SPI/OPI)
2. Read Status Register (repeat until SR[1] = 1)
3. write preamble pattern to addr = ACAR0

4. Read Status Register (repeat until SR[0] = 0)
5. Set CDSR accordingly

57.5.5.2 For OctaRAM

1. write preamble pattern to addr = ACAR1
2. Set CDSR accordingly

If CDSR[23] equals 1, the default preamble pattern is 0xFFFF0000 000800FF 00FFF700 F700F708 (Order: Byte[15] … Byte[0]).

57.5.6 Manual calibration

This section introduces manual calibration sequence in the beginning stage.

The steps for manual calibration are shown in the table below.

Table 57.43 Manual calibration steps

Manual Calibration Sequence			1st boot time		Every boot time (after 1st boot)	
			OctaFlash*1	OctaRAM*2	OctaFlash*3	OctaRAM*4
Prepare for calibration	1.	User programs calibration data to specified address area in OCTA device	V (in SPI Mode)	V	Optional (in SPI Mode)	V
	2.	User uses software to read the specified address area from OCTA device, and check the correctness				
Manual calibration	1.	User uses software to read the specified address area from OCAT device, and check the correctness on DQS delay set to 0 (MDTR[7:0])	V	V	V	V
	2.	Increment 1 to MDTR[7:0] value and perform read access from OCTA device (specified address area). It continue to until MDTR[7:0]=0xFF.				
	3.	User uses software to check the correctness of read data for each MDTR[7:0] value.				
	4.	User uses software to confirm the consecutive pass area for MDTR[7:0] value.				
	5.	User uses software to choose biggest pass window and select min pass value and max pass value.				
	6.	User uses software to calculate the average value from min / max pass value ((min value + max value) / 2)				
	7.	User uses software to set this average value to MDTR[7:0] register.				

Note 1. For the detailed sequence please refer to **Table 57.44**.
Note 2. For the detailed sequence please refer to **Table 57.47**.
Note 3. For the detailed sequence please refer to **Table 57.45**.
Note 4. For the detailed sequence please refer to **Table 57.48**.

For above steps, there are related registers need to be used as below:

(1) Set the DSR0 and DSR1 registers for the mounted devices.

(2) Set the ACAR0 and ACAR1 for programming the pattern.

(3) Set the ACAR0 and ACAR1 for the start address of calibration 16-byte data.

(4) Set the MDTR[23:16] for device 1 and MDTR[7:0] for device 0 to control DQS delay line and enable DQS delay.

Manual Calibration Algorithm Description:

While doing manual calibration, the manual calibration flow can be divided into two parts, coarse part and fine part.

The example of setting MDTR[7:0] for manual calibration is shown in the below. User can reference the following steps to set MDTR[23:16]. In coarse part, users need to read calibration data 16 times at first and choose the maximum and minimum in the longest consecutive successful read interval for the delay line control as shown in **Figure 57.3**. The numbers in **Figure 57.3** represent the DQS delay line control (MDTR[7:4]) from 0 to 7 in coarse part. Yellow represents successfully read the calibration data. Red represents fail to read the calibration data. If there are more than one interval and their lengths are equal, users just select one of them for the delay line control and set the delay line control (Maximum + Minimum)/2. Here, the Maximum and Minimum are the maximal and minimal DQS delay line control of the selected interval. After deciding the highest 4 bit of DQS delay line control, users utilize the same way to determine the lowest 4 bit (MDTR[3:0]) of DQS delay line control as shown in **Figure 57.3**.

If all the combinations of 8-bit DQS delay line control are failed, user should enter error handle flow to handle this condition. For example, reset memory controller and memory devices, check whether the settings of controller registers are correct and re-run manual calibration again.

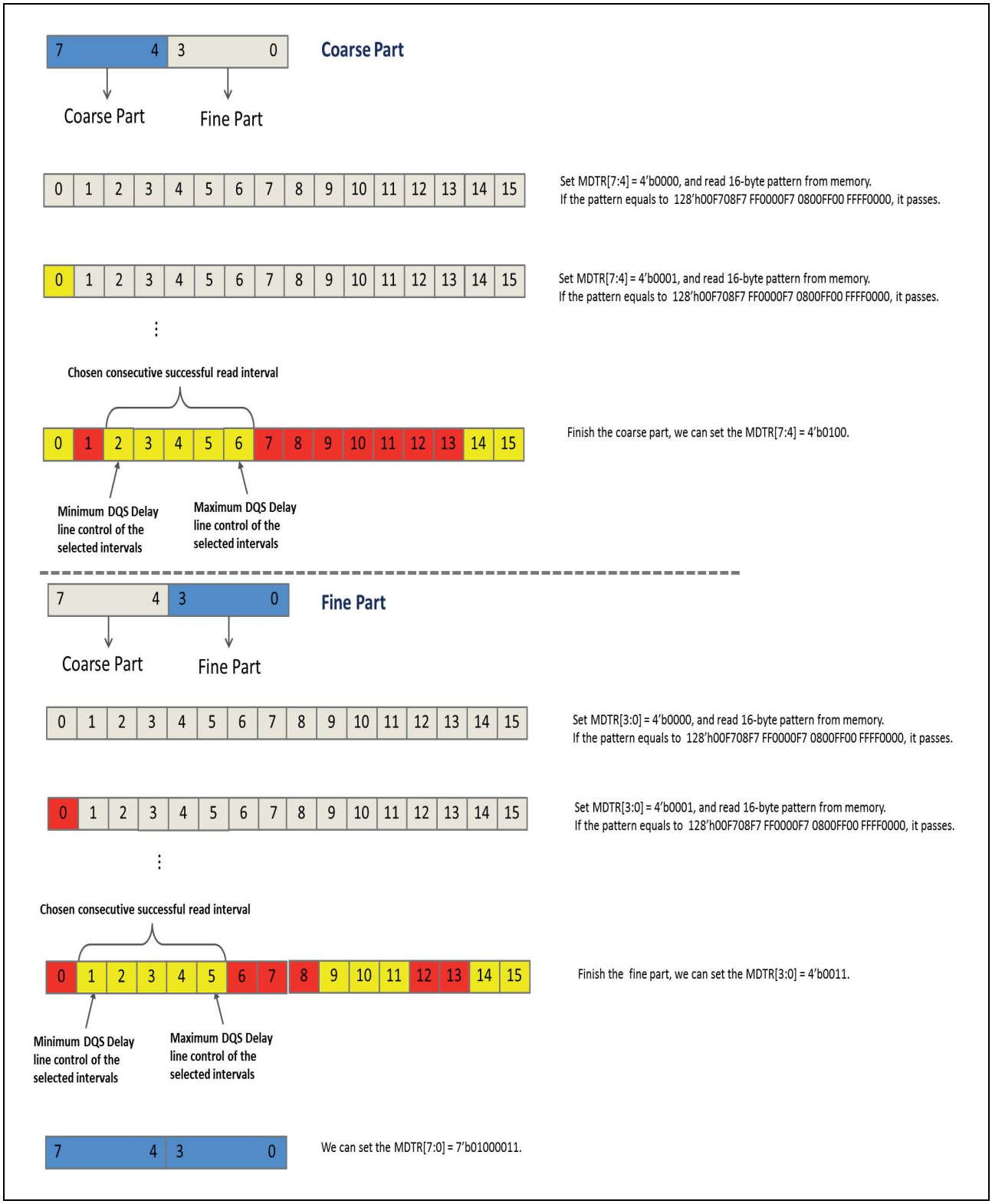


Figure 57.3 An example for setting delay line control value of MDTR[7:0] for device 0

Table 57.44 Preparation for manual calibration sequences on 1st boot time of OctaFlash (when device 0 connected flash)

Read / Write	Register	Data	Data Length (byte)	Description
N/A	N/A	N/A	N/A	Wait tVSL (3000 µs) before OctaFlash is ready to access
Do controller initialization, please reference Section 57.5.1, Controller initialization. (User should set the initial setting according to the actual situation.)				
Set the flash device in SPI mode, please reference Section 57.5.4, Set Flash Transfer Type.				
Set the Memory-Map Related Command and Timing, please reference Section 57.5.1, Controller initialization.				
Write	ACAR0	address	4	Set the start address of manual calibration pattern
Memory-Map Read	0x6000_0000 + ACAR0	N/A	16	Read the data from address
1. Compare whether the 16 bytes *data* == 0xFFFF FFFF FFFF FFFF? 2. If no, erase the data in the *address* and do the following steps.				
Memory-Map Write	0x6000_0000 + ACAR0	preamble pattern	16	Program the preamble pattern for manual/auto calibration
1. Wait tPP for flash programming data. (Please reference the Flash data sheet for tPP). 2. Repeat reading the status register (SR) of OctaFlash until SR[0] == 0.				
Write	DCR	0x00000005	4	
Write	DCSR	0x00100001	4	
Read	CRR	{23'b0, SR}	1	
Program Ready				
Memory-Map Read	0x6000_0000 + ACAR0	N/A	16	Read the data from address
Software compares the preamble patterns and the data read from flash. If they are not the same, repeat above actions again.				

Table 57.45 Manual calibration sequences on boot time of OctaFlash (1st or every time chosen by users; when device 0 connected flash)

Read / Write	Register	Data	Data Length (byte)	Description
N/A	N/A	N/A	N/A	Wait tVSL(3000us) before OctaFlash is ready to access
Do controller initialization, please reference Section 57.5.1, Controller initialization. (User should set the initial setting according to the actual situation.)				
Set the flash device in DOPI mode, please reference Section 57.5.4, Set Flash Transfer Type.				
Determine the highest 4-bit of DQS delay line control. Start: 1. Repeat the following actions 16 times 2. The *delay* = 0x0 ~ 0xF				
Write	MDTR	[31:8] = 0x0 [7:4] = *delay* [3:0] = 0x0	4	Set the memory delay trim register
Memory-Map Read	0x6000_0000 + ACAR0	N/A	16	Read the preamble pattern
Software compare the preamble patterns read from OctaFlash.				
End: Software confirms the consecutive pass area of MDTR value and does 1. Fetch the maximum delay and minimum delay in the longest consecutive successful read interval. 2. Let the *d_coarse* = (maximum+minimum)/2, and set the MDTR[7:4] = *d_coarse.* 3. Software sets the *d_coarse* to MDTR[7:4]. Else Determine the 8-bit DQS delay line in the following steps, please skip to ***Table 57.46***				
Write	MDTR	[31:8] = 0x0 [7:4] = *d_coarse* [3:0] = 0x0	4	Set the memory delay trim register
Determine the lowest 4-bit of DQS delay line control. Start: 1. Repeat the following actions 16 times 2. The *delay* = 0x0 ~ 0xF				
Write	MDTR	[31:8] = 0x0 [7:4] = *d_coarse* [3:0] = *delay*	4	Set the memory delay trim register
Memory-Map Read	0x6000_0000 + ACAR0	N/A	16	Read the preamble pattern
Software compare the preamble patterns read from OctaFlash.				
End: Software confirms the consecutive pass area of MDTR value and does 1. Fetch the maximum delay and minimum delay in the longest consecutive successful read interval. 2. Let the *d_fine* = (maximum+minimum)/2, and set the MDTR[3:0] = *d_fine.* 3. Software sets the *d_fine* to MDTR[3:0].				
Write	MDTR	[31:8] = 0x0 [7:4] = *d_coarse* [3:0] = *d_fine*	4	Set the memory delay trim register
Reference Auto-calibration setting in Section 57.5.5, Auto-calibration setting.				

Table 57.46 Manual calibration sequences of OctaFlash (When there is no suitable highest 4-bit of delay line control)

Read / Write	Register	Data	Data Length (byte)	Description
Start: 1. Repeat the following actions 256 times 2. The *delay* = 0x00 ~ 0xFF				
Write	MDTR	[31:8] = 0x0 [7:0] = *delay*	4	Set the memory delay trim register
Memory-Map Read	0x6000_0000 + ACAR0	N/A	16	Read the preamble pattern
Software compare the preamble patterns read from OctaFlash.				
End: Software confirms the consecutive pass area of MDTR value and does 1. Fetch the maximum delay and minimum delay in the longest consecutive successful read interval. 2. Let the *dqs_delay* = (maximum+minimum)/2, and set the MDTR[7:0] = *dqs_delay.* 3. Software sets the *dqs_delay* to MDTR[7:0].				
Write	MDTR	[31:8] = 0x0 [7:0] = *dqs_delay*	4	Set the memory delay trim register
Reference Auto-calibration setting in Section 57.5.5, Auto-calibration setting.				

Table 57.47 Preparation for Manual calibration sequences on 1st boot time of OctaRAM (when device 0 connected RAM)

Read / Write	Register	Data	Data Length (byte)	Description
N/A	N/A	N/A	N/A	Wait tPU (150 µs) before RAM is ready to access
Do controller initialization, please reference Section 57.5.1, Controller initialization. (User should set the initial setting according to the actual situation.)				
Write	ACAR0	address	4	Set the start address of manual calibration pattern
Memory-Map Read	0x6000_0000 + ACAR0	N/A	16	Read the data from address
1. Compare whether the 16 bytes *data* == 0xFFFF FFFF FFFF FFFF? 2. If no, erase the data in the *address* and do the following steps.				
Memory-Map Write	0x6000_0000 + ACAR0	preamble pattern	16	Program the preamble pattern for manual/auto calibration
Memory-Map Read	0x6000_0000 + ACAR0	N/A	16	Read the data from address
Software compares the preamble patterns and the data read from flash. If they are not the same, repeat above actions again.				

Table 57.48 Manual calibration sequences on boot time of OctaRAM (1st or every time chosen by users; when device 0 connected RAM)

Read / Write	Register	Data	Data Length (byte)	Description
N/A	N/A	N/A	N/A	Wait tPU(150us) before RAM is ready to access
Do controller initialization, please reference Section 57.5.1, Controller initialization. (User should set the initial setting according to the actual situation.)				
Determine the highest 4-bit of DQS delay line control. Start: 1. Repeat the following actions 16 times 2. The *delay* = 0x0 ~ 0xF				
Write	MDTR	[31:8] = 0x0 [7:4] = *delay* [3:0] = 0x0	4	Set the memory delay trim register
Memory-Map Read	0x6000_0000 + ACAR0	N/A	16	Read the preamble pattern
Software compare the preamble patterns read from OctaRAM				
End: Software confirms the consecutive pass area of MDTR value and does 1. Fetch the maximum delay and minimum delay in the longest consecutive successful read interval. 2. Let the *d_coarse* = (maximum+minimum)/2, and set the MDTR[7:4] = *d_coarse.* 3. Software sets the *d_coarse* to MDTR[7:4]. Else Determine the 8-bit DQS delay line in the following steps, please skip to **Table 57.49**.				
Write	MDTR	[31:8] = 0x0 [7:4] = *d_coarse* [3:0] = 0x0	4	Set the memory delay trim register
Determine the lowest 4-bit of DQS delay line control. Start: 1. Repeat the following actions 16 times 2. The *delay* = 0x0 ~ 0xF				
Write	MDTR	[31:8] = 0x0 [7:4] = *d_coarse* [3:0] = *delay*	4	Set the memory delay trim register
Memory-Map Read	0x6000_0000 + ACAR0	N/A	16	Read the preamble pattern
Software compare the preamble patterns read from OctaRAM.				
End: Software confirms the consecutive pass area of MDTR value and does 1. Fetch the maximum delay and minimum delay in the longest consecutive successful read interval. 2. Let the *d_fine* = (maximum+minimum)/2, and set the MDTR[3:0] = *d_fine.* 3. Software sets the *d_fine* to MDTR[3:0].				
Write	MDTR	[31:8] = 0x0 [7:4] = *d_coarse* [3:0] = *d_fine*	4	Set the memory delay trim register
Reference Auto-calibration setting in Section 57.5.5, Auto-calibration setting.				

Table 57.49 Table 54. Manual calibration sequences of OctaRAM (When there is no suitable highest 4-bit of delay line control)

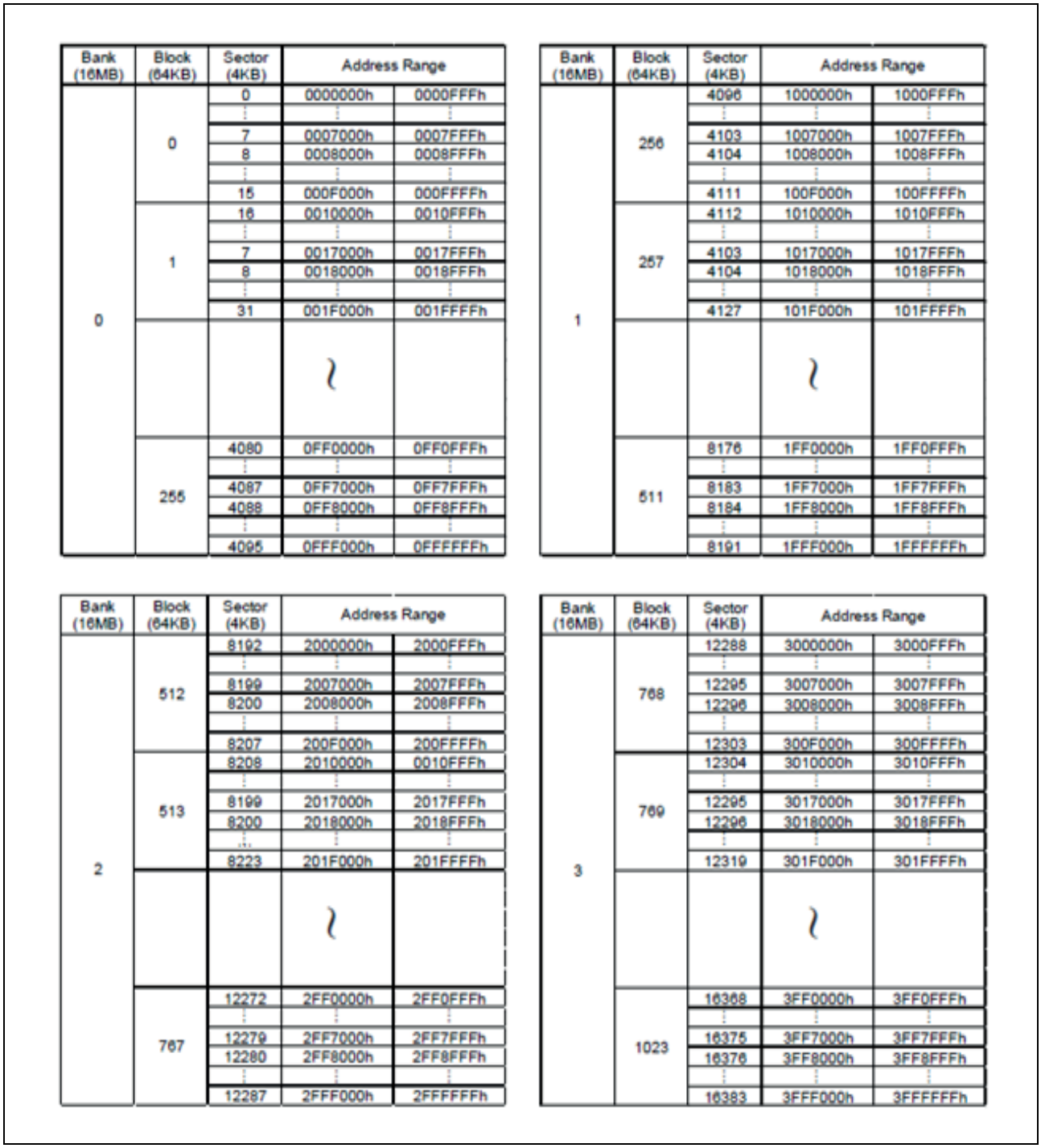
Read / Write	Register	Data	Data Length (byte)	Description
Start: 1. Repeat the following actions 256 times 2. The *delay* = 0x00 ~ 0xFF				
Write	MDTR	[31:8] = 0x0 [7:0] = *delay*	4	Set the memory delay trim register
Memory-Map Read	0x6000_0000 + ACAR0	N/A	16	Read the preamble pattern
Software compare the preamble patterns read from OctaRAM.				
End: Software confirms the consecutive pass area of MDTR value and does 1. Fetch the maximum delay and minimum delay in the longest consecutive successful read interval. 2. Let the *dqs_delay* = (maximum+minimum)/2, and set the MDTR[7:0] = *dqs_delay.* 3. Software sets the *dqs_delay* to MDTR[7:0].				
Write	MDTR	[31:8] = 0x0 [7:0] = *dqs_delay*	4	Set the memory delay trim register
Reference Auto-calibration setting in Section 57.5.5, Auto-calibration setting.				

57.6 Read while write operation flow

Octa controller supports read-while-write (RWW) feature. The brief introduction of Flash memory which supports RWW function is in Section 57.6.1, Flash memory organization, the operation flow of RWW function is shown in Section 57.6.2, Read-While-Write (RWW) operation flow, and the operation sequence of controller for executing the RWW function is shown in **Section 57.6.3, Controller sequences for Read-While-Write (RWW) function**.

57.6.1 Flash memory organization

The latest MX25LW series Flash memory supports the RWW function. Take the MX25LW51245G-Automotive for example. This Flash memory array is divided into banks. The multi bank structure enables RWW feature, which means read data from one bank while another bank is programing or erasing. The details of the address ranges and the corresponding sector and block addresses are shown in the figure below.



Bank (16MB)	Block (64KB)	Sector (4KB)	Address Range	
0	0	0	0000000h	0000FFFh
		⋮	⋮	⋮
		7	0007000h	0007FFFh
		8	0008000h	0008FFFh
		⋮	⋮	⋮
		15	000F000h	000FFFFh
	1	16	0010000h	0010FFFh
		⋮	⋮	⋮
		7	0017000h	0017FFFh
		8	0018000h	0018FFFh
		⋮	⋮	⋮
		31	001F000h	001FFFFh
	≀	≀	≀	≀
	255	4080	0FF0000h	0FF0FFFh
		⋮	⋮	⋮
		4087	0FF7000h	0FF7FFFh
		4088	0FF8000h	0FF8FFFh
		⋮	⋮	⋮
		4095	0FFF000h	0FFFFFFh

Bank (16MB)	Block (64KB)	Sector (4KB)	Address Range	
1	256	4096	1000000h	1000FFFh
		⋮	⋮	⋮
		4103	1007000h	1007FFFh
		4104	1008000h	1008FFFh
		⋮	⋮	⋮
		4111	100F000h	100FFFFh
	257	4112	1010000h	1010FFFh
		⋮	⋮	⋮
		4103	1017000h	1017FFFh
		4104	1018000h	1018FFFh
		⋮	⋮	⋮
		4127	101F000h	101FFFFh
	≀	≀	≀	≀
	511	8176	1FF0000h	1FF0FFFh
		⋮	⋮	⋮
		8183	1FF7000h	1FF7FFFh
		8184	1FF8000h	1FF8FFFh
		⋮	⋮	⋮
		8191	1FFF000h	1FFFFFFh

Bank (16MB)	Block (64KB)	Sector (4KB)	Address Range	
2	512	8192	2000000h	2000FFFh
		⋮	⋮	⋮
		8199	2007000h	2007FFFh
		8200	2008000h	2008FFFh
		⋮	⋮	⋮
		8207	200F000h	200FFFFh
	513	8208	2010000h	0010FFFh
		⋮	⋮	⋮
		8199	2017000h	2017FFFh
		8200	2018000h	2018FFFh
		⋮	⋮	⋮
		8223	201F000h	201FFFFh
	≀	≀	≀	≀
	767	12272	2FF0000h	2FF0FFFh
		⋮	⋮	⋮
		12279	2FF7000h	2FF7FFFh
		12280	2FF8000h	2FF8FFFh
		⋮	⋮	⋮
		12287	2FFF000h	2FFFFFFh

Bank (16MB)	Block (64KB)	Sector (4KB)	Address Range	
3	768	12288	3000000h	3000FFFh
		⋮	⋮	⋮
		12295	3007000h	3007FFFh
		12296	3008000h	3008FFFh
		⋮	⋮	⋮
		12303	300F000h	300FFFFh
	769	12304	3010000h	3010FFFh
		⋮	⋮	⋮
		12295	3017000h	3017FFFh
		12296	3018000h	3018FFFh
		⋮	⋮	⋮
		12319	301F000h	301FFFFh
	≀	≀	≀	≀
	1023	16368	3FF0000h	3FF0FFFh
		⋮	⋮	⋮
		16375	3FF7000h	3FF7FFFh
		16376	3FF8000h	3FF8FFFh
		⋮	⋮	⋮
		16383	3FFF000h	3FFFFFFh

Figure 57.4 MX25LW51245G-Automotive memory organization

57.6.2 Read-While-Write (RWW) operation flow

The multi-bank read-while-write flash memory provides an interruptible write-to-buffer sequence during programming. This sequence provides the advantage that read operations could be inserted among program data write cycles. There are three steps for the operation: issuing WRBI (Write Buffer Initial command), WRCT (Write Buffer Continue command), and WRCF (Write Confirm command) as shown in the figure below.

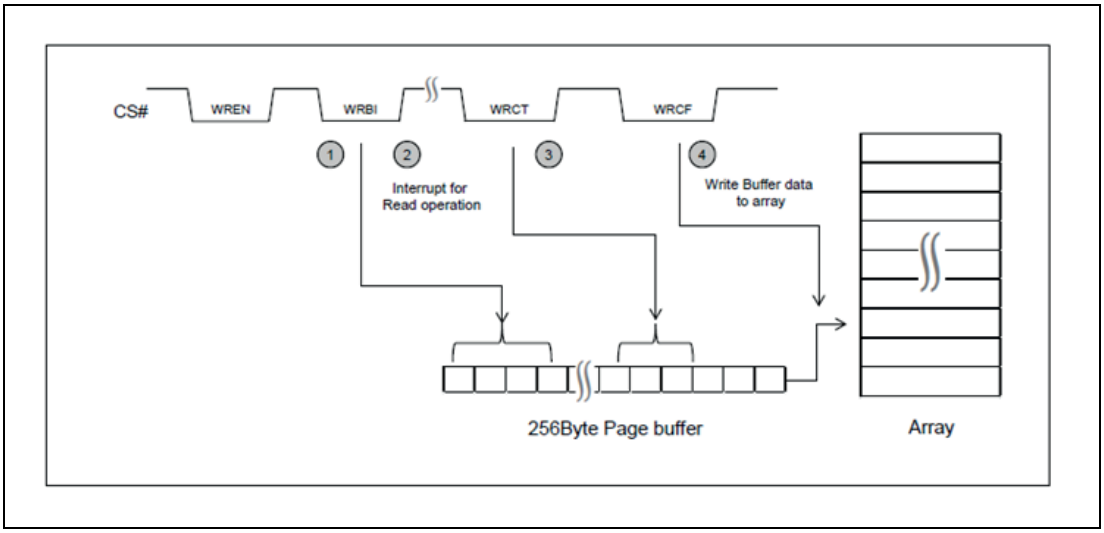


Figure 57.5 Write buffer sequence

To trigger the interruptible write-to-buffer sequence, the system issues WREN command first to enable the WEL bit, and then issues WRBI with 4-byte address and data of 1 ~ 256 byte. After CS# has gone high, the flash memory will return to the standby status, waiting for the next command. The system could either perform read operation or issue Write Buffer Continue command (WRCT) to write more data to the page buffer. After the data writing has finished, the system will issue a Write Confirm (WRCF) command to initiate an automatic program operation to write the page buffer data into the array cell.

Multiple read or WRCT commands can be issued between the WRBI and WRCF commands. The system could also issue a WRBI directly followed by a WRCF command if there is no needs for write interrupt on the system. WRBI command can clear the page buffer area. If a WRBI command is issued following another WRBI command, the data written by the previous WRBI command would be abandoned. Any WRCT or WRCF command without a preceding WRBI command would be ignored. A WRDI, program, erase, write register or reset command would abort the sequence; system has to do WREN and WRBI again to initiate another write-to-buffer sequence.

57.6.3 Controller sequences for Read-While-Write (RWW) function

The reference RWW flow is in the below. This will guarantee the ordering of sending WRBI command and WRCT command.

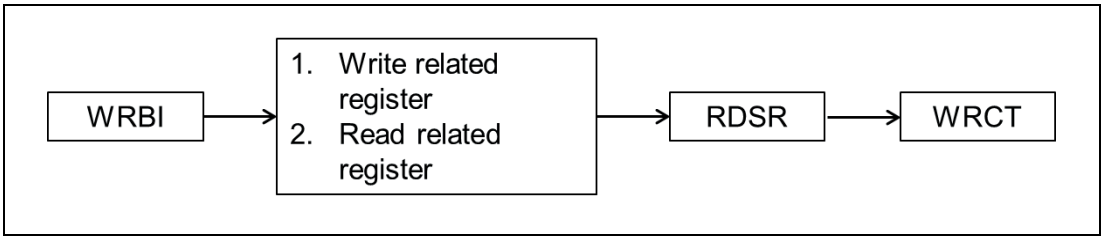


Figure 57.6 RWW commands reference flow

After sending the WRBI command, user needs to write related register then read back related register to make sure register already updated before RDSR. This is for ensuring WRBI before RDSR. Then, user sends RDSR to make sure WRBI ahead of WRCT.

The reference operation sequences of controller for executing RWW function is described in the below table.

After initializing the controller in DOPI mode, the RWW reference operation is described in the below table. The OctaFlash is connected to device 0. The parameters defined in the below are for the controller sequences of RWW function.

address_1: 32-bit address of the Flash.

address_2: address_2[31:8] = address_1[31:8]; address_2[7:0] inside {0x00 ~ 0xFF};

The address_3 is defined as 32-bit address of the Flash which is in another bank.

Step#	Read / Write	Register	Data	Data Length (byte)	Description
1	Write	DCR	0x000006F9	4	Set WREN command
2	Write	DCSR	0x00200000	4	Set DCSR for WREN command
3	Write	CWNDR	0x00000000	4	Send WREN command
4	Write	MRWCR0	0x24DBEE11	4	Set WRCT command for mem-map write
5	Write	DCR	0x000022DD	4	Set WRBI command
6	Write	DAR	address_1	4	Set address to DAR
7	Write	DCSR	0x04A00000	4	Set DCSR for WRBI command
8	Write	CWNDR	0x00000000	4	Send WRBI command
9	Write	DCR	0x000005FA	4	Write related register (RDSR)
10	Write	DAR	0x00000000	4	Set address to DAR
11	Write	DCSR	0x0c200401	4	Set DCSR for RDSR command
12	Read	DCSR	N/A	4	Read DCSR again to confirm the ordering
13	Memory-Map Write	0x6000_0000 + address_2	User_data	User_defined	Memory-Map Write
14*1	Memory-Map Read	0x6000_0000 + address_3	N/A	User_defined	Memory-Map Read
15	Memory-Map Write	0x6000_0000 + address_2	User_data	User_defined	Memory-Map Write
16	Write	DCR	0x000031CE	4	Set WRCF command
17	Write	DCSR	0x00200000	4	Set DCSR for WRCF command
18	Write	CWNDR	0x00000000	4	Send WRCF command

Note 1. User can repeat the memory-map read to another banks without disturbing the RWW program sequence.

57.7 DQS auto-calibration

Although controller has the capability of delay line adjustment by configuration register, the variation of voltage and temperature will affect the delay line / data valid window. These variations may lead to bad latch position and get wrong data from memory device. The periodic auto-calibration (CDSR[13:12] = 01_B) is shown in the below.

The concept of auto-calibration is similar to memory-mapped read/write function. When the calibration timer is time-out, OctaBus Controller generates a series of calibration sequences to read the specific preamble patterns with different delay-line setting and choose the best setting from previous step and

store to configuration register. Section 57.4.1.7, Auto-calibration timer register (ACTR) is the detailed information of auto-calibration timer register (ACTR). Please set reasonable value for the auto-calibration period. When auto-calibration sequence is active, the controller will not process any memory transaction. Before enabling the auto-calibration, user needs to do manual calibration first and set the suitable delay line control value to MDTR. Notice that auto-calibration only supports DOPI mode. If exiting the DOPI mode, please turn off the auto-calibration.

However, to reach such kind of auto-calibration, user needs to reserved 16 byte user area for store preamble pattern. Here, the auto-calibration is implemented by read and write 16-byte data space. User need to set the 16-bytes address for the preamble pattern in ACAR0 and ACAR1 and program the default preamble patterns into the devices.
If MRWCSR[22]/MRWCSR[6] equals 1, the default preamble pattern is
0xF700F708 00FFF700 000800FF FFFF0000 (Order: Byte[15] … Byte[0]).
If MRWCSR[22]/MRWCSR[6] equals 0, the default preamble pattern is
0x00F708F7 FF0000F7 0800FF00 FFFF0000 (Order: Byte[15] … Byte[0]).

Flash device: Program flash with 16 byte preamble pattern to reserved address

RAM device: Write RAM with 16 byte preamble pattern to reserved address

Initial sequence:

Use the manual calibration to obtain the initial sequence.

Auto-calibration sequence:

Auto-calibration timer time-out (Equal to ACTR)

- Arrange to Read/Write transaction order queue (In-order transfer)
- Configure MDTR for different delay setting
  - Read preamble data sequence
  - Check data correctness with preamble pattern (PPR0/PPR1/PPR2/PPR3)
  - Repeat until all valid delay setting is applied (16 stage)
- Choose best delay setting and store to MDTR and update the detailed results in ACASRR0 and ACASRR1
- Restart auto-calibration timer

The auto-calibration flow is shown in Figure 57.7, Auto-calibration flow chart. In the beginning, the auto-calibration start DQS delay line control set up by manual calibration. Here, we define the DQS delay line control (MDTR[23:16] / MDTR[7:0]) as two groups, {Coarse[3:0], Fine[3:0]}.

Step 1: Auto-calibration process utilizes the coarse part and searches all the combinations of fine part.

Step 2: Auto-calibration does low boundary check. If the passed delay line control of fine tune combinations from first step are only between 0000_B to 0011_B, auto-calibration will do step 3.

Step 3: Auto-calibration will minus 1 from coarse part and search all the combinations of its fine part. Then, auto-calibration will find the middle value of the consecutive longest passed DQS control delay line.

Example:
Initial: Manual Calibration Result = 0011_0111_B
Step 1: There are only passed read between 0011_0000_B to 0011_0010_B (low boundary)
Step 2: Determine the low boundary condition.
Step 3: Search 0010_0000_B ~ 0010_1111_B, and find 0010_1000_B to 0010_1111_B are passed. The final delay value is (0011_0111_B + 0010_1000_B)/2 = $0010\_ 1101_B$

Step 4: Auto-calibration does high boundary check. If the passed delay line control of fine tune combinations from first step are only between 1100b to 1111b, auto-calibration will do step 5.

Step 5: Auto-calibration will add 1 to coarse part and search all the combinations of its fine part. Then, auto-calibration will find the middle value of the consecutive longest passed DQS control delay line.

example:
Initial: Manual Calibration Result = 0011_0111_B
Step 1: There are only passed read between 0011_1100_B to 0011_1111_B (low boundary)
Step 2: Determine the low boundary condition and failed.
Step 4: Determine the high boundary condition and passed.
Step 5: Search 0100_0000_B ~ 0100_1111_B, and find 0100_0000_B to 0100_1100_B are passed. The final delay value is (0011_1100b_B + 0100_1100_B)/2 = 0100_0100_B

Step 6: Check whether all the combinations from Step 1 are fail. If it is true, auto-calibration will do Step 7.

Step 7: Scan the coarse part first and get the middle one. Then, scan the fine part with the known coarse part and get the middle one.

Step 8: Select the longest consecutive interval of Step 1 results and get the middle one.

Step 9: Set the MDTR[23:16] / MDTR[7:0]. Finish the auto-calibration process.

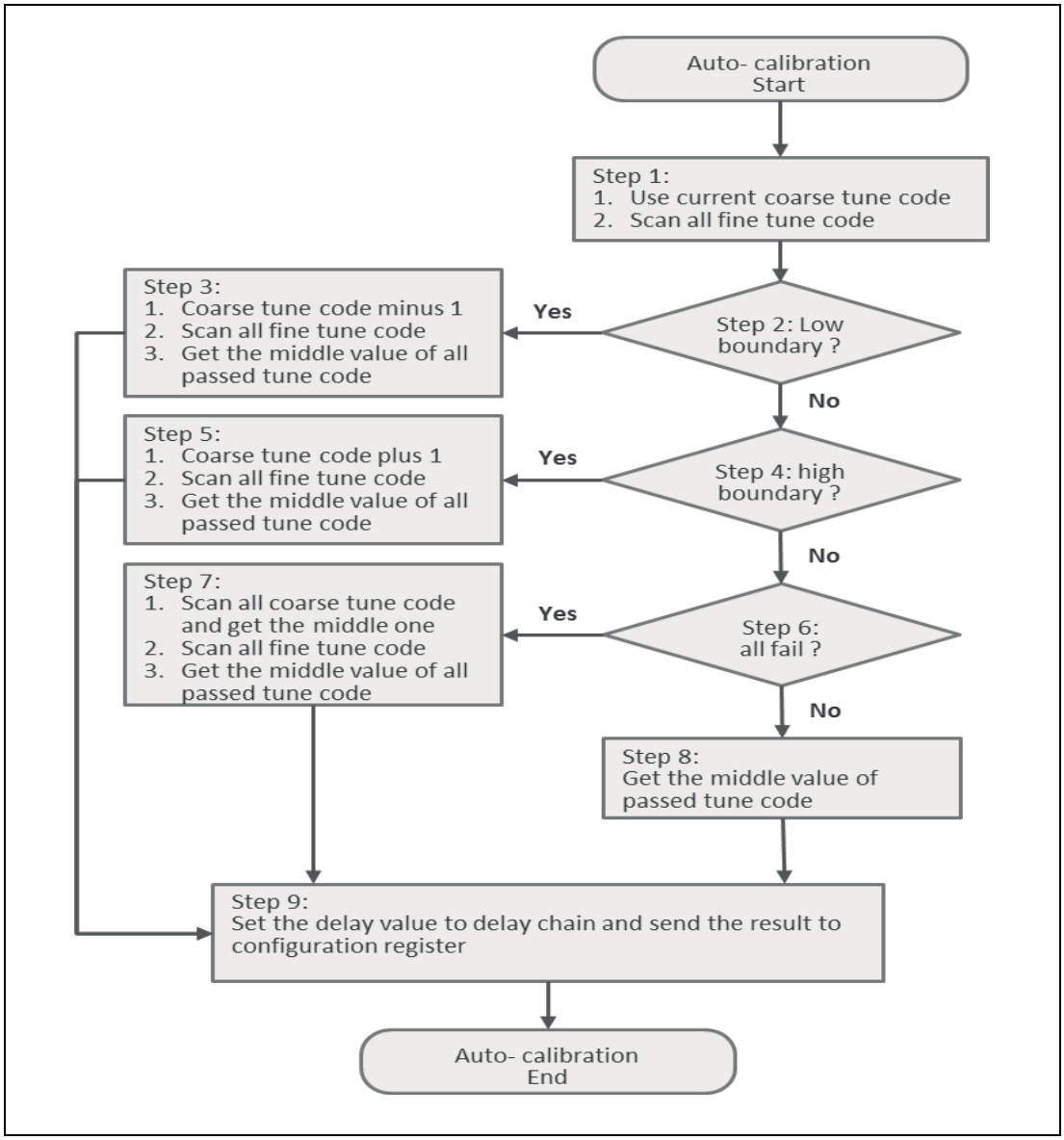


Figure 57.7 Auto-calibration flow chart

Section 58 NAND Flash Memory Interface A (NFMA)

This section contains a generic description of the NAND Flash Memory Interface.

The first section describes all properties specific to the RH850/D1L/D1M, such as units, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

58.1 Overview of the RH850/D1L/D1M NAND Flash Memory Interface (NFMA)

58.1.1 Units

This microcontroller has the following number of units of the NAND Flash Memory Interface.

Table 58.1 Units

NAND Flash Memory Interface	D1L1	D1L2(H)	D1M1	D1M1H	D1M1-V2	D1M1A	D1M2(H)
Units	–	–	–	–	–	1	–
Names	–	–	–	–	–	NFMA0	–

58.1.2 Indices

Following indices are used in this section:

Table 58.2 Indices

Index	Meaning
n	The individual NFMA units are generically indicated by the index “n”.

58.1.3 Register addresses

All NAND Flash Memory Interfaces register addresses are given as address offsets from the individual base addresses <NFMAn_base>.
The <NFMAn_base> addresses of each NFMAn are listed in the following table:

Table 58.3 Register base addresses <NFMAn_base>

NFMAn unit	<NFMAn_base> address
NFMA0	FFDE 2000_H

58.1.4 Clock supply

All NAND Flash Memory Interfaces provide two clock inputs.

Table 58.4 Clock supply

NFMAn unit	NFMAn clock	Connected to
NFMA0	Internal bus clock CLKA	Clock Controller C_ISO_XCCLK
	Operation clock CLKB	Clock Controller C_ISO_XCCLK

58.1.5 Interrupt requests

NFMAn interrupt requests are listed in the following table.

Table 58.5 Interrupt Requests

TPUn signals	Function	Connected to
NFMA0:		
INTNFMA	NFMA0 global interrupt	Interrupt Controller INTNFMA0

58.1.6 Reset sources

The NAND Flash Memory Interfaces and their registers are initialized by the following reset signal:

Table 58.6 Reset sources

NFMAn unit	Reset signal
NFMA0	• Reset Controller SYSRES

58.1.7 External Input/output signals

The following table shows the I/O signals of the NAND Flash Memory Interfaces.

Table 58.7 NFMAn I/O signals

NFMAn signals	Function	Connected to
NFMA0		
CE0#	Chip enable output 0	Port NAND_CS#
CLE	Command latch enable output	Port NAND_CLE
ALE	Address latch enable	Port NAND_ALE
RE#	Read enable output	Port NAND_RE#
WE#	Write enable output	Port NAND_WE#
R0B	Ready/busy inputs	Port NAND_RB0
DATA[7:0]	Data I/O	Port NAND_D[7:0]
DATA[15:8]		Not connected

58.2 Features

The NAND Flash Memory Interface controller implements the function of a high level interface to one NAND flash device.

NOTE

Throughout this chapter the term “system memory” denotes the memories, that are accessible via the NAND Flash Memory I/F master interface unit (MIU). These memories are

- external SDRAM via the SDR-SDRAM Memory Controller (SDRA)
- on-chip Local RAM
- on-chip Video RAMs (VRAM0 and VRAM1)
- external HyperRAM via the HyperBus Controller (HYPB)
- external OctaRAM via the OctaBus Controller (OCTA)

- Compatibility
  - Compatible with the ONFi 1.x specification.
  - Support for the Small Block devices - only devices that allow to disable CE signal when device is in the busy state are supported.
  - Support for the low capacity device (which use four address bytes) with the help of generic sequence.
- DMA Controller
  - Scatter-gather mode is supported.
  - Single transfer mode is supported.
  - Supports precise and imprecise burst modes.
  - Supports configurable burst types: single transfer, incrementing and stream.
  - 32-bit system memory addressing.
  - Triggering mechanism that depends on the programmable FIFO data level.
- Interrupt Controller
  - Each interrupt can be masked.
  - Each interrupt has its own status flag.
  - The status flags are also valid when the given interrupt is masked, and can be checked by the software polling mechanism.
  - Single interrupt signal output
- Data buffering
  - FIFO module based on dual port memory.
- ECC unit
  - The ECC module is based on the BCH algorithms.
  - The syndrome calculation, error detection and error correction phases in the BCH-based modules are pipelined.
  - The ECC unit supports the following options: 256, 512 bytes and 1024 bytes data blocks. The data block size depends on the selected ECC option.

- Advanced features
  - The page cache read/write sequences are supported.
  - The multiple planes read/write sequences are supported.
  - Command queuing mechanism.[Note]
- The advanced Bad Blocks Management system
  - Records tables are stored in the system memory.
  - Hardware implementation of the search algorithm.

NOTE

The queue is used when write access to registers is performed by the host. It is valid for all registers except:

- INT_STATUS
- STATUS
- FIFO_INIT
- ECC_STAT
- FIFO_DATA

Write to registers listed above will be performed immediately. Write to any other register will be performed through the command queue.

58.3 Register Descriptions

The table below shows the NFMA register configuration.

Register Access Size

All NFMAn registers are accessible in 32-bit units.

<NFMAn_base>

The base addresses <NFMAn_base> of the NFMAn are defined in the first section of this chapter under the key word “Register base addresses”.

CAUTION

The registers can be written only when the bit CTRL_STAT in STATUS register is ‘0’.

Table 58.8 Register Configuration (1/2)

Register Name	Symbol	Address
Controller commands register	COMMAND	<NFMAn_base> + 00_H
Main configurations register	CONTROL	<NFMAn_base> + 04_H
Controller status register	STATUS	<NFMAn_base> + 08_H
Mask register for the READ STATUS commands	STATUS_MASK	<NFMAn_base> + $0C_H$
Interrupts mask register	INT_MASK	<NFMAn_base> + 10_H
Interrupts status register	INT_STATUS	<NFMAn_base> + 14_H
ECC module control register	ECC_CTRL	<NFMAn_base> + 18_H
ECC offset in the spare area	ECC_OFFSET	<NFMAn_base> + $1C_H$
ECC module status register	ECC_STAT	<NFMAn_base> + 20_H
Column address register 0	ADDR0_COL	<NFMAn_base> + 24_H
Row address register 0	ADDR0_ROW	<NFMAn_base> + 28_H
Column address register 1	ADDR1_COL	<NFMAn_base> + $2C_H$
Row address register 1	ADDR1_ROW	<NFMAn_base> + 30_H
FIFO module interface	FIFO_DATA	<NFMAn_base> + 38_H
Data register	DATA_REG	<NFMAn_base> + $3C_H$
Data register size select	DATA_REG_SIZE	<NFMAn_base> + 40_H
Records table pointer register	DEV0_PTR	<NFMAn_base> + 44_H
Least significant part of the DMA base addressn	DMA_ADDR_L	<NFMAn_base> + 64_H
DMA module counter initial value	DMA_CNT	<NFMAn_base> + $6C_H$
DMA module control register	DMA_CTRL	<NFMAn_base> + 70_H
BBM module control register	BBM_CTRL	<NFMAn_base> + 74_H
Page size value	DATA_SIZE	<NFMAn_base> + 84_H
Timing configurations register 0	TIMINGS_ASYN	<NFMAn_base> + 88_H
Command sequence timing configuration	TIME_SEQ_0	<NFMAn_base> + 90_H
Command sequence timing configuration	TIME_SEQ_1	<NFMAn_base> + 94_H
Generic command sequence timing configuration	TIME_GEN_SEQ_0	<NFMAn_base> + 98_H
Generic command sequence timing configuration	TIME_GEN_SEQ_1	<NFMAn_base> + $9C_H$

Table 58.8 Register Configuration (2/2)

Register Name	Symbol	Address
Generic command sequence timing configuration	TIME_GEN_SEQ_2	<NFMAn_base> + $A0_H$
Control register for the FIFO module	FIFO_INIT	<NFMAn_base> + $B0_H$
FIFO module status	FIFO_STATE	<NFMAn_base> + $B4_H$
GENERIC_SEQ register	GEN_SEQ_CTRL	<NFMAn_base> + $B8_H$
Records table size register	DEV0_SIZE	<NFMAn_base> + $C0_H$
DMA trigger level value	DMA_TRIG_TLVL	<NFMAn_base> + 114_H
CMD ID initial value	CMD_MARK	<NFMAn_base> + 124_H
LUN per device status register	LUN STATUS 0	<NFMAn_base> + 128_H
Generic command sequence timing configuration	TIME_GEN_SEQ_3	<NFMAn_base> + 134_H
ECC errors counter register	ECC_CNT	<NFMAn_base> + $14C_H$

58.3.1 Controller commands register (COMMAND)

The write of the command sequence code to the COMMAND register triggers the programmed command sequence execution as soon as it is possible. If the execution cannot be started immediately, the transfer to this register is prolonged by the series of the WAIT responses. Each command sequence can trigger the interrupt when it is completed.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CMD_2[7:0]								CMD_1[7:0]/CMD_3[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CMD_0[7:0]								DATA_SEL	INPUT_SEL	CMD_SEQ[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.9 COMMAND Register Contents

Bit Position	Bit Name	Function
31 to 24	CMD_2[7:0]	Code of the third command in a sequence.
23 to 16	CMD_1[7:0]/ CMD_3[7:0]*1	Code of the second command in a sequence.
15 to 8	CMD_0[7:0]	Code of the first command in a sequence.
7	DATA_SEL	Data / FIFO selection flag: 0 – the FIFO module selected 1 – the DATA register selected
6	INPUT_SEL	Input module selection flag: 0 – select the registers Slave I/F Unit (SIU) module as input 1 – select the DMA module as input
5 to 0	CMD_SEQ[5:0]	Command code.

Note 1. Depending on the selected command sequence, this field will store CMD1 or CMD3 code. Generic Sequence is only one case where both commands are used in a single sequence.

58.3.2 Main configurations register (CONTROL)

The CONTROL register stores the configuration parameters that are common to all controller modules.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	AUTO_READ_STAT_EN	—	SMALL_BLOCK_EN	—	—	—	ADDR1_AUTO_INCR	ADDR0_AUTO_INCR
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	BBM_EN	IO_WIDTH	—	—	—	—	BLOCK_SIZE [1:0]		ECC_EN	INT_EN	—	ECC_BLOCK_SIZE [1:0]		READ_STATUS_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 58.10 CONTROL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	—	Reserved
23	AUTO_READ_STAT_EN	Auto Read Status mode enable. If active, the controller reads the status after the PROGRAM PAGE and BLOCK ERASE commands. It can trigger interrupt. The ERROR_MASK field in the STATUS_MASK register must be configured when this feature is enabled. It enables operation status checking after PROGRAM_PAGE/ERASE commands. It allows to detect that those operations failed. 0: Auto Read Status mode disabled 1: Auto Read Status mode enabled
22	—	Reserved
21	SMALL_BLOCK_EN	Enable small block mode. In this mode, the controller sends only a single byte as the column address instead of two bytes, as done in the big block NAND Flash devices. 0: big block mode enabled 1: small block mode enabled **CAUTION** Only devices that allow to disable CE signal when device is in the busy state are supported.
20 to 18	—	Reserved
17	ADDR1_AUTO_INCR	Address auto increment for row address register 1 (ADDR1_ROW). 0: auto increment disabled 1: auto increment enabled When this bit is set, sending any command sequence using address register 1 causes the increment of address register 1.
16	ADDR0_AUTO_INCR	Address auto increment for row address register 0 (ADDR0_ROW). 0: auto increment disabled 1: auto increment enabled When this bit is set, sending any command sequence using address register 0 causes the increment of address register 0.
15 to 14	—	Reserved
13	BBM_EN	Bad Block Management enable flag. For more details, see Section 58.4.4, Remapping Mechanism.
12	IO_WIDTH	NAND Flash I/O width. 0: 8 bits 1: Reserved
11 to 8	—	Reserved

Table 58.10 CONTROL Register Contents (2/2)

Bit Position	Bit Name	Function
7, 6	BLOCK_ SIZE[1:0]	The Block Size. 00: 32 pages per block 01: 64 pages per block 10: 128 pages per block 11: 256 pages per block
5	ECC_EN	Hardware ECC support enable. 0: ECC disabled 1: ECC enabled Hardware ECC can be used only when m * (ECC_BLOCK_SIZE[1:0]) ≤ DATA_SIZE[14:0] ≤ m * (ECC_BLOCK_SIZE[1:0] + 32), where m is 1,2,3,...
4	INT_EN	Global Interrupt enable. 0: Interrupts disabled 1: Interrupts enabled For more details, see Section 58.4.5, Interrupts Mechanism.
3	—	Reserved
2, 1	ECC_BLOCK_ SIZE[1:0]	The ECC Block Size (depends on core configuration): 00: 256 bytes 01: 512 bytes 10: 1024 bytes 11: not available The ECC block size can be changed only when all memory devices are ready.
0	READ_STATUS _EN	Automatically READ STATUS / check RnB lines. The STATUS_MASK field in the STATUS_MASK register must be configured when this feature is enabled. It selects how the controller detects device ready/busy state: 0: The controller checks RnB lines 1: The controller sends READ STATUS commands **CAUTION** Automatically sent READ STATUS command is only available in devices compatible with ONFI 1.0.

58.3.3 GENERIC_SEQ register (GEN_SEQ_CTRL)

The GENERIC_SEQ register is used to parameterize the generic command sequences.For more details, see Section 58.4.2, Generic Command Sequence. For examples, see **Table 58.47**.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CMD3[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	IMD_SEQ	DELAY_EN[1:0]		DATA_EN	ROW_A1[1:0]		ROW_A0[1:0]		COL_A1[1:0]		COL_A0[1:0]		CMD3_EN	CMD2_EN	CMD1_EN	CMD0_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.11 GEN_SEQ_CTRL Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	—	Reserved
23 to 16	CMD3[7:0]	Command 3 code value.
15	IMD_SEQ	Enable immediate command execution. This bit allows the command sequence to be executed without checking the selected target state. 0: feature disabled 1: feature enabled
14, 13	DELAY_EN[1:0]	Enable the busy 0 or 1 phase. This bit allows enabling or disabling the presence of the “busy” phase in the universal command sequence. 00: disable both delays 01: enable delay 0 10: enable delay 1 11: disable both delays
12	DATA_EN	Enable data part sequence. This bit allows enabling or disabling the data phase of the universal command sequence. 0: disable data phase 1: enable data phase
11, 10	ROW_A1[1:0]	Row Address Cycles. Number of the row address bytes sent to NAND Flash device. 00: 0 address cycles 01: 1 address cycles 10: 2 address cycles 11: 3 address cycles
9, 8	ROW_A0[1:0]	Row Address Cycles. Number of the row address bytes sent to NAND Flash device. 00: 0 address cycles 01: 1 address cycles 10: 2 address cycles 11: 3 address cycles
7, 6	COL_A1[1:0]	Column Address Cycles. Number of the column address bytes sent to NAND Flash device. 00: 0 address cycles 01: 1 address cycles 10: 2 address cycles 11: not available

Table 58.11 GEN_SEQ_CTRL Register Contents (2/2)

Bit Position	Bit Name	Function
5, 4	COL_A0[1:0]	Column Address Cycles. Number of the column address bytes sent to NAND Flash device. 00: 0 address cycles 01: 1 address cycles 10: 2 address cycles 11: not available
3	CMD3_EN	Enable Command 3 phase. This bit allows enabling or disabling the presence of the “command 3” phase in the universal command sequence. 1: enable 0: disable
2	CMD2_EN	Enable Command 2 phase. This bit allows enabling or disabling the presence of the “command 2” phase in the universal command sequence. 1: enable 0: disable
1	CMD1_EN	Enable Command 1 phase. This bit allows enabling or disabling the presence of the “command 1” phase in the universal command sequence. 1: enable 0: disable
0	CMD0_EN	Enable Command 0 phase. This bit allows enabling or disabling the presence of the “command 0” phase in the universal command sequence. 1: enable 0: disable

58.3.4 Controller status register (STATUS)

The STATUS register stores the NAND Flash controller and connected device status flags. These flags can be used to obtain the current controller internal state.

The CTRL_STAT flag is set after the controller starts to execute the requested command for the selected NAND Flash device, and it is active while the command execution is not completed. Command execution can be divided into two phases. In the first phase, the command sequence is executed at the moment when the NAND Flash device goes into the busy state. After that, the controller stores information about the pending operation on the selected device. In the second phase, the controller automatically finishes the pending command execution based on the previously stored data. As long as this flag is set, the controller does not accept any new commands.

The MEMx_ST flags correspond to the NAND Flash device with the same index value. The flags give information about the NAND Flash devices' state. The flag has the same function as the NAND Flash device RnB line.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	CMD_ID[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	DATA_REG_ST	DATASIZE_ERROR_ST	CTRL_STAT	—	—	—	—	—	—	—	MEM0_ST
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 58.12 STATUS Register Contents (1/2)

Bit Position	Bit Name	Function
31 to 24	—	Reserved
23 to 16	CMD_ID[7:0]	Command ID The current command under execution identification marker added before command was put in to the command FIFO.
15 to 11	—	Reserved
10	DATA_REG_ST	The DATA_REG: Resetting of this flag is possible only by reading the data from the DATA_REG register. 1: data in DATA_REG is available 0: data in DATA_REG is not available
9	DATASIZE_ERROR_ST	The Data Size value error: When the ECC is enabled, this bit signal incorrect value in the DATA_SIZE register. The algorithm for correct data size value can be found in the DATA_SIZE register description. 0: correct value 1: incorrect value **Note:** When using "READ PARAMETER PAGE" command, ECC Block Size should same with DATA_SIZE. When using "SET FEATURES" or "GET FEATURES" or "READ ID" commands, ECC function should be disabled.
8	CTRL_STAT	The main controller status bit: 0: controller ready 1: controller busy
7 to 1	—	Reserved

Table 58.12 STATUS Register Contents (2/2)

Bit Position	Bit Name	Function
0	MEM0_ST	Device 0 status flag: 1: device ready 0: device busy

58.3.5 LUN per device status register (LUN_STATUS0)

The LUN_STATUS0 register allows to access the LUN status information for devices 0-3. Each bit of the LUN status field contain the status of single LUN of device. The busy status is marked as logical ‘0’, the ready state is marked as logical ‘1’.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	MEM0_LUN[7:0]							
Initial value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 58.13 LUN_STATUS0 Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved
7 to 0	MEM0_LUN[7:0]	Memory 0 LUN-s status field.

58.3.6 Interrupts mask register (INT_MASK)

The INT_MASK register allows masking of the selected interrupts source in the NAND Flash controller. The masked interrupts still set the appropriate bits in the status register, but do not assert the interrupt signal.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	ECC_INT0_EN	—	—	—	—	—	—	—	STAT_ERR_INT0_EN
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	MEM0_RDY_INT_EN	—	PG_SZ_ERR_INT_EN	—	TRANS_ERR_EN	DMA_INT_EN	DATA_REG_INT_EN	CMD_END_INT_EN	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R

Table 58.14 INT_MASK Register Contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved
24	ECC_INT0_EN	Enables the interrupt from the ECC module status for Memory device 0. 0: interrupt disabled 1: interrupt enabled
23 to 17	—	Reserved
16	STAT_ERR_INT0_EN	Enables the interrupt when the most recently finished operation on the Memory device 0 failed. This applies to PROGRAM PAGE and BLOCK ERASE operations. It is not valid following a READ-series operation. 0: interrupt disabled 1: interrupt enabled
15 to 9	—	Reserved
8	MEM0_RDY_INT_EN	The memory device 0 is ready for the new command: 0: interrupt disabled 1: interrupt enabled For more details see **Figure 58.30**.
7	—	Reserved
6	PG_SZ_ERR_INT_EN	Data Size error occur. 0: interrupt disabled 1: interrupt enabled
5	—	Reserved
4	TRANS_ERR_EN	The transfer on the slave interface error: 0: interrupt disabled 1: interrupt enabled
3	DMA_INT_EN	DMA transfer ended. 0: interrupt disabled 1: interrupt enabled
2	DATA_REG_INT_EN	Data in DATA_REG is available. 0: interrupt disabled 1: interrupt enabled
1	CMD_END_INT_EN	Command sequence ended. 0 – interrupt disabled 1 – interrupt enabled For more details see **Figure 58.30**.
0	—	Reserved

58.3.7 Interrupts status register (INT_STATUS)

The INT_STATUS register stores the NAND Flash controller interrupt flags. If the given bit is 0, the corresponding interrupt condition is not met. If the given bit is 1, the corresponding interrupt condition is met.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	ECC_INT0_FL	—	—	—	—	—	—	—	STAT_ERR_INT0_FL
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	MEM0_RDY_INT_FL	—	PG_SZ_ERR_INT_FL	—	TRANS_ERR_FL	DMA_INT_FL	DATA_REG_INT_FL	CMD_END_INT_FL	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W	R

Table 58.15 INT_STATUS Register Contents

Bit Position	Bit Name	Function
31 to 25	—	Reserved
24	ECC_INT0_FL	Selected flag (source is ECC_UNC_0, ECC_ERROR_0 or ECC_OVER_0) in the ECC module is set.
23 to 17	—	Reserved
16	STAT_ERR_INT0_FL	Most recently finished operation on the Memory device 0 failed. This applies to PROGRAM PAGE and BLOCK ERASE operations. It is not valid following a READ-series operation.
15 to 9	—	Reserved
8	MEM0_RDY_INT_FL	The memory device 0 is ready for the new command.
7	—	Reserved
6	PG_SZ_ERR_INT_FL	Data Size error flag. When the ECC is enabled, the value written into the DATA_SIZE register has some restrictions. Interrupt condition is met when the value written to the DATA_SIZE register is not correct.
5	—	Reserved
4	TRANS_ERR_FL	The transfer on the slave interface error. The flag is set when the access to the FIFO memory has the opposite direction to the current FIFO configuration.
3	DMA_INT_FL	DMA transfer ended flag.
2	DATA_REG_INT_FL	Data in DATA_REG is available.
1	CMD_END_INT_FL	Transfer sequence ended.
0	—	Reserved

58.3.8 ECC module control register (ECC_CTRL)

The ECC_CTRL register stores all configuration parameters required by the ECC.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC_SEL[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	ERR_THRESHOLD[5:0]						—	—	—	—	—	ECC_CAP[2:0]		
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W

Table 58.16 ECC_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 18	—	Reserved
17, 16	ECC_SEL[1:0]	The ECC interrupt source select. These bits selects the ECC module flag that will be used as a source for the interrupt signal: 00: select ECC_ERROR (correctable error) flag as interrupt source. 01: select ECC_UNC (uncorrectable error) flag as interrupt source. 1x: select ECC_OVER (acceptable errors level overflow) flag as interrupt source. (ECC_OVER flag is not set by uncorrectable errors (ECC_UNC_0 = 1))
15, 14	—	Reserved
13 to 8	ERR_THRESHOLD[5:0]	The acceptable errors level. The value of this field contains the number of errors that is acceptable. This field must be initialized by the application program.
7 to 3	—	Reserved
2 to 0	ECC_CAP[2:0]	The ECC module correction ability. The correction ability can be changed only when all memory devices are ready. 000: 2 001: 4 010: 8 011: 16 100: 24 All others: 32

58.3.9 ECC module status register (ECC_STAT)

The ECC_STAT register stores all ECC module status information.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECC_OVER_0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	ECC_UNC_0	—	—	—	—	—	—	—	ECC_ERROR_0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R	R/W

Table 58.17 ECC_STAT Register Contents

Bit Position	Bit Name	Function
31 to 17	—	Reserved
16	ECC_OVER_0	The Memory device 0 acceptable errors level overflow. The bit is set when the number of errors is bigger than the value of declared ERR_THRESHOLD[5:0] bits. (Uncorrectable errors are not counted)
15 to 9	—	Reserved
8	ECC_UNC_0	The Memory device 0 uncorrectable error flag. The bit is set when the uncorrectable errors occur during the read operation.
7 to 1	—	Reserved
0	ECC_ERROR_0	The Memory device 0 correctable error flag. The bit is set when the correctable errors or the uncorrectable errors (ECC_UNC_0 = 1) occur during the read operation.

58.3.10 ECC offset in the spare area register (ECC_OFFSET)

The ECC_OFFSET register stores the offset value from the beginning of the page to the place where correction words will be stored.

The value of the ECC_OFFSET register must be bigger than the value in the DATA_SIZE register.

In small block mode, the value in the ECC_OFFSET is ignored and the correction words are located in the NAND Flash memory device just behind the data.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ECC_OFFSET[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.18 ECC_OFFSET Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved
15 to 0	ECC_ OFFSET[15:0]	Correction words block offset.

58.3.11 ECC error level counter register (ECC_CNT)

The ECC_CNT register stores number of value detected during last page read operation. This register content is not automatically cleared, it must be done by software. The new page read operation does not overwrite previous register value. Register contain value of the largest error level detected in processed ECC blocks.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	ERR_LVL[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.19 ECC_CNT Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved
5 to 0	ERR_LVL[5:0]	Detected error level. (Uncorrectable errors are not counted)

58.3.12 Column/row address registers (ADDR[1:0]_COL, ADDR[1:0]_ROW)

The address registers store the packaged version of the address that will be used by the next command sequence during access to the NAND Flash device.

ADDR[1:0]_COL:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[1:0]_COL[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ADDR[1:0]_ROW:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	ADDR[1:0]_ROW[23:16]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ADDR[1:0]_ROW[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.20 ADDR[1:0]_COL Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved
15 to 0	ADDR[1:0]_ COL[15:0]	Column address. A15-A0 address bits.

Table 58.21 ADDR[1:0]_ROW Register Contents

Bit Position	Bit Name	Function
31 to 24	—	Reserved
23 to 0	ADDR[1:0]_ ROW[23:0]	Row address. A39-A16 address bits (Page address, Block address and LUN address in the ONFI case).

CAUTIONS

1. **There is no register that defines the total memory size of the NAND Flash memory chip, thus the controller is not able to determine which address bits in ADDR[1:0]_COL and ADDR[1:0]_ROW are important and which must be zero. For this reason, the software must take special care with the values written to these registers. Incorrect values of unused address bits (none '0' values) can cause errors in memory access.**

A relation between address registers and memory device address width is configured by the command sequence field of the COMMAND register. This field determines which command sequence has to be used and how many address bytes are used when addressing a NAND Flash memory device. (Example: In order to erase blocks, the three address cycles containing the row address are written into the NAND Flash memory device. The NAND Flash Controller automatically writes bits A39-A16 to the NAND Flash device). Refer to Section 58.4.1.2, Command Sequence Encoding and Table 58.47 in order to see how many address cycles are written into the NAND Flash memory device by each Command Sequence.
The address written to the address register must be aligned according to the NAND Flash device. Unused bits must be padded with zeros.

2. **When the auto increment for the row address register is enabled, the proper value of this register can be read only when the bit CTRL_STAT in STATUS register is clear.**

Table 58.22 Address Registers and Address Bytes Relationship

Address Cycle	I/O 0	I/O 1	I/O 2	I/O 3	I/O 4	I/O 5	I/O 6	I/O 7
1st cycle	A0	A1	A2	A3	A4	A5	A6	A7
2nd cycle	A8	A9	A10	A11	A12	A13	A14	A15
3rd cycle	A16	A17	A18	A19	A20	A21	A22	A23
4th cycle	A24	A25	A26	A27	A28	A29	A30	A31
5th cycle	A32	A33	A34	A35	A36	A37	A38	A39

58.3.13 Page size value register (DATA_SIZE)

The DATA_SIZE register stores the value of the data block size. The data size value is remembered as the number of bytes per transferred block, but its size must be declared as the multiple of the chosen NAND Flash word size. The unused bits for the given word size configuration are ignored and replaced with zeros.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	DATA_SIZE[14:0]														
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.23 DATA_SIZE Register Contents

Bit Position	Bit Name	Function
31 to 15	—	Reserved
14 to 0	DATA_SIZE[14:0]	Data size. The value of this field defines data size.

CAUTION

Write the proper value to the DATA_SIZE register when ECC is enabled:

(ECC_BLOCK_SIZE[1:0]) × m ≤ DATA_SIZE[14:0] ≤ (ECC_BLOCK_SIZE[1:0] + 32) × m, where m is 1, 2, 3 …

Here is an example of which values are correct when ECC_BLOCK_SIZE[1:0] equals 512 bytes:

Table 58.24 Example, in which DATA_SIZE Values are Correct (1/2)

DATA_SIZE[14:0]	Value Correctness
0x0000	Not available
….	….
0x001F	Not available
0x0020	Not available
0x0021	Not available
….	….
0x01FF	Not available
0x0200	Available
0x0201	Available
….	….
0x0220	Available
0x0221	Not available

Table 58.24 Example, in which DATA_SIZE Values are Correct (2/2)

DATA_SIZE[14:0]	Value Correctness
….	….
0x03FF	Not available
0x0400	Available
0x0401	Available
….	….
0x0440	Available
0x0441	Not available
….	….

Note 1. ECC_BLOCK_SIZE[1:0] equals 512 bytes.

58.3.14 FIFO module interface register (FIFO_DATA)

The FIFO_DATA register is used as an entry point to the FIFO module. The CPU can access the FIFO module by reading from or writing to the FIFO_DATA register in the same way as it accesses any other register.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	FIFO_DATA[31:16]															
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FIFO_DATA[15:0]															
Initial value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.25 FIFO_DATA Register Contents

Bit Position	Bit Name	Function
31 to 0	FIFO_ DATA[31:0]	FIFO Data.

NOTE

FIFO_DATA register is able to access after issuing read or write command.

58.3.15 Bad block management (BBM) module control register (BBM_CTRL)

The BBM control register stores the BBM specific control parameters.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RMP_INIT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W

Table 58.26 BBM_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved
0	RMP_INIT	Remap initial flag. If set, this flag forces the BBM module to reread the remapping table after it was updated by software. This flag is set by software and cleared by hardware after rereading the remapping table.

58.3.16 Records table pointer register (DEV0_PTR)

The bad block management mechanism uses the tables in the system memory to store the remapping records. The DEV0_PTR register stores the table of the remapping record address.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	PTR_ADDR[11:2]										—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 58.27 DEV0_PTR Register Contents

Bit Position	Bit Name	Function
31 to 12	–	Reserved
11 to 2	PTR_ ADDR[11:2]	Remap table pointer. The field contains an address of the remap table in the internal memory.
1, 0	–	Reserved

58.3.17 Records table size register (DEV0_SIZE)

The bad block management mechanism implemented in the controller uses the tables in the system memory to store the remapping records. Each table can store a variable number of records depending on the number of bad blocks in the NAND Flash device.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	DEV_SIZE[11:0]											
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.28 DEV0_SIZE Register Contents

Bit Position	Bit Name	Function
31 to 12	—	Reserved
11 to 0	DEV_SIZE[11:0]	Number of record.

58.3.18 DMA base address - least significant part register (DMA_ADDR_L)

The DMA_ADDR_L register is the least significant part of the 64-bit DMA base address. The register contains an address for the first data in the data block in the system memory, or the address of the first descriptor. The DMA module can read data from the memory location set by DMA_ADDR and write it to the FIFO module, or read data from the FIFO module and write it to the memory, starting from the location indicated by the DMA_ADDR.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DMA_ADDR_L[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMA_ADDR_L[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.29 DMA_ADDR_L Register Contents

Bit Position	Bit Name	Function
31 to 0	DMA_ADDR_L[31:0]	The least significant part of the DMA base address. The two least significant bits are ignored, thus the address must be aligned to 32-bit words.

58.3.19 DMA counter initial value register (DMA_CNT)

The DMA_CNT defines the number of bytes that will be transferred by the DMA module. The register remains unchanged during the transfer process.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	CNT_INIT[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CNT_INIT[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.30 DMA_CNT Register Contents

Bit Position	Bit Name	Function
31 to 0	CNT_INIT[31:0]	Bytes counter initial value. The field contains data page length in bytes ($0000\ 0004_H$ – $FFFF\ FFFC_H$). The number of the bytes has to be divided by 4.

58.3.20 DMA control register (DMA_CTRL)

DMA_CTRL is a control register for the DMA channel. This register defines the parameters of the DMA transfer.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DMA_ START	—	DMA_ MODE	DMA_ BURST[2:0]			—	DMA_ READY
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R/W	R	R	R	R	R	R	R	R	R/W	R	R/W	R/W	R/W	R/W	R	R

Table 58.31 DMA_CTRL Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved
7	DMA_START	DMA start. Set bit DMA_START to start DMA when the command sequence is sent to the NAND Flash memory. For more details see Section 58.5.2.2, DMA Description.
6	—	Reserved
5	DMA_MODE	DMA work mode: 0: the registers managed mode 1: the Scatter-Gather mode
4 to 2	DMA_ BURST[2:0]	Burst type. These bits define the main transfer type used by the DMA to precede the requested transfer. 000: incrementing precise burst of precisely four transfers (address increment) 010: single transfer (address increment) 011: burst of unspecified length (address increment) 100: incrementing precise burst of precisely eight transfers (address increment) 101: incrementing precise burst of precisely sixteen transfers (address increment) All others: Setting prohibited
1	—	Reserved
0	DMA_READY	DMA ready flag. The flag is set when transfer is completed.

58.3.21 DMA trigger level value register (DMA_TRIG_TLVL)

This register allows the setting of the data FIFO occupancy level that will trigger the DMA module. For more details see Section 58.5.2.2, DMA Description.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DMA_TRIG_LVL[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.32 DMA_TRIG_TLVL Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved
7 to 0	DMA_TRIG_ LVL[7:0]	DMA trigger level The trigger level is counted using the 32-bit words as entity.

58.3.22 Mask register for the READ STATUS commands (STATUS_MASK)

The STATE_MASK field is used to mark the ready/busy bits in the NAND Flash device status byte. This field is used during the internal read status operation. In the case of ONFI, the user must mask all fields except RDY or ARDY (depending on the application).

The ERROR_MASK field is used to mask unused fields when the controller automatically reads the status of the NAND Flash memory device. In the case of ONFI, the user must mask all fields except FAIL or FAILC (depending on the application).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	ERROR_MASK[7:0]								STATE_MASK[7:0]							
Initial value	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.33 STATUS_MASK Register Contents

Bit Position	Bit Name	Function
31 to 16	—	Reserved
15 to 8	ERROR_ MASK[7:0]	Error State Mask – used to mask the error bits if automatic read status feature is enabled.
7 to 0	STATE_ MASK[7:0]	State Mask – used to mask status bits when the read status command is used to obtain the NAND flash status.

58.3.23 Command sequence timing configuration register 0 (TIME_SEQ_0)

Some waveform configuration parameters are defined in the TIME_SEQ_0 register. The time delay generated by the controller equals the minimum value written into the register, increased by 1. All the timings are generated using the CLKB clock signal.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	TWHR[5:0]						—	—	TRHW[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TADL[5:0]						—	—	TCCS[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.34 TIME_SEQ_0 Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved
29 to 24	TWHR[5:0]	WE# high to RE# low time
23, 22	—	Reserved
21 to 16	TRHW[5:0]	RE# high to WE# low time
15, 14	—	Reserved
13 to 8	TADL[5:0]	ALE to data start time
7, 6	—	Reserved
5 to 0	TCCS[5:0]	Change column setup.

58.3.24 Command sequence timing configuration register 0 (TIME_SEQ_1)

Some waveform configuration parameters are defined in the TIME_SEQ_1 register. The time delay generated by the controller equals the minimum value written into the register, increased by 1. All the timings are generated using the CLKB clock signal.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	TRR[5:0]						—	—	TWB[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.35 TIME_SEQ_1 Register Contents

Bit Position	Bit Name	Function
31 to 14	—	Reserved
13 to 8	TRR[5:0]	Read high to Read low. TRR time period from rising edge on read/busy input line to the moment when the read enable signal can be asserted.
7, 6	—	Reserved
5 to 0	TWB[5:0]	tWB delay. Time period measured from rising edge of the WE# or CLK signal to a falling edge on the RnB line or the NAND flash device SR[6] low.

58.3.25 Generic command sequence timing configuration register 0 (TIME_GEN_SEQ_0)

Some waveform configuration parameters for the Generic Sequence are defined in the TIME_GEN_SEQ_0 register. The time delay generated by the controller equals the minimum value written into the register, increased by 1. For more detail see Section 58.4.2, Generic Command Sequence. All the timings are generated using the CLKB clock signal.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T0_D3[5:0]						—	—	T0_D2[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	T0_D1[5:0]						—	—	T0_D0[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.36 TIME_GEN_SEQ_0 Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved
29 to 24	T0_D3[5:0]	Command to Data time. The time between sending a command and data transferring.
23, 22	—	Reserved
21 to 16	T0_D2[5:0]	Command to Delay time. The time between sending a command to the NAND Flash memory device and waiting until the memory is ready.
15, 14	—	Reserved
13 to 8	T0_D1[5:0]	Command to Command time. The time between two subsequent commands sent to the NAND Flash memory device.
7, 6	—	Reserved
5 to 0	T0_D0[5:0]	Command to Address time. The time between sending a command and sending the address to the NAND Flash memory device.

58.3.26 Generic command sequence timing configuration register 1 (TIME_GEN_SEQ_1)

Some waveform configuration parameters for the Generic Sequence are defined in the TIME_GEN_SEQ_1 register. The time delay generated by the controller equals the minimum value written into the register, increased by 1. For more detail see Section 58.4.2, Generic Command Sequence. All the timings are generated using the CLKB clock signal.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T0_D7[5:0]						—	—	T0_D6[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	T0_D5[5:0]						—	—	T0_D4[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.37 TIME_GEN_SEQ_1 Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved
29 to 24	T0_D7[5:0]	Address to Data time. The time between sending the address and data transferring.
23, 22	—	Reserved
21 to 16	T0_D6[5:0]	Address to Delay time. The time between sending the address to the NAND Flash memory device and waiting until the memory is ready.
15, 14	—	Reserved
13 to 8	T0_D5[5:0]	Address to Address time. The time between two subsequent addresses sent to the NAND Flash memory device.
7, 6	—	Reserved
5 to 0	T0_D4[5:0]	Address to Command time. The time between sending the address and sending a command to the NAND Flash memory device.

58.3.27 Generic command sequence timing configuration register 2 (TIME_GEN_SEQ_2)

Some waveform configuration parameters for the Generic Sequence are defined in the TIME_GEN_SEQ_2 register. The time delay generated by the controller equals the minimum value written into the register, increased by 1. For more detail see Section 58.4.2, Generic Command Sequence. All the timings are generated using the CLKB clock signal.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	T0_D11[5:0]						—	—	T0_D10[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	T0_D9[5:0]						—	—	T0_D8[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.38 TIME_GEN_SEQ_2 Register Contents

Bit Position	Bit Name	Function
31, 30	—	Reserved
29 to 24	T0_D11[5:0]	Data to Delay time. The time between data transferring and waiting until the memory is ready.
23, 22	—	Reserved
21 to 16	T0_D10[5:0]	Data to Command time. The time between data transferring and sending a command to the NAND Flash memory device.
15, 14	—	Reserved
13 to 8	T0_D9[5:0]	Delay to Command time. The time between waiting until the memory is ready and sending a command to the NAND Flash memory device.
7, 6	—	Reserved
5 to 0	T0_D8[5:0]	Delay to Data time. The time between waiting until the memory is ready and data transferring.

58.3.28 Generic command sequence timing configuration register 3 (TIME_GEN_SEQ_3)

Some waveform configuration parameters for the Generic Sequence are defined in the TIME_GEN_SEQ_3 register. For more details see Section 58.4.2, Generic Command Sequence. All the timings are generated using the CLKB clock signal.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	T0_D12[5:0]					
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.39 TIME_GEN_SEQ_3 Register Contents

Bit Position	Bit Name	Function
31 to 6	—	Reserved
5 to 0	T0_D12[5:0]	Data to sequence end time. The time between the data transferring phase and sequence end. The sequence is ended when any other sequence phases after the data transfer phase are not enabled.

58.3.29 Timing configuration register (TIMINGS_ASYN)

Two waveform configuration parameters are defined in the TIMINGS_ASYN register. The time delay generated by the controller equals the minimum value written into the register, increased by 1. All the timings are generated using the CLKB clock signal.

Figure 58.1 shows how timings parameters are mapped to the NAND flash interface. The upper part of figure shows read transfer, lower part shows write transfer.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	TRWH[3:0]				TRWP[3:0]			
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 58.40 TIMINGS_ASYN Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved
7 to 4	TRWH[3:0]	RE# or WE# high hold time.
3 to 0	TRWP[3:0]	RE# or WE# pulse width.

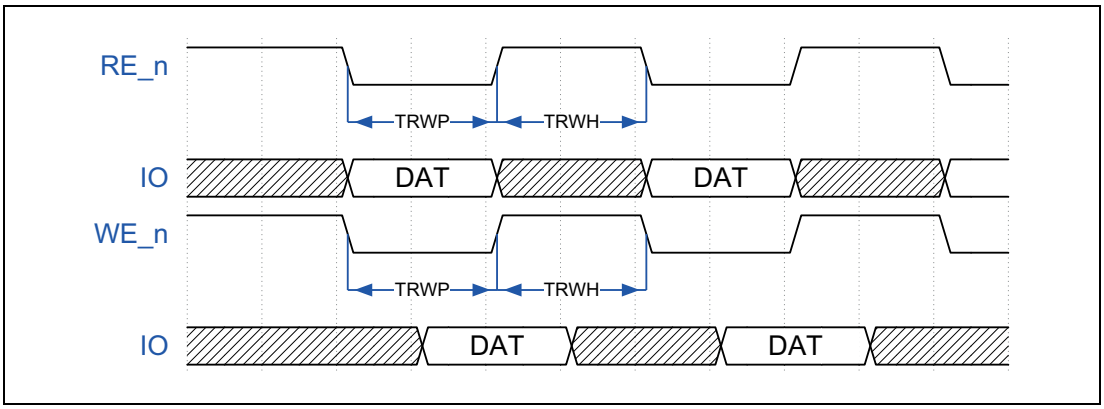


Figure 58.1 Asynchronous timings

58.3.30 Data register (DATA_REG)

The DATA register is used for storage of the data that is read in the registered mode. The registered mode is allowed in the read direction only.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	DATA_REG[31:16]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DATA_REG[15:0]															
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 58.41 DATA_REG Register Contents

Bit Position	Bit Name	Function
31 to 0	DATA_ REG[31:0]	DATA_REG register.

58.3.31 Data register size selection register (DATA_REG_SIZE)

The DATA_REG_SIZE register allows the selection of data size in the registered work mode. The data size in the registered mode is limited to four bytes.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATA_REG_ SIZE[1:0]	
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

Table 58.42 DATA_REG_SIZE Register Contents

Bit Position	Bit Name	Function
31 to 2	—	Reserved
1, 0	DATA_REG_ SIZE[1:0]	DATA_REG_SIZE register. Allows selection of the number of valid bytes in the DATA register: 00: single byte valid 01: two lower bytes valid 10: three lower bytes valid 11: all four bytes valid

58.3.32 FIFO control register (FIFO_INIT)

The FIFO_INIT register contains the FIFO_INIT bit.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FIFO_INIT
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	W

Table 58.43 FIFO_INIT Register Contents

Bit Position	Bit Name	Function
31 to 1	—	Reserved
0	FIFO_INIT	FIFO init bit. The setting of this bit causes the flushing of FIFO. It is not necessary to set this bit before sending each command to the NAND Flash memory device. This feature is reserved only for a situation where previous FIFO content must be purged before a new operation.

58.3.33 FIFO status control register (FIFO_STATE)

The FIFO_STATE register contains the data buffer status.

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	DF_W_EMPTY	DF_R_FULL	CF_ACCPT_W	CF_ACCPT_R	CF_FULL	CF_EMPTY	DF_W_FULL	DF_R_EMPTY
Initial value	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 58.44 FIFO_STATE Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved
7	DF_W_EMPTY	FIFO empty state bit. This bit indicates that there is no data in the FIFO available. This flag is valid for the write direction.
6	DF_R_FULL	FIFO full state bit. This bit indicates that there is no free space for the data in FIFO available. This flag is valid for the read direction.
5	CF_ACCPT_W	Command FIFO accept flag – write direction. If this flag is set then next write access will finish with any additional delay.
4	CF_ACCPT_R	Command FIFO accept flag – read direction. This is informational flag. If it is set then read transfer on the CMD FIFO internal interface will be accepted with any additional delay.
3	CF_FULL	Command FIFO full flag. This bit indicates actual command FIFO state. It can't be used to check if next transfer will be accepted.
2	CF_EMPTY	Command FIFO empty flag. This bit indicates actual command FIFO state. It can't be used to check if next transfer will be accepted.
1	DF_W_FULL	FIFO full state bit. This bit indicates that there is no free space for the data in FIFO available. This flag is valid for the write direction.
0	DF_R_EMPTY	FIFO empty state bit. This bit indicates that there is no data in the FIFO available. This flag is valid for the read direction.

58.3.34 CMD ID initial value register (CMD_MARK)

The CMD MARK register allows to write initial value to the command marking generator in the register Slave I/F Unit (SIU).

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—	CMD_ID[7:0]							
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	W

Table 58.45 CMD_MARK Register Contents

Bit Position	Bit Name	Function
31 to 8	—	Reserved
7 to 0	CMD_ID[7:0]	CMD ID initial value.

58.4 Operation

58.4.1 Command Generation

The NAND Flash devices are constantly extended and evaluated to the larger capacities and higher data throughput. Frequently during this process, new commands appear in the next generation of devices. To allow use of the NAND Flash controller with future generations of devices, the parameterizable command sequences have been added.

The parameterization allows for a defining set of parameters for each supported command sequence.

58.4.1.1 Instruction Encoding

The controller instruction field is constant and has 32 bits. The instruction field contains the command sequence code and optional parameters. Those parameters are:

- The command codes present in the instruction sequence.
- The flag used to select data destination; possible options are data register and FIFO module.
- The flag used to select data source/sink for the command sequence. The possible choices are the registers or the DMA unit.
- The command sequence code.

If the given command sequence does not use all parameter fields, unused fields are ignored. The instruction encoding scheme is presented in the table below:

Table 58.46 Instruction Encoding

Field Name	Bits	Description
CMD_2	[31:24]	Code of the third command in a sequence.
CMD_1/CMD3*1	[23:16]	Code of the second or fourth command in a sequence.
CMD_0	[15:8]	Code of the first command in a sequence.
DATA_SEL	[7]	Data register / FIFO select flag: 0 – the FIFO selected 1 – the DATA register selected
INPUT_SEL	[6]	Input module select flag: 0 – select the register as input 1 – select the DMA module as input
CMD_SEQ	[5:0]	Command code.

Note 1. Depending on the selected command sequence, this field will store the CMD1 or CMD3 code. Both commands are never used in a single sequence.

58.4.1.2 Command Sequence Encoding

The NAND Flash devices use the same set of signals independently of the memory capacity. This allows upgrading of obsolete devices with newer ones without PCB redesign. The NAND Flash devices use common IO bus to transfer commands, addresses and data. The predefined set of command sequences is used for the read and write operations on those devices. The set of supported command sequences is not constant for all NAND Flash device producers and evolves as the devices are more capable.

The NAND Flash controller must be able to support the new NAND Flash device features as they appear with a minimum effort from the designer side. This objective can be achieved in many cases because most of the new instructions use the previously defined sequence of commands and addresses, along with new command codes, data page size, data spare area size, etc..

The NAND Flash controller defines the set of commands, addresses and data sequences that allows implementation of all present and many future instructions. The following description of those sequences is adequate to define most of the future NAND Flash device instructions.

Table 58.47 contains the command sequence encoding details. Each sequence is encoded according to the fields defined in the GENERIC_SEQUENCE_CTRL register. For more details, **Section 58.3.3, GENERIC_SEQ register (GEN_SEQ_CTRL)**.

Table 58.47 Command Sequence Encoding

Sequence symbol	Sequence encoding	CMD0	CMD1	CMD2	CMD3	COL_A0*2	COL_A1	ROW_A0	ROW_A1	DATA_EN	DELAY_EN	IMD_SEQ*3
SEQ_0	000000	√	-	-	-	-	-	-	-	-	DELAY_1	-
SEQ_1	100001	√	-	-	-	1	-	-	-	√	-	-
SEQ_2	100010	√	-	-	-	1	-	-	-	√	DELAY_0	-
SEQ_3	000011	√	-	-	-	1	-	-	-	√	DELAY_1	-
SEQ_4	100100	√	-	-	-	-	-	-	-	√	-	√
SEQ_5	100101	√	-	-	-	-	-	3	-	√	-	√
SEQ_6	100110	√	-	√	-	2(1)	-	-	-	√	-	-
SEQ_7	100111	√	-	√	-	2(1)	-	3	-	√	DELAY_0	-
SEQ_8	001000	√	-	-	-	2(1)	-	-	-	√	-	-
SEQ_9	101001	√	√	-	-	2(1)	-	3	-	-	DELAY_1	-
SEQ_10	101010	√	-	√	-	2(1)	-	3	-	√	DELAY_0	-
SEQ_11	101011	√	-	-	-	-	-	-	-	√	DELAY_0	-
SEQ_12	001100	√	√	-	-	2(1)	-	3	-	√	DELAY_1	-
SEQ_13	001101	√	-	-	-	2(1)	-	3	-	√	DELAY_1	-
SEQ_14	001110	√	√	-	-	-	-	3	-	-	DELAY_1	-
SEQ_15	101111	√	-	√	√	2	2	3	3	√	DELAY_0	-
SEQ_17	110001	√	-	-	-	2(1)	-	3	-	√	DELAY_1	-
SEQ_18	110010	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
SEQ_19	010011	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1
SEQ_20	110100	√	-	-	-	-	-	3	-	-	DELAY_1	-
SEQ_21	010101	√	-	-	-	1	-	-	-	-	-	-
SEQ_22	110110	√	-	√	-	2(1)	2	-	3	√	DELAY_0	-
SEQ_23	010111	√	√	-	-	-	-	3	-	√	DELAY_1	-
SEQ_24	011000	√	-	√	√	-	-	3	3	-	DELAY_0	-
SEQ_25	111001	√	-	√	√	2(1)	2	3	-	√	-	-

Note 1. SEQ_18 and SEQ_19 are the parameterized Generic Sequences. The GENERIC_SEQ_CTRL register defines which parts of sequences are executed.

Note 2. The value given in the brackets relates to the small block mode. In this mode, the controller sends only a single byte as the column address.

Note 3. IMD_SEQ - The command will be sent immediately.

Note: Gray rows – read from NAND Flash memory; White rows – write to NAND Flash memory; Blue rows – Non-directional commands

(1) Sequence SEQ_0

This non-directional sequence is composed from only one command. After the command is written to the NAND Flash device, the controller waits until the device goes into the busy state and drives the RnB line low, or sends the READ STATUS command. When delay time (tWB) passes or the device is ready, the sequence ends. The figure below shows the sequence.

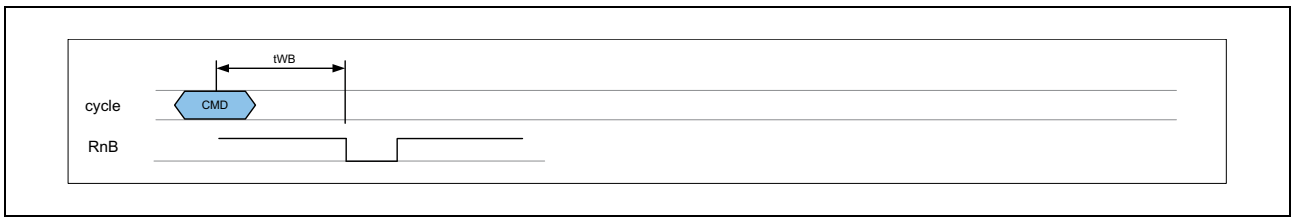


Figure 58.2 SEQ_0 Sequence

(2) Sequence SEQ_1

This is a read-sequence that is composed from a single command cycle, single address cycle and the single data cycle with a programmable number of read sequences. After the address sequence is finished, the controller measures the standard delay of first data read after the last write (tWHR). Next, the read data words are written to the FIFO module. The input module is selected by the INPUT_SEL field of the COMMAND register, the source for the address is placed in the ADDR0_COL register, and the command code is stored in the CMD_0 field. The figure below shows the sequence execution:

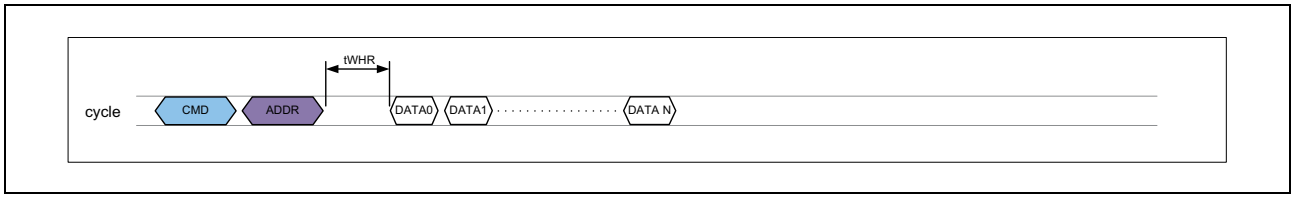


Figure 58.3 SEQ_1 Sequence

(3) Sequence SEQ_2

This is a read-sequence and it is similar to the SEQ_1 sequence except that after the address cycle the controller expects that device goes to the busy state. The controller sequentially checks state of the RnB line or send READ STATUS command to obtain the NAND Flash device status. The figure below shows the sequence execution:

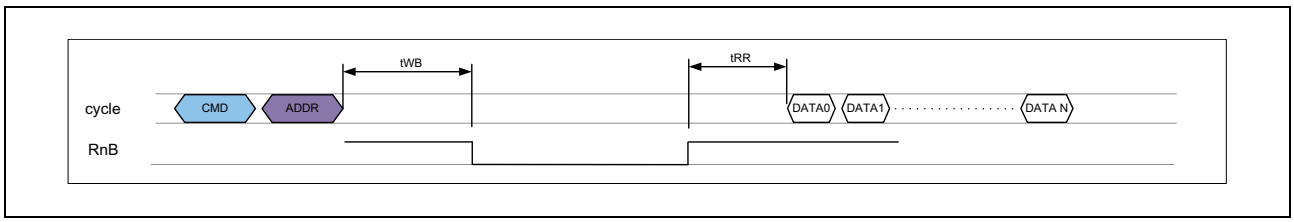


Figure 58.4 SEQ_2 Sequence

(4) Sequence SEQ_3

This is a write-sequence that is composed from a single command cycle, single address cycle and single data cycle with a programmable number of write sequences. After the address sequence is finished, the controller measures the standard delay of first data write after the last address cycle (tADL). The written words are read from the FIFO module. The controller sequentially checks the state of the RnB line or sends the READ STATUS command to obtain the NAND Flash device status. The input module is selected by the INPUT_SEL field of the COMMAND register; the source for the address is placed in the ADDR0_COL register, and the commend code is stored in the CMD_0 field. The figure below shows the sequence execution:

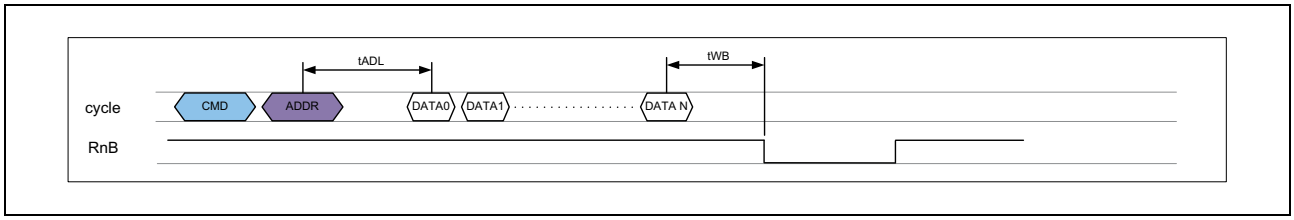


Figure 58.5 SEQ_3 Sequence

(5) Sequence SEQ_4

This is a special read-sequence that is used to implement the read status command sequences. The command is sent immediately. The sequence is composed of a single command cycle and a single data cycle. Between those cycles, the delay is counted (tWHR). The command code is read from the CMD_0 field.

When the DATA register is selected in the COMMAND register, the data is stored in the DATA register. The user defines the number of data in the DATA_REG_SIZE register. The registered mode is allowed only for the read direction.

When the FIFO register is selected in the COMMAND register, the data is stored in the FIFO. Because user has to change DATA_SIZE register, the command will be sent when all memories are ready and the Controller is in the idle state. The figure below shows the sequence execution:

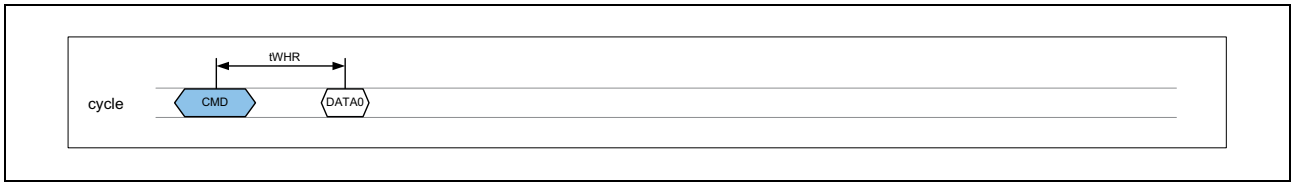


Figure 58.6 SEQ_4 Sequence

(6) Sequence SEQ_5

This is a read-sequence and it is similar to the SEQ_4 sequence. The command is sent immediately. The only difference is that after the command cycle, an additional address cycle is performed. The ADDR0_ROW register is used in this sequence. The figure below shows the sequence execution:

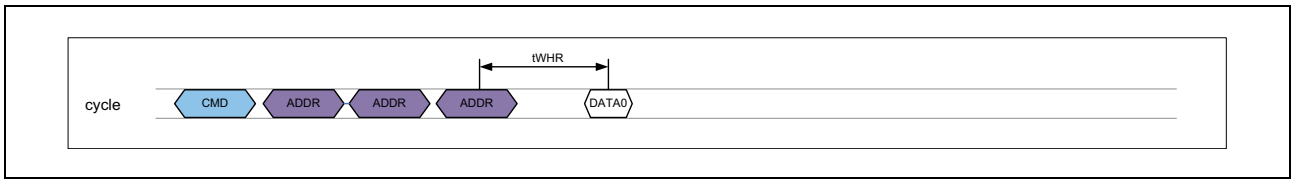


Figure 58.7 SEQ_5 Sequence

(7) Sequence SEQ_6

This is a read-sequence. The sequence of command cycle, address cycle, command cycle is executed. After that, the delay from the change column to the next operation (tCCS) is measured. Finally, the read data cycle is executed. The first command code is encoded in the CMD_0 instruction field; the second command code is encoded in the CMD_2 instruction field; the ADDR0_COL register is used in this sequence; the input module is selected by the INPUT_SEL field. The figure below shows the sequence execution:

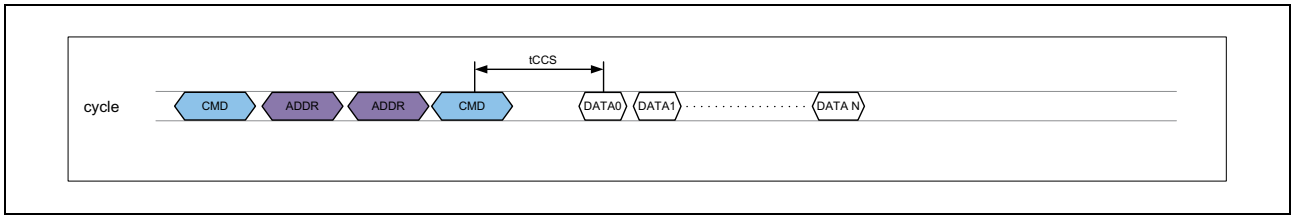


Figure 58.8 SEQ_6 Sequence

(8) Sequence SEQ_7

This read-sequence is similar to the SEQ_6 sequence, differing only because the address cycle in this sequence is composed of five bytes (ADDR0_COL[15:0] and ADDR0_ROW[23:0]) instead of three bytes. All else is the same as in the SEQ_6 sequence. The figure below shows the sequence execution:

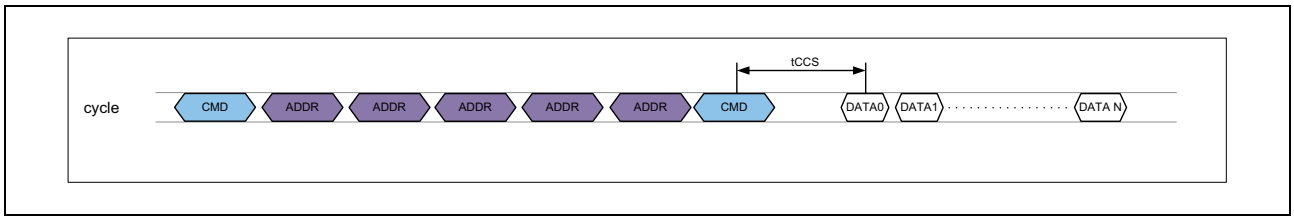


Figure 58.9 SEQ_7 Sequence

(9) Sequence SEQ_8

This is a write-sequence. First, the sequence of command cycle and two bytes address cycle is executed. Next, the delay after the column address changes (tCCS) is measured. Finally, the single data cycle with programmable number of write sequences is executed. The first command code is encoded in the CMD_0 instruction field; the ADDR0_COL register is used in this sequence; the input module is selected by the INPUT_SEL field. The figure below shows the sequence execution:

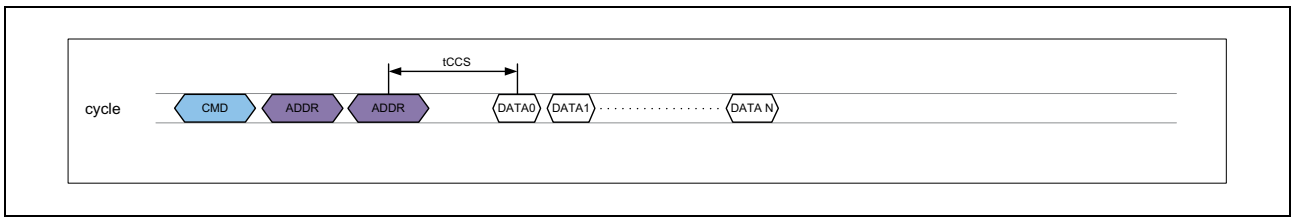


Figure 58.10 SEQ_8 Sequence

(10) Sequence SEQ_9

This is a non-directional sequence. The first step is to execute the five bytes address command cycle. The controller sequentially checks the state of the RnB line or sends the READ STATUS command to obtain the NAND Flash device status. The first command code is encoded in the CMD_0 instruction field; the second command code is encoded in the CMD_1 instruction field; the ADDR0_COL and ADDR0_ROW registers are used in this sequence. The figure below shows the sequence execution:

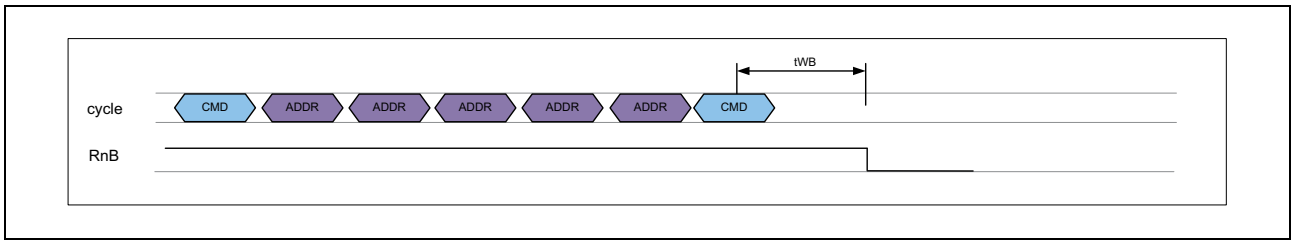


Figure 58.11 SEQ_9 Sequence

(11) Sequence SEQ_10

This is a non-directional sequence. The first step is to execute the five bytes address command cycle. The controller sequentially checks the state of the RnB line or sends the READ STATUS command to obtain the NAND Flash device status. Finally data block is read. The first command code is encoded in the CMD_0 instruction field; the second command code is encoded in the CMD_2 instruction field; the ADDR0_COL and ADDR0_ROW registers are used in this sequence. The figure below shows the sequence execution:

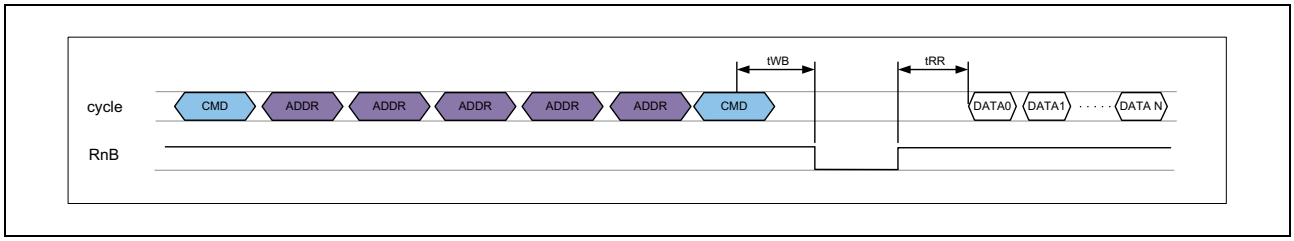


Figure 58.12 SEQ_10 Sequence

(12) Sequence SEQ_11

This is the read-sequence. The first step is to execute the command cycle. Next, the device goes to the busy state. The controller sequentially checks the state of the RnB line or sends the READ STATUS command to obtain the NAND Flash device status. As soon as the device reaches the ready state, the write data cycle with configurable read sequences is executed. The command code is encoded in the CMD_0 instruction field; the input module is selected by the INPUT_SEL field. The number of transferred bytes are configured using the DATA_SIZE register. The figure below shows the sequence execution:

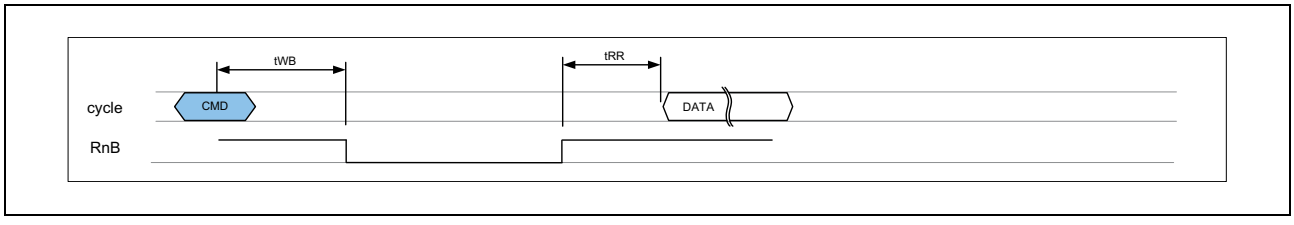


Figure 58.13 SEQ_11 Sequence

(13) SEQ_12 Sequence

This is a write-sequence. The SEQ_12 sequence is a series of command cycle, address cycle, and data cycle with a configurable number of write operations and another command cycle. Between the last address cycle and first data cycle, a delay is measured (tADL) and, after the second command cycle, another delay is measured (tWB). The first command code is encoded in the CMD_0 instruction field; the second command code is encoded in the CMD_1 instruction field; the ADDR0_COL and ADDR0_ROW registers are used in this sequence; the input module is selected by the INPUT_SEL field. The figure below shows the sequence execution:

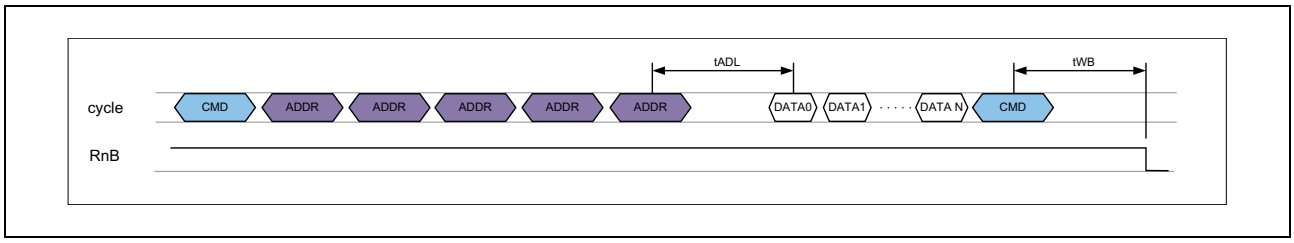


Figure 58.14 SEQ_12 Sequence

(14) SEQ_13 Sequence

This is a write-sequence. The SEQ_13 sequence is a series of command cycle and address cycles, data cycle with a configurable number of write operations. Between the last address cycle and first data cycle, a delay is measured (tADL). The command code is encoded in the CMD_0 instruction field; the ADDR0_COL and ADDR0_ROW registers are used in this sequence; the input module is selected by the INPUT_SEL field. The figure below shows the sequence execution:

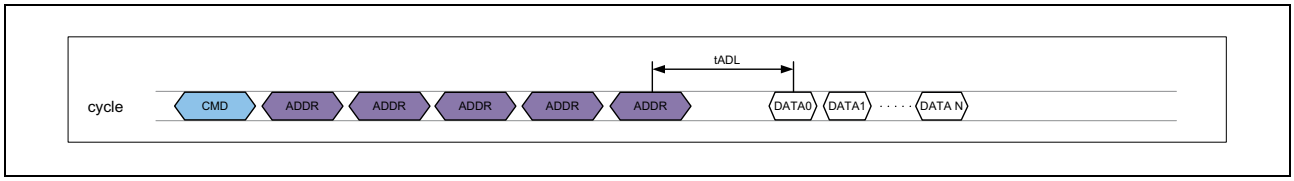


Figure 58.15 SEQ_13 Sequence

(15) SEQ_14 Sequence

This is a non-directional sequence. First, the series of command cycle, address cycle, command cycle is executed. The controller sequentially checks the state of the RnB line or sends the READ STATUS command to obtain the NAND Flash device status. The first command code is encoded in the CMD_0 instruction field; the second command code is encoded in the CMD_1 instruction field; the ADDR0_ROW and ADDR0_COL registers are used in this sequence. The figure below shows the sequence execution:

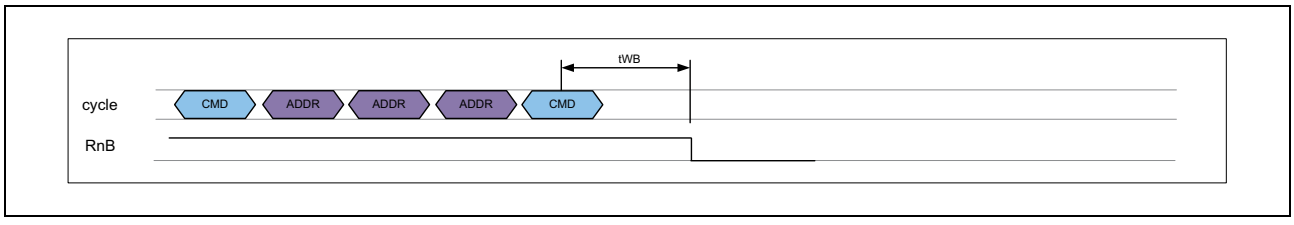


Figure 58.16 SEQ_14 Sequence

(16) SEQ_15 sequence

This is a read-sequence. First, the series of command cycle, address cycle, second command cycle, second address cycle, third command cycle is executed. The controller sequentially checks the state of the RnB line or sends the READ STATUS command to obtain the NAND Flash device status. After the NAND Flash device returns to the ready state, the data sequence, with a configurable number of read operations, is executed. The first command code is encoded in the CMD_0 instruction field; the second command code is encoded in the CMD_3 instruction field; the third command code is encoded in the CMD_2 instruction field. In this sequence, both address registers are used. The ADDR0_ROW and ADDR0_COL register content is sent after the first command in the sequence, the ADDR1_ROW and ADDR1_COL register content is sent after the second command in the sequence. The figure below shows the sequence execution:

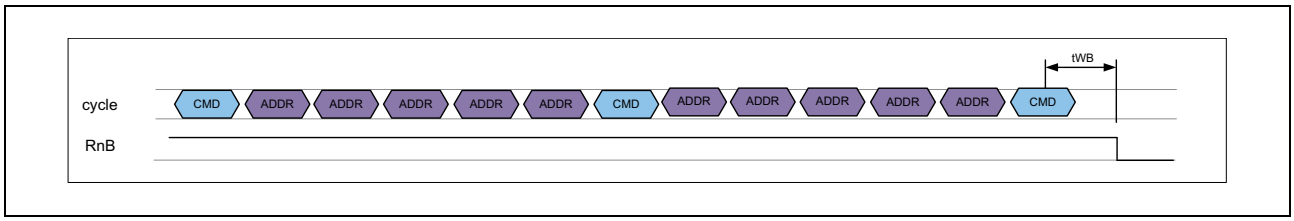


Figure 58.17 SEQ_15 Sequence

(17) SEQ_17 sequence

This is a read-sequence. This sequence is similar to the SEQ_10 sequence, except that the second command cycle is omitted. This sequence is implemented to use small block memories. The controller sends only four bytes of the address when the small block mode is enabled. The figure below shows the sequence execution:

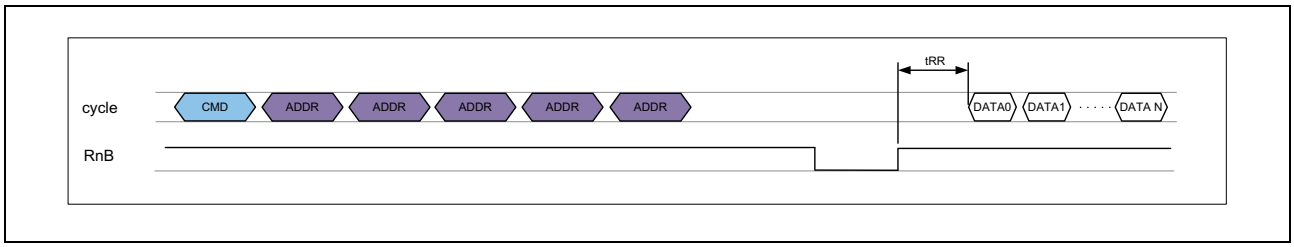


Figure 58.18 SEQ_17 Sequence

(18) SEQ_18 sequence

This generic read sequence is described in detail in Section 58.4.2, Generic Command Sequence.

(19) SEQ_19 sequence

This generic write sequence is described in detail in Section 58.4.2, Generic Command Sequence.

(20) SEQ_20 sequence

This non-directional sequence is composed from one command and three addresses bytes. After the command and addresses are written to the NAND Flash device, the controller waits until the device goes into the busy state and drives the RnB line low, or sends the READ STATUS command. When delay time (tWB) passes or the device is ready, the sequence ends. The figure below shows the sequence execution:

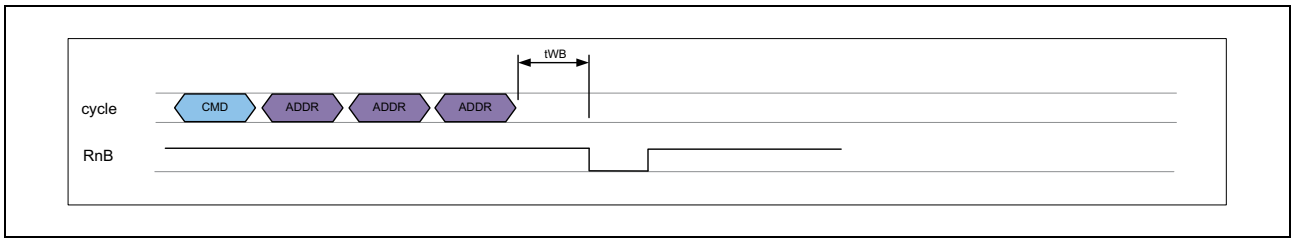


Figure 58.19 SEQ_20 Sequence

(21) SEQ_21 sequence

This non-directional sequence is composed from one command and one addresses byte. After the command and address are written to the NAND Flash device, the sequence ends. The figure below shows the sequence execution:

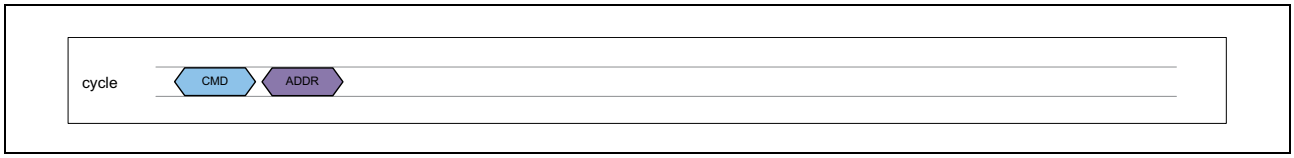


Figure 58.20 SEQ_21 Sequence

(22) SEQ_22 sequence

This is a read-sequence. The first step is to execute the five bytes address command cycle. The controller sequentially checks the state of the RnB line or sends the READ STATUS command to obtain the NAND Flash device status. The first command code is encoded in the CMD_0 instruction field; the second command code is encoded in the CMD_2 instruction field; the ADDR0_COL, ADDR1_COL and ADDR1_ROW registers are used in this sequence. The figure below shows the sequence execution:

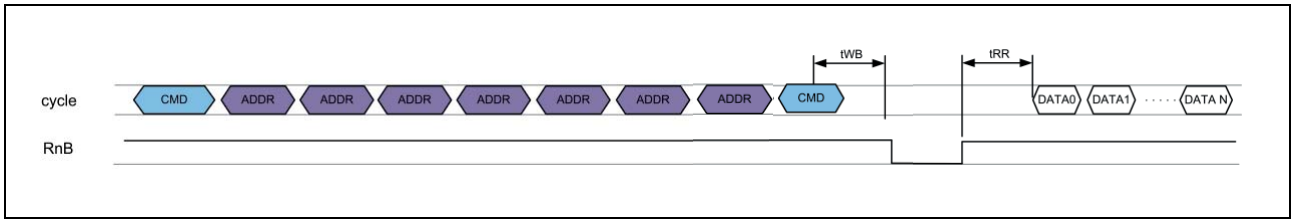


Figure 58.21 SEQ_22 Sequence

(23) SEQ_23 sequence

This is a write-sequence. The SEQ_23 sequence is a series of command cycle, address cycle, and data cycle with a configurable number of write operations and another command cycle. Between the last address cycle and first data cycle, a delay is measured (tADL) and, after the second command cycle, another delay is measured (tWB). The first command code is encoded in the CMD_0 instruction field; the second command code is encoded in the CMD_1 instruction field; the ADDR0_ROW registers is used in this sequence; the input module is selected by the INPUT_SEL field. The figure below shows the sequence execution:

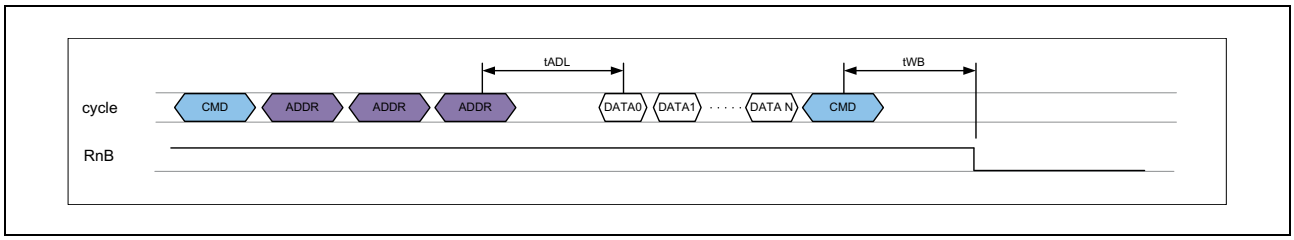


Figure 58.22 SEQ_23Sequence

(24) SEQ_24 sequence

This is a write-sequence. It is composed from the three commands cycles and two addresses cycles. Both addresses cycle contain the row address part. After the last command cycle the tWB delay is measured. The figure below shows the sequence execution.

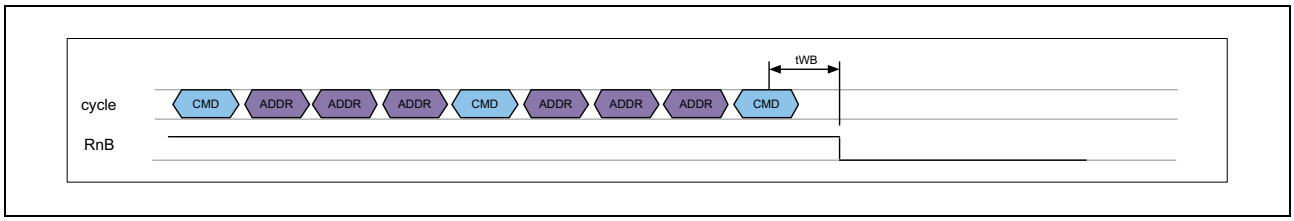


Figure 58.23 SEQ_24 Sequence

(25) SEQ_25 sequence

This is a read-sequence. It is composed from the three commands cycles and two addresses cycles. the first addresses cycle contain the column and row address part. The second address cycle contain only the column address part. After the last command cycle the tWHR delay is measured. The figure below shows the sequence execution.

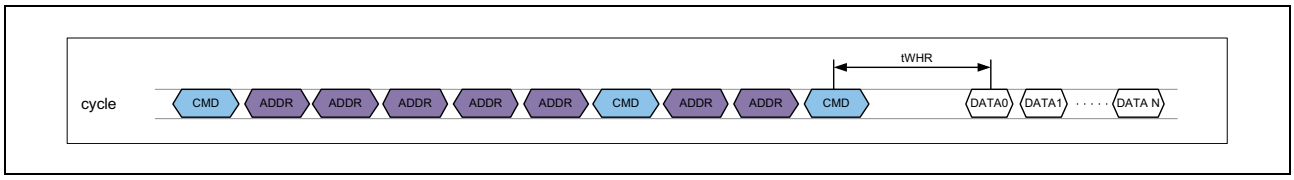


Figure 58.24 SEQ_25 Sequence

(26) SEQ_26 sequence

This is a write-sequence. The SEQ_26 sequence is a series of command cycle, address cycle, and data cycle with a configurable number of write operations and another command cycle. Between the last address cycle and first data cycle, a delay is measured (tADL). The first command code is encoded in the CMD_0 instruction field; the second command code is encoded in the CMD_1 instruction field; the ADDR0_ROW registers is used in this sequence; the input module is selected by the INPUT_SEL field. The figure below shows the sequence execution:

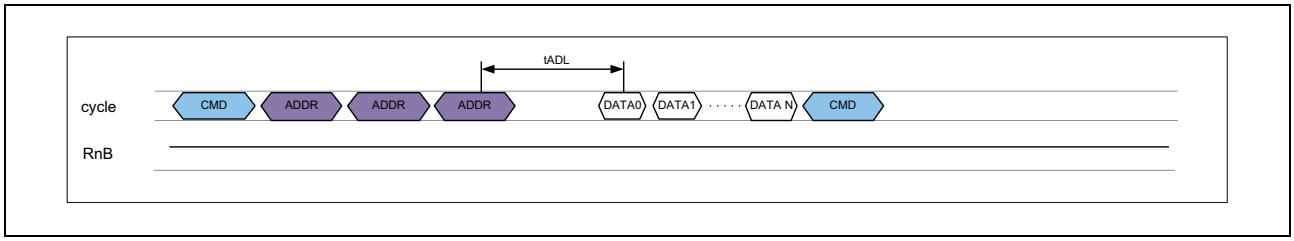


Figure 58.25 SEQ_26 Sequence

58.4.2 Generic Command Sequence

There will be cases where the set of predefined sequences above will not be sufficient to handle a new command sequence. If that occurs, the generic command sequence feature of the NAND flash controller can be used.

This sequence is designed to mimic almost every available command supported by the NAND Flash devices; however, additional effort required to trigger such commands.

Generic command sequence is executed in the following steps:

CMD0 – The first command in the sequence. The value of this command is stored in the CMD_0 field of the COMMAND register, described in **Table 58.9**.

ADDR0 – The first address sequence. This is enabled if the COL_A0 and ROW_A0 fields in the GEN_SEQ_CTRL register described in **Table 58.11** have values other than zero. In this phase, the address is sent to the NAND Flash device and is read from the ADDR0_COL and ADDR0_ROW registers. The number of the bytes in the address cycle is configured by the COL_A0 and ROW_A0 fields of the GEN_SEQ_CTRL register.

CMD1 – The fourth command in the sequence. The value of this command is stored in the CMD_1 field of the COMMAND register, described in **Table 58.9**. This is enabled by the CMD1_EN field in the GENERIC_SEQ_CTRL register, described in **Table 58.11**.

ADDR1 – The second address in the sequence. This is enabled if the COL_A1 and ROW_A1 fields of the GEN_SEQ_CTRL register described in **Table 58.11** have a value other than zero. In this phase, the address is sent to the NAND Flash device from the ADDR1_COL and ADDR1_ROW registers. The number of bytes in the address cycle is configured by the COL_A1 and ROW_A1 fields of the GEN_SEQ_CTRL register.

CMD2 – The third command in the sequence. The value of this command is stored in the CMD_2 field of the COMMAND register described in **Table 58.9**. This is enabled by the CMD2_EN field in the GEN_SEQ_CTRL register, described in **Table 58.11**.

DELAY0 – Waiting for the device to return to the ready state and continue the sequence. This is enabled by the DELAY_EN field in the GEN_SEQ_CTRL register, described in **Table 58.11**. Only one delay phase can be present in the generic sequence. For more details, see **Table 58.11**.

DATA – The data phase of the sequence. This is enabled by the DATA_EN field in the GEN_SEQ_CTRL register described in **Table 58.11**. Additionally, the transfer direction must be selected by the sequence number. Sequence number 18 reads data from the NAND Flash memory, sequence number 19 writes data to the NAND Flash memory. The size of the transferred data block is configured by the DATA_SIZE register value.

CMD3 – The second command in the sequence. This is enabled by the CMD3_EN field in the GEN_SEQ_CTRL register described in the **Table 58.11**. The value of this command is stored in the CMD_3 field of the GEN_SEQ_CTRL register.

DELAY1 – Waiting for the device to return to the ready state and finish the sequence. This is enabled by the DELAY_EN field in the GENERIC_SEQ_CTRL register described in **Table 58.11**. The controller waits for the device to return to the ready state and finish the sequence. Only one delay phase can be present in generic sequence.

The figure below presents the generic sequence composition:

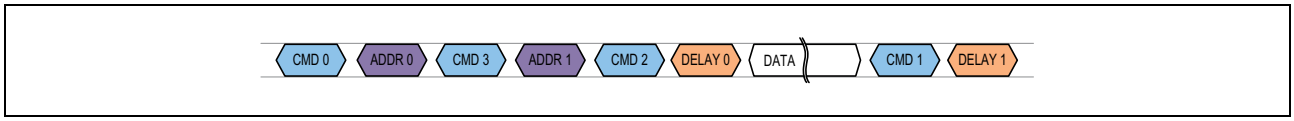


Figure 58.26 Generic Sequence

There are few constraints on the generic sequence usage:

- The DEL0 and DEL1 delay phases cannot both be enabled in a single command sequence
- The TIMINGS _ASYN register must be set even when the generic sequence is used.

It is possible to force an immediate command sequence execution by enabling the IMD_SEQ bit in the GEN_SEQ_CTRL register. In this case, the triggered command will be executed even if the previously sent command for a selected device is not completed. If both the IMD_SEQ and DATA_EN flags are enabled in a single sequence, then the register must be selected as data source/sink. The IMD_SEQ is valid only for the read direction. This feature is intended to implement all state read operations.

After each step of the generic sequence, the programmable time delay is measured. These delays are configured using the TIME_GEN_SEQ[0-3] registers. Refer to the TIME_GEN_SEQ[0-3] registers description for further information and see figures below.

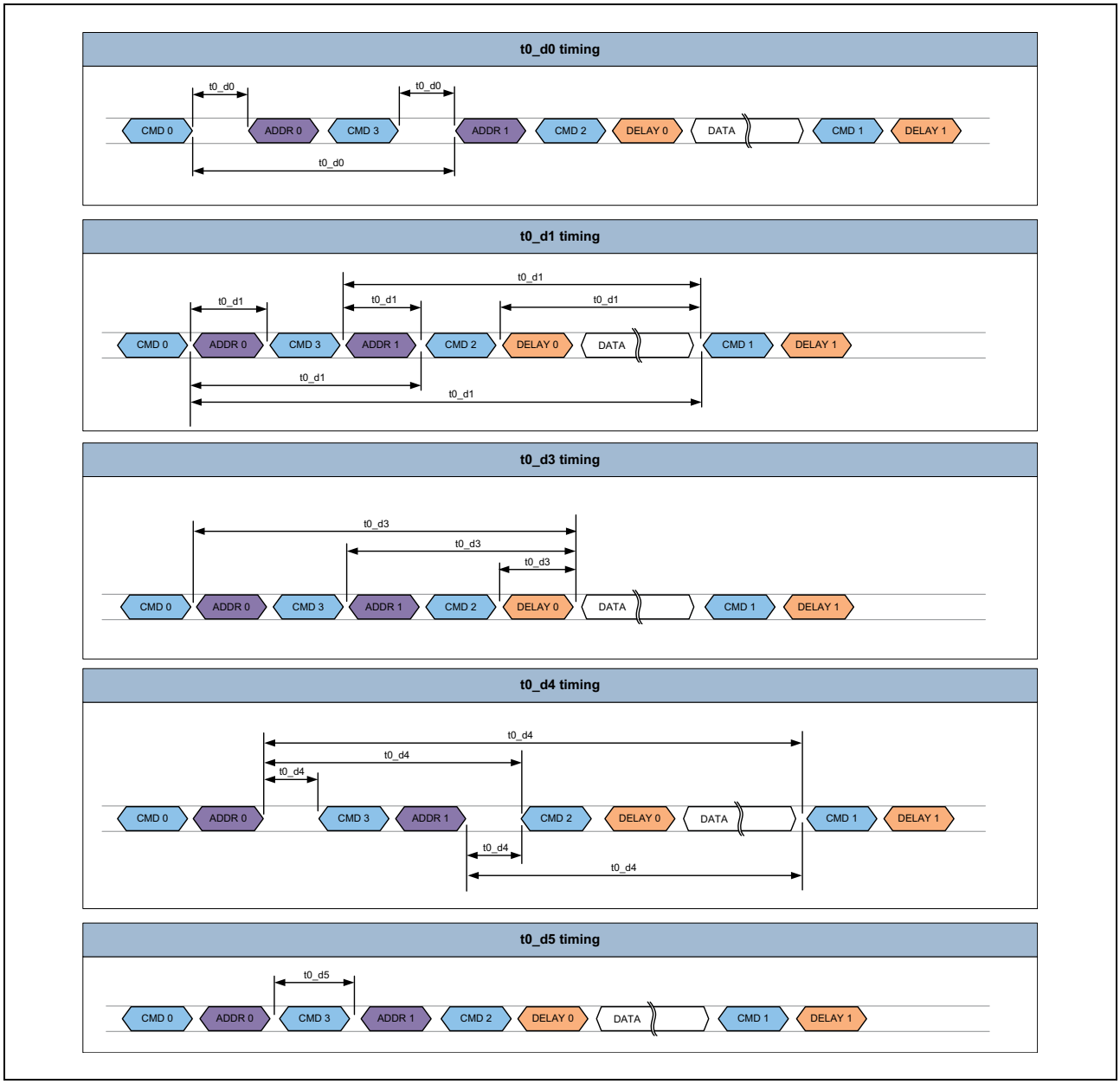


Figure 58.27 Generic Sequence Timing parameters

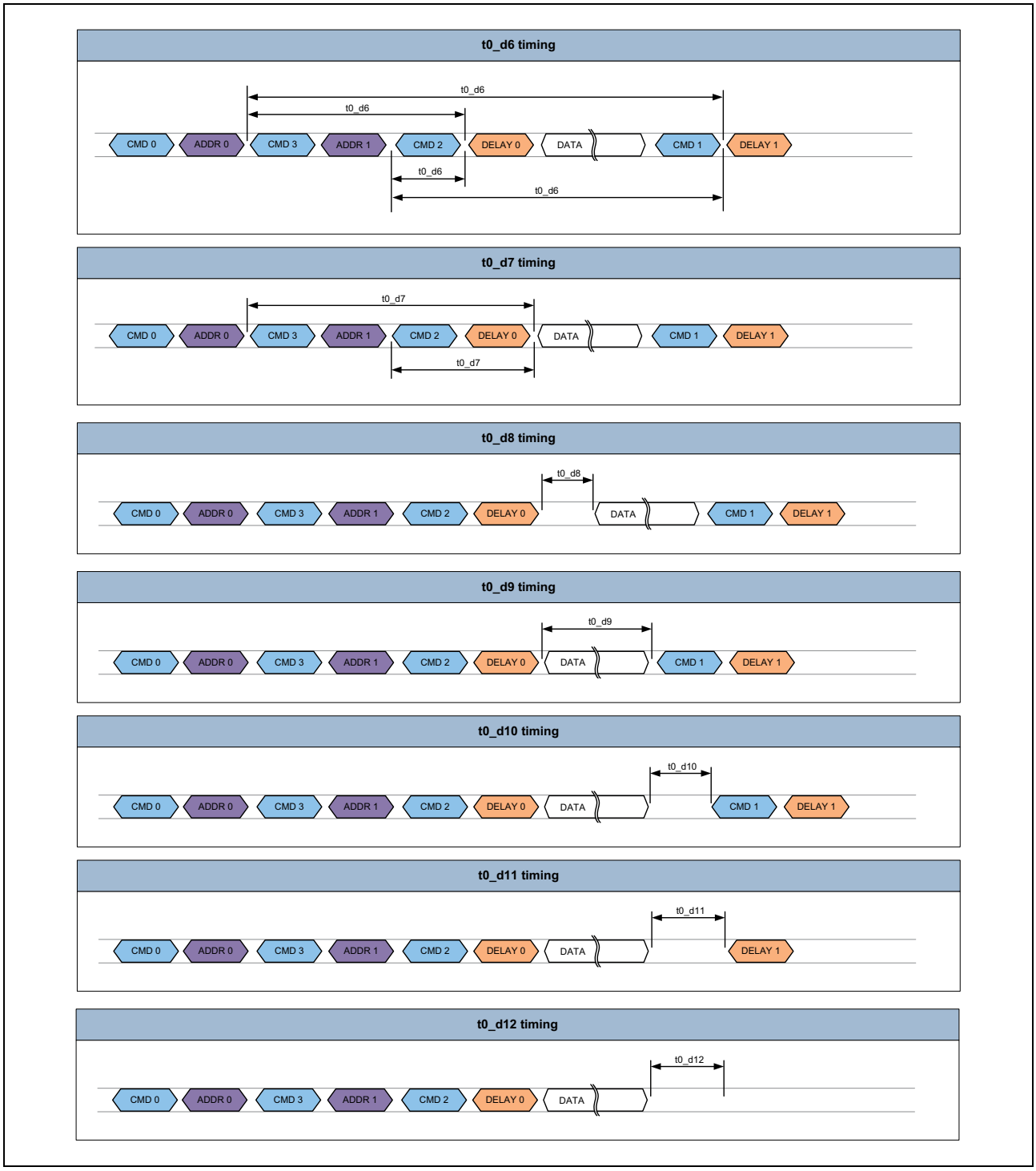


Figure 58.28 Generic Sequence Timing parameters

58.4.3 Instructions

The implementation of the instruction set presented at this point is an example of how to use the instruction encoding scheme presented in Section 58.4.1.1, Instruction Encoding. This chapter and the command sequences presented previously give sufficient information to implement new commands and command sequences for future NAND Flash devices.

58.4.3.1 Instructions Set

The table below contains the basic instruction defined to implement the all command sequences accessible in the ONFI 1.x standards.

Table 58.48 Instructions set (1/2)

Instruction	CMD_0	CMD_1 / CMD_3	CMD_2	CMD_SEQ	Send when memory is busy
Reset Commands					
RESET	0xFF	-	-	SEQ_0	No
Identification Operations					
READ ID	0x90	-	-	SEQ_1	No
READ PARAMETER PAGE	0xEC	-	-	SEQ_2	No
READ UNIQUE ID	0xED	-	-	SEQ_2	No
Configuration Operations					
GET FEATURES	0xEE	-	-	SEQ_2	No
SET FEATURES	0xEF	-	-	SEQ_3	No
Status Operations					
READ STATUS	0x70	-	-	SEQ_4	Yes
SELECT LUN WITH STATUS	0x78	-	-	SEQ_5	Yes
LUN STATUS	0x71	-	-	SEQ_5	Yes
DEVICE STATUS	0x72	-	-	SEQ_4	Yes
VOLUME_SELECT	0xE1	-	-	SEQ21	Yes
Column Address Operations					
CHANGE READ COLUMN	0x05	-	0xE0	SEQ_6	No
SELECT CACHE REGISTER	0x06	-	0xE0	SEQ_7	No
CHANGE WRITE COLUMN	0x85	-	-	SEQ_8	No
CHANGE ROW ADDRESS	0x85	0x11	-	SEQ_12	No
Read Operations					
READ PAGE	0x00	-	0x30	SEQ_10	No
READ PAGE CACHE	0x31	-	-	SEQ_11	No
READ PAGE CACHE LAST	0x3F	-	-	SEQ_11	No
READ MULTIPLANE	0x00	0x32	-	SEQ_9	No
TWO PLANE PAGE READ	0x00	0x00	0x30	SEQ_15	No
QUEUE PAGE READ	0x07	-	0x37	SEQ22	No
Program Operation					
PROGRAM PAGE	0x80	0x10	-	SEQ_12	No
PROGRAM PAGE IMD	0x80	0x10	-	SEQ_23	No
PROGRAM PAGE DEL	0x80	0x13	-	SEQ_23	No
PROGRAM PAGE 1	0x80	-	-	SEQ_13	No
PROGRAM PAGE CACHE	0x80	0x15	-	SEQ_12	No

Table 58.48 Instructions set (2/2)

Instruction	CMD_0	CMD_1 / CMD_3	CMD_2	CMD_SEQ	Send when memory is busy
PROGRAM MULTIPLANE	0x80	0x11	-	SEQ_12	No
WRITE PAGE	0x10	-	-	SEQ_0	No
WRITE PAGE CACHE	0x15	-	-	SEQ_0	No
WRITE MULTIPLANE	0x11	-	-	SEQ_0	No
Erase Operation					
ERASE BLOCK	0x60	0xD0	-	SEQ_14	No
ERASE MULTIPLANE	0x60	0xD1	-	SEQ_14	No
Copyback Operation					
COPYBACK READ	0x00	-	0x35	SEQ_10	No
COPYBACK PROGRAM	0x85	0x10	-	SEQ_9	No
COPYBACK PROGRAM 1	0x85	-	-	SEQ_13	No
COPYBACK MULTIPLANE	0x85	0x11	-	SEQ_12	No
OTP Operation					
PROGRAM OTP	0xA0	0x10	-	SEQ_12	No
DATA PROTECT OTP	0xA5	0x10	-	SEQ_9	No
READ PAGE OTP	0xAF	-	0x30	SEQ_10	No

58.4.3.2 RESET Command

(1) Command Description

The RESET command is used to put a target into a known condition and to abort command sequences in progress.

(2) Command Encoding

The RESET instruction uses the SEQ_0 commands sequence. The command is encoded, as shown in the table below:

Table 58.49 RESET Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0xFF	-	-	SEQ_0_ID

58.4.3.3 READ ID Command

(1) Command Description

The READ ID command is used to read identifier codes programmed into the target. This command is accepted by the target only when all LUNs on the target are in the IDLE state.

When the command is followed by an address cycle of 0x00, the target returns a 5-byte identifier code that includes the manufacturer's ID, device configuration, and part-specific information.

When the READ ID command is followed by an address cycle of 0x20, the target returns the 4-byte ONFI identifier code.

(2) Command Encoding

The READ ID instruction uses the SEQ_1 commands sequence. The command is encoded, as shown in the table below:

Table 58.50 READ ID Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x90	0/1	0/1	SEQ_1_ID

58.4.3.4 READ PARAMETER PAGE Command

(1) Command Description

The READ PARAMETER PAGE command is used to read the ONFI parameter page programmed into the target. This command is accepted by the target only when all LUNs on the target are idle.

When the command is followed by an address cycle of 00h, the target goes into busy state. After the read process is completed, the controller enables the data output mode to read the parameter page.

(2) Command Encoding

The READ PARAMETER PAGE instruction uses the SEQ_2 commands sequence. The command is encoded, as shown in the table below:

Table 58.51 READ PARAMETER PAGE Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0xEC	0/1	0/1	SEQ_2_ID

58.4.3.5 READ UNIQUE ID Command

(1) Command Description

The READ UNIQUE ID instruction is used to read a unique identifier programmed into the target. This command is accepted by the target only when all LUNs on the target are idle.

When the address cycle of 00h is written to the target, then the target goes into busy state. After the read process is complete, the controller enables the data output mode to read the unique ID. Sixteen copies of the unique ID data are stored in the device. Each copy is 32 bytes. The first 16 bytes are unique data and the second 16 bytes are the complement of the first 16 bytes. The application program will XOR the first 16 bytes with the second 16 bytes. If the result is 16 bytes of 0xFF, then that copy of the unique ID data is correct. In the event that there is a non-0xFF result, the application program repeats the XOR operation on a subsequent copy of the unique ID data.

(2) Command Encoding

The READ UNIQUE ID instruction uses the SEQ_2 commands sequence. The command is encoded, as shown in the table below:

Table 58.52 READ UNIQUE ID Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0xED	0/1	0/1	SEQ_2_ID

58.4.3.6 GET FEATURES Command

(1) Command Description

The GET FEATURES instruction reads the sub-feature parameters (P1-P4) from the specified feature address. This command is accepted by the target only when all LUNs on the target are idle.

When the 0xEE command is followed by a feature address, the target goes into busy state. After the target internal read operation completes, the controller enables the data output mode to read the sub-feature parameters.

(2) Command Encoding

The GET FEATURES instruction uses the SEQ_2 commands sequence. The command is encoded, as shown in the table below:

Table 58.53 GET FEATURES Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0xEE	0/1	0/1	SEQ_2_ID

58.4.3.7 SET FEATURES Command

(1) Command Description

The SET FEATURES instruction writes the sub-feature parameters (P1-P4) to the specified feature address to enable or disable target-specific features. This command is accepted by the target only when all LUNs on the target are idle.

The 0xEF command is followed by a valid feature address. The possible address value depends on the features set implemented in the target device. The address cycle is followed by the configurable number of data cycles. Values of the address and data encoding scheme allowed are found in the device vendor documentation.

(2) Command Encoding

The SET FEATURES instruction uses the SEQ_3 commands sequence. The command is encoded, as shown in the table below:

Table 58.54 SET FEATURES Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0xEF	-	0/1	SEQ_3_ID

58.4.3.8 READ STATUS Command

(1) Command Description

Each LUN provides its status independently of other LUNs on the same target through its 8-bit status register. Once the READ STATUS instruction is issued, status register output is enabled. The contents of the status register are returned on DQ[7: 0] for each data output request.

The READ STATUS command returns the status of the most recently selected LUN.

(2) Command Encoding

The READ STATUS instruction uses the SEQ_4 commands sequence. The command is encoded, as shown in the table below:

Table 58.55 READ STATUS Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x70	0/1	-	SEQ_4_ID

This instruction has special meaning because it can be executed when the target is in the BUSY state. The FIFO cannot be used to access the read data because it can be occupied by the operation under execution. For the sequence the data field of command must select the DATA_REG register as data destination. The DATA_REG_SIZE must be select single byte.

58.4.3.9 DEVICE STATUS Command

(1) Command Description

Each LUN provides its status independently of other LUNs on the same target through its 8-bit status register. Once the DEVICE STATUS instruction is issued, status register output is enabled. The contents of the status register are returned on DQ[7: 0] for each data output request.

The DEVICE STATUS command returns the status of the most recently selected LUN.

(2) Command Encoding

The DEVICE STATUS instruction uses the SEQ_4 commands sequence. The command is encoded, as shown in the table below:

Table 58.56 DEVICE STATUS Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x72	0/1	-	SEQ_4_ID

This instruction has special meaning because it can be executed when the target is in the BUSY state. The FIFO cannot be used to access the read data because it can be occupied by the operation under execution. For the sequence the data field of command must select the DATA_REG register as data destination. The DATA_REG_SIZE must be select single byte.

58.4.3.10 VOLUME SELECT Command

(1) Command Description

The VOLUME SELECT command is used to select a particular volume based on the address specified. This command is accepted by all initialized devices that share a CE pin. The command may be executed with any volume on the target in any state. When the VOLUME SELECT command is issued, all volumes with unselected volume addresses will be deselected to save power. If the Volume address entered is invalid or does not match any appointed volume address, all volume addresses will be deselected. If the VOLUME SELECT command is not issued after CE high time then all volumes revert to their previous state.

(2) Command Encoding

The VOLUME SELECT instruction uses the SEQ_21 commands sequence. The command is encoded, as shown in the table below:

Table 58.57 VOLUME SELECT Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0xE1	0/1	-	SEQ_21_ID

58.4.3.11 SELECT LUN WITH STATUS Command

(1) Command Description

Each LUN provides its status independently of other LUNs on the same target through its 8-bit status register. Once the SELECT LUN WITH STATUS instruction is issued, status register output is enabled. The contents of the status register are returned on DQ[7: 0] for each data output request.

The SELECT LUN WITH STATUS command returns the status of the selected LUN.

(2) Command Encoding

The SELECT LUN WITH STATUS instruction uses the SEQ_5 commands sequence. The command is encoded, as shown in the table below:

Table 58.58 SELECT LUN WITH STATUS Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x78	0/1	-	SEQ_5_ID

This instruction has special meaning because it can be executed when the target is in the BUSY state. The FIFO cannot be used to access the read data because it can be occupied by the operation under execution. For the sequence the data field of command must select the DATA_REG register as data destination. The DATA_REG_SIZE must be select single byte.

58.4.3.12 LUN STATUS Command

(1) Command Description

Each LUN provides its status independently of other LUNs on the same target through its 8-bit status register. Once the SELECT LUN WITH STATUS instruction is issued, status register output is enabled. The contents of the status register are returned on DQ[7: 0] for each data output request.

The LUN STATUS command returns the status of the selected LUN.

(2) Command Encoding

The LUN STATUS instruction uses the SEQ_5 commands sequence. The command is encoded, as shown in the table below:

Table 58.59 SELECT LUN WITH STATUS Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x71	0/1	-	SEQ_5_ID

This instruction has special meaning because it can be executed when the target is in the BUSY state. The FIFO cannot be used to access the read data because it can be occupied by the operation under execution. For the sequence the data field of command must select the DATA_REG register as data destination. The DATA_REG_SIZE must be select single byte.

58.4.3.13 CHANGE READ COLUMN Command

(1) Command Description

The CHANGE READ COLUMN command changes the column address of the selected cache register and enables data output of the last selected LUN. This command is accepted by the selected LUN when it is ready. Writing 0x05 to the target command register, followed by two column address cycles containing the column address, followed by the 0xE0 command puts the selected LUN into data output mode. The selected LUN stays in data output mode until another valid command is issued.

(2) Command Encoding

The CHANGE READ COLUMN instruction uses the SEQ_6 commands sequence. The command is encoded, as shown in the table below:

Table 58.60 CHANGE READ COLUMN Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
0xE0	-	0x05	0/1	-	SEQ_6_ID

58.4.3.14 SELECT CACHE REGISTER Command

(1) Command Description

The SELECT CACHE REGISTER command enables data output on the addressed LUN and cache register at the specified column address. This command is accepted by a LUN when it is ready. Writing the 0x06 to the target internal command register, followed by two column address cycles and three row address cycles, followed by 0xE0 enables data output mode on the address LUN and cache register at the specified column address.

(2) Command Encoding

The SELECT CACHE REGISTER instruction uses the SEQ_7 commands sequence. The command is encoded, as shown in the table below:

Table 58.61 SELECT CACHE REGISTER Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
0xE0	-	0x06	0/1	-	SEQ_7_ID

58.4.3.15 CHANGE WRITE COLUMN Command

(1) Command Description

The CHANGE WRITE COLUMN command changes the column address of the selected cache register and enables data input on the last selected LUN. Writing the 0x85 to the target internal command register, followed by two column address cycles containing the column address puts the selected LUN into data input mode.

(2) Command Encoding

The CHANGE WRITE COLUMN instruction uses the SEQ_8 commands sequence. The command is encoded, as shown in the table below:

Table 58.62 CHANGE WRITE COLUMN Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x85	-	-	SEQ_8_ID

58.4.3.16 CHANGE ROW ADDRESS Command

(1) Command Description

The CHANGE ROW ADDRESS command changes the row address (block and page) where the cache register contents are to be programmed in the NAND array. It also changes the column address of the selected cache register and enables data input on the specified LUN.

(2) Command Encoding

The CHANGE ROW ADDRESS instruction uses the SEQ_12 commands sequence. The command is encoded, as shown in the table below:

Table 58.63 CHANGE ROW ADDRESS Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0x11	0x85	0	-	SEQ_12_ID

58.4.3.17 READ PAGE Command

(1) Command Description

The READ PAGE command copies a page from the NAND Flash array to its respective cache register and enables data output. This command is accepted by the LUN when it is ready. To read a page from the NAND Flash array, the controller writes the 0x00 command to the target internal command register, then writes 5 address cycles to the address registers, and concludes with the 0x30 command. The selected LUN will go into busy state as the data is transferred. When the LUN is ready, data output is enabled for the cache register linked to the plane addressed in the READ PAGE command. The controller reads the programmed number of bytes to the FIFO.

(2) Command Encoding

The READ PAGE instruction uses the SEQ_10 commands sequence. The command is encoded, as shown in the table below:

Table 58.64 READ PAGE Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
0x30	-	0x00	0/1	0/1	SEQ_10_ID

58.4.3.18 READ PAGE CACHE Command

(1) Command Description

The READ PAGE CACHE command reads the next sequential page within a block into the data register, while the previous page is output from the cache register. To issue this command, the controller writes 0x31 to the target internal command register. After this command is issued, the R/B# goes LOW and the LUN goes into busy state. Afterwards, the R/B# goes HIGH and the LUN is busy with a cache operation, indicating that the cache register is available and that the specified page is copying from the NAND Flash array to the data register. At this point, data is read from the cache register.

(2) Command Encoding

The READ PAGE CACHE instruction uses the SEQ_11 commands sequence. The command is encoded, as shown in the table below:

Table 58.65 READ PAGE CACHE Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x31	0/1	0/1	SEQ_11_ID

58.4.3.19 READ PAGE CACHE LAST Command

(1) Command Description

The READ PAGE CACHE LAST command ends the READ PAGE CACHE sequence and copies a page from the data register to the cache register. This command is accepted by the LUN when it is ready. To issue the READ PAGE CACHE LAST command, the controller writes 0x3F to the target internal command register. After this command is issued, R/B# goes LOW and the LUN goes into busy state. Afterwards, the R/B# goes HIGH and the LUN is ready. At this point, data from the targets cache register is read into the FIFO.

(2) Command Encoding

The READ PAGE CACHE LAST instruction uses the SEQ_11 commands sequence. The command is encoded, as shown in the table below:

Table 58.66 READ PAGE CACHE LAST Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x3F	0/1	0/1	SEQ_11_ID

58.4.3.20 READ MULTIPLANE Command

(1) Command Description

The READ MULTIPLANE command queues a plane to transfer data from the NAND array to its cache register. This command can be issued one or more times. Each time a new plane address is specified, that plane is also queued for data transfer. To select the final plane and to begin the read operation for all previously queued planes, issue the READ PAGE command. All queued planes will transfer data from the NAND array to their cache registers.

(2) Command Encoding

The READ MULTIPLANE instruction uses the SEQ_9 commands sequence. The command is encoded, as shown in the table below:

Table 58.67 READ MULTIPLANE Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0x32	0x00	-	-	SEQ_9_ID

58.4.3.21 QUEUE PAGE READ Command

(1) Command Description

The QUEUE PAGE READ operation allows for partial-page reads by using a 7-address cycle. The first two bytes of the address cycle indicate the length of the page to read - those bytes are stored in ADDR0 register, followed by the column and row addresses - those bytes are stored in the ADDR1 register. This can help overall performance should only a portion of the page data be needed because only the codeword containing the requested data will have ECC decoded. This command is exclusive for the CLEAR NAND devices.

(2) Command Encoding

The QUEUE PAGE READ instruction uses the SEQ_22 commands sequence. The command is encoded, as shown in the table below:

Table 58.68 QUEUE PAGE READ Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
0x37	-	0x07	-	-	SEQ_22_ID

58.4.3.22 TWO PLANE PAGE READ command

(1) Command description

This command was implemented to preserve the compatibility with the ONFI 1.x and some older devices.

The TWO PLANE PAGE READ (00h-00h-30h) operation is similar to the PAGE READ (00h-30h) operation. It transfers two pages of data from the NAND Flash array to the data registers. Each page must be from a different plane on the same die. The software is responsible for generating correct addresses for the requested pages. Both the ADDR0 and ADDR1 address registers are used in this case.

(2) Command Encoding

The TWO PLANE PAGE READ instruction uses the SEQ_15 commands sequence. The command is encoded, as shown in the table below. In this case, both the address registers are used.

Table 58.69 TWO PLANE PAGE READ Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
0x30	0x00	0x00	0/1	0/1	SEQ_15_ID

58.4.3.23 PROGRAM PAGE Command

(1) Command Description

The PROGRAM PAGE command allows the application program to input data to a cache register and moves the data from the cache register to the specified block and page address in the array of the selected LUN. This command is accepted by the LUN when it is ready. To input a page to the cache register and move it to the NAND array at the block and page address specified, the controller writes 0x80 to the target internal command register. Then five address cycles containing the column address and row address are written. Data input cycles follow. Serial data is input, beginning at the column address specified. When data input is complete, the controller writes 0x10 to the target internal command register. The selected LUN goes into the busy state.

(2) Command Encoding

The PROGRAM PAGE instruction uses the SEQ_12 commands sequence. The command is encoded, as shown in the table below:

Table 58.70 PROGRAM PAGE Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0x10	0x80	0	0/1	SEQ_12_ID

58.4.3.24 PROGRAM PAGE IMMEDIATE Command

(1) Command Description

The PROGRAM PAGE IMMEDIATE command allows the application program to input data to a cache register and moves the data from the cache register to the specified block and page address in the array of the selected LUN. This command is accepted by the LUN when it is ready. To input a page to the cache register and move it to the NAND array at the block and page address specified, the controller writes 0x80 to the target internal command register. Then five address cycles containing the column address and row address are written. Data input cycles follow. Serial data is input, beginning at the column address specified. When data input is complete, the controller writes 0x10 to the target internal command register. The selected LUN goes into the busy state. This command writes only the row address to the NAND flash device

(2) Command Encoding

The PROGRAM PAGE IMMEDIATE instruction uses the SEQ_23 commands sequence. The command is encoded, as shown in the table below:

Table 58.71 PROGRAM PAGE IMD Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0x10	0x80	0	0/1	SEQ_23_ID

58.4.3.25 PROGRAM PAGE DELAYED Command

(1) Command Description

The device's internal controller can automatically manage multi-plane programming. It does this with PROGRAM PAGE DELAYED command. When this command issued, the controller will delay issuing the program operation to the array until the address for the subsequent program operation is examined. If that operation allows the previous operation to complete as part of multi-plane operation, the controller will issue a multi-plane program to the array. Multi-plane operations are only completed if an LUN address from plane 0 is issued prior to a LUN address from plane 1. If the subsequent program operation does not allow the multi-plane operation, the controller will immediately start the previous program. It is presumed that the application program may use this command to initiate all the program operations. In this way the application program does not have to maintain information regarding multi-plane operation usage.

(2) Command Encoding

The PROGRAM PAGE DELAY instruction uses the SEQ_12 commands sequence. The command is encoded, as shown in the table below:

Table 58.72 PROGRAM PAGE DELAY Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0x13	0x80	0	0/1	SEQ_23_ID

58.4.3.26 PROGRAM PAGE 1 Command

(1) Command Description

The PROGRAM PAGE 1 command allows the application program to input data to a cache register and moves the data from the cache register to the specified block and page address in the array of the selected LUN. This command is accepted by the LUN when it is ready. To input a page to the cache register and move it to the NAND array at the block and page address specified, the controller writes 0x80 to the target internal command register. Then five address cycles containing the column address and row address are written. Data input cycles follow. Serial data is input, beginning at the column address specified. When data input is complete, the commands sequence ends.

(2) Command Encoding

The PROGRAM PAGE 1 instruction uses the SEQ_13 commands sequence. The command is encoded, as shown in the table below:

Table 58.73 PROGRAM PAGE 1 Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x80	0	0/1	SEQ_13_ID

58.4.3.27 PROGRAM PAGE CACHE Command

(1) Command Description

The PROGRAM PAGE CACHE command allows the controller to input data to a cache register, copies the data from the cache register to the data register, and then moves the data register contents to the specified block and page address in the array of the selected LUN. After the data is copied to the data register, the cache register is available for additional PROGRAM PAGE CACHE or PROGRAM PAGE commands. The PROGRAM PAGE CACHE command is accepted by the LUN when it is ready. To input a page to the cache register to move it to the NAND array at the block and page address specified, the controller writes 0x80 to the target internal command register. Then five address cycles containing the column address and row address are written. Data input cycles follow. Serial data is input, beginning at the column address specified. When data input is complete, 0x15 is written to the command register. The selected LUN goes into busy state to allow the data register to become available from a previous program cache operation, to copy data from the cache register to the data register, and then to begin moving the data register contents to the specified page and block address.

(2) Command Encoding

The PROGRAM PAGE CACHE instruction uses the SEQ_12 commands sequence. Command is encoded as shown in the table below.

Table 58.74 PROGRAM PAGE CACHE Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0x15	0x80	0	0/1	SEQ_12_ID

58.4.3.28 PROGRAM MULTIPLANE Command

(1) Command Description

The PROGRAM MULTIPLANE command allows the controller to input data to the addressed plane's cache register and queue the cache register to ultimately be moved to the NAND array. This command can be issued one or more times. Each time a new plane address is specified that plane is also queued for data transfer. This command is accepted by the LUN when it is ready. The controller writes 0x80 to the target internal command register. Then five address cycles containing the column address and row address are written. Data input cycles follow. Serial data is input beginning at the column address specified. When data input is complete, the controller writes 0x11 to the target internal command register.

(2) Command Encoding

The PROGRAM MULTIPLANE instruction uses the SEQ_12 commands sequence. The command is encoded, as shown in the table below:

Table 58.75 PROGRAM MULTIPLANE Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0x11	0x80	0	0/1	SEQ_12_ID

58.4.3.29 WRITE PAGE Command

(1) Command Description

The WRITE PAGE command allows the controller to move data from the targets cache register to the NAND array. This command is accepted by the LUN when it is ready. The controller writes 0x10 to the target internal command register.

(2) Command Encoding

The WRITE PAGE instruction uses the SEQ_0 commands sequence. The command is encoded, as shown in the table below:

Table 58.76 WRITE PAGE Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x10	-	-	SEQ_0_ID

58.4.3.30 WRITE PAGE CACHE Command

(1) Command Description

The WRITE PAGE CACHE command allows the controller to moved data from the targets cache register to the targets data register. This command is accepted by the LUN when it is ready. The controller writes 0x15 to the target internal command register.

(2) Command Encoding

The WRITE PAGE CACHE instruction uses the SEQ_0 commands sequence. The command is encoded, as shown in the table below:

Table 58.77 WRITE PAGE CACHE Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x15	-	-	SEQ_0_ID

58.4.3.31 WRITE MULTIPLANE Command

(1) Command Description

The WRITE MULTIPLANE command allows the controller to queue data from the targets cache register to the NAND array. This command is accepted by the LUN when it is ready. The controller writes 0x11 to the target internal command register.

(2) Command Encoding

The WRITE MULTIPLANE instruction uses the SEQ_0 commands sequence. The command is encoded, as shown in the table below:

Table 58.78 WRITE MULTIPLANE Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x11	-	-	SEQ_0_ID

58.4.3.32 ERASE BLOCK Command

(1) Command Description

The ERASE BLOCK command erases the specified block in the NAND array. This command is accepted by the LUN when it is ready. To erase a block, the controller writes 0x60 to the target internal command register. Then three address cycles containing the row address are written; the column address is ignored. Finally, 0xD0 is written to the target internal command register.

(2) Command Encoding

The ERASE BLOCK instruction uses the SEQ_14 commands sequence. The command is encoded, as shown in the table below:

Table 58.79 ERASE BLOCK Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0xD0	0x60	-	-	SEQ_14_ID

58.4.3.33 ERASE MULTIPLANE Command

(1) Command Description

The ERASE MULTIPLANE command queues a block in the specified plane to be erased from the NAND array. This command can be issued one or more times. Each time a new plane address is specified, that plane is also queued for a block to be erased. This command is accepted by the LUN when it is ready. To queue a block to be erased, the controller writes0x60 to the command register. Then three address cycles containing the row address are written; the column address is ignored. Finally, 0xD1 is written to the command register.

(2) Command Encoding

The ERASE MULTIPLANE instruction uses the SEQ_14 commands sequence. The command is encoded, as shown in the table below:

Table 58.80 ERASE MULTIPLANE Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0xD1	0x60	-	-	SEQ_14_ID

58.4.3.34 COPYBACK READ Command

(1) Command Description

The COPYBACK READ command is functionally identical to the READ PAGE command, except that 0x35 is written to the target internal command register instead of 0x30. For more details, see Section 58.4.3.17, READ PAGE Command.

(2) Command Encoding

The COPYBACK READ instruction uses the SEQ_10 commands sequence. The command is encoded, as shown in the table below:

Table 58.81 COPYBACK READ Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
0x35	-	0x00	0/1	0/1	SEQ_10_ID

58.4.3.35 COPYBACK PROGRAM Command

(1) Command Description

The Copyback function reads a page of data from one location and then moves that data to a second location. The COPYBACK PROGRAM command is functionally identical to the PROGRAM PAGE command, except that when 85h is written to the target internal command register, the cache register contents are not cleared. The SEQ_9 command sequence does not have the data phase, so the data from the cache register are written into the second location without modification. If data must be written with modification to the second location, the SEQ_12 command sequence, which includes the data phase, is used.

(2) Command Encoding

The COPYBACK PROGRAM instruction uses the SEQ_9 commands sequence. The command is encoded, as shown in the table below:

Table 58.82 COPYBACK PROGRAM Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0x10	0x85	-	-	SEQ_9_ID

58.4.3.36 COPYBACK PROGRAM 1 Command

(1) Command Description

The COPYBACK PROGRAM 1 command is functionally identical to the PROGRAM PAGE 1 command, except that when 85h is written to the target internal command register, the cache register contents are not cleared. See Section 58.4.3.26, PROGRAM PAGE 1 Command for further details.

(2) Command Encoding

The COPYBACK PROGRAM 1 instruction uses the SEQ_13 commands sequence. The command is encoded as shown in the table below:

Table 58.83 COPYBACK PROGRAM 1 Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	-	0x85	0	-	SEQ_13_ID

58.4.3.37 COPYBACK MULTIPLANE Command

(1) Command Description

The COPYBACK MULTIPLANE command is functionally identical to the PROGRAM MULTIPLANE command, except that when 0x85 is written to the target internal command register, cache register contents are not cleared. See Section 58.4.3.28, PROGRAM MULTIPLANE Command for further details.

(2) Command Encoding

The COPYBACK MULTIPLANE instruction uses the SEQ_12 commands sequence. The command is encoded, as shown in the table below:

Table 58.84 COPYBACK MULTIPLANE Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0x11	0x85	0	0/1	SEQ_12_ID

58.4.3.38 PROGRAM OTP Command

(1) Command Description

The PROGRAM OTP command is used to write data to the pages within the OTP area. An entire page can be programmed at one time, or a page can be partially programmed up to four times. There is no ERASE operation for the OTP pages. To use the PROGRAM OTP command, the controller issues the 0xA0 command. Next, 5 address cycles are issued. The address write is followed by a programmable number of data cycles. After data input is complete, the controller issues the 0x10 command. The internal control logic automatically executes the proper programming algorithm and controls the necessary timing for programming and verification.

(2) Command Encoding

The PROGRAM OTP instruction uses the SEQ_12 commands sequence. The command is encoded as shown in the table below.

Table 58.85 PROGRAM OTP Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0x10	0xA0	0	0/1	SEQ_12_ID

58.4.3.39 DATA PROTECT OTP Command

(1) Command Description

The DATA PROTECT OTP command is used to protect all the data in the OTP area. After the data is protected. it cannot be further programmed. When the OTP area is protected, the pages within the area are no longer programmable and cannot be unprotected. To use the DATA PROTECT OTP command, the controller issues the 0xA5 command. Next, the controller issues the following 5 addresses cycles. Finally, the 0x10 command is issued.

(2) Command Encoding

The DATA PROTECT OTP instruction uses the SEQ_9 commands sequence. The command is encoded, as shown in the table below:

Table 58.86 DATA PROTECT OTP Instruction Encoding

CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
-	0x10	0xA5	-	-	SEQ_9_ID

58.4.3.40 PAGE READ OTP Command

(1) Command Description

The PAGE READ OTP command is used to read data from a page within the OTP area. An OTP page within the OTP area is available for reading data whether or not the area is protected. To use the PAGE READ OTP command, the controller issues the 0xAF command. Next, 5 address cycles are issued. Finally, the 0x30 command is issued. After internal read from the NAND matrix is ended, the data is copied to the FIFO.

(2) Command Encoding

The PAGE READ OTP instruction uses the SEQ_9 commands sequence. The command is encoded, as shown in the table below:

Table 58.87 PAGE READ OTP Instruction Encoding

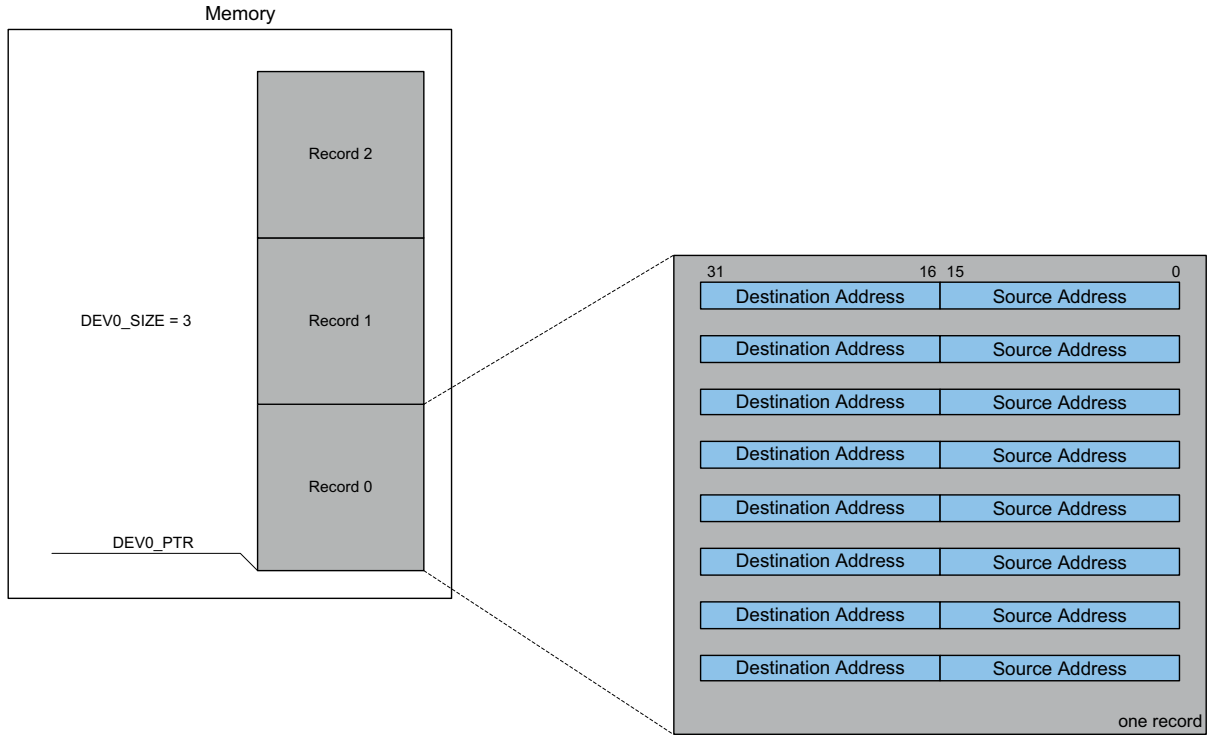
CMD_2	CMD_1 / CMD_3	CMD_0	DATA_SEL	INPUT_SEL	CMD_SEQ
0x30	-	0xAF	0/1	0/1	SEQ_10_ID

58.4.4 Remapping Mechanism

The remapping mechanism supports the bad blocks management (BBM) solution. The hardware remapping mechanism relieves software from the time-consuming operations of finding the physical address for the given linear address in the requested operation. The software initializes only the remapping tables for uses in the application of the NAND Flash device. It sorts those tables in ascending order, and then the whole operation of searching tables and substituting addresses is accomplished automatically.

The remapping solution uses two groups of the control registers:

- The pointer register DEV0_PTR. This register stores the address in the internal memory where record tables used by the BBM mechanism are placed.
- The size register DEV0_SIZE. This register stores the number of records in a table.



Note 1. Block Number are written to Destination Address and Source Address.
The table have to set by ascending order of Source Address.
When some Address of one record is unused, these Address should filled by 0xFFFF_FFFF.

Figure 58.29 Example of BBM records

58.4.5 Interrupts Mechanism

The NAND Flash controller interrupt system uses two control registers:

The interrupt mask register – each bit of this register masks a single interrupt. The register is described in more detail in Section 58.3.6, Interrupts mask register (INT_MASK).

The interrupt flag register – each bit of this register is an active flag from the single interrupt source. The register is described in more detail in Section 58.3.7, Interrupts status register (INT_STATUS).

In case an interrupt event occurs, i.e. the respective interrupt flag in INT_STATUS is 1, and its respective mask bit in INT_MASK is 1, the interrupt signal INTNFMA is asserted.
Note that the interrupt mask bits in INT_MASK affects only the interrupt signal INTNFMA, but not the flags in INT_STATUS.
A set flag in INT_STATUS must be cleared by the application program.

The available interrupt sources and their respective INT_STATUS flags are:

INT_STATUS.CMD_END_INT_FL

Command sequence finished interrupt – This interrupt occurs when the previously triggered command sequence is finished and the new one can be started. The command sequence is marked as finished when the full sequence is executed or when the NAND Flash device goes into the busy state.

INT_STATUS.ECC_INT0_FLAG

The ECC module detects an uncorrectable error in the transmitted data.
The ECC module detects when the configured errors threshold level is exceeded.

INT_STATUS.MEM0_RDY_INT_FL

The memory device is ready – This interrupt occurs when the NAND Flash device finishes executing the programmed command sequence and is ready for the new one. Each NAND Flash device has a single interrupt flag. The difference between “command sequence finished” interrupt and “memory device is ready” interrupt is presented in **Figure 58.30**.

INT_STATUS.TRANS_ERR_FL

The error on the slave interface during access to the controller FIFO – This interrupt occurs if the access to the FIFO memory has the opposite direction to the current FIFO configuration: the FIFO is read when it is configured for write, or the FIFO is write when it is configured for read.

INT_STATUS.STAT_ERR_INT0_FL

Most recently finished operation on the memory device failed. This applies to PROGRAM PAGE and BLOCK ERASE operations. It is not valid following a READ-series operation.

INT_STATUS.PG_SZ_ERR_INT_FL

Data Size error flag.
When the ECC is enabled, the value written into the DATA_SIZE register has some restrictions.
Interrupt condition is met when the value written to the DATA_SIZE register is not correct.

INT_STATUS.DMA_INT_FL

DMA transfer ended flag.

CAUTION

The DMA transfer end interrupt is asserted before the bus transaction completion.

To avoid data coherency issues following workaround needs to be implemented:

Always load additional unused 64 byte of data. This delays the interrupt and thus ensures that it is asserted when all intended data was completely transferred to the SDRAM.

INT_STATUS.DATA_REG_INT_FL

Data in DATA_REG is available.

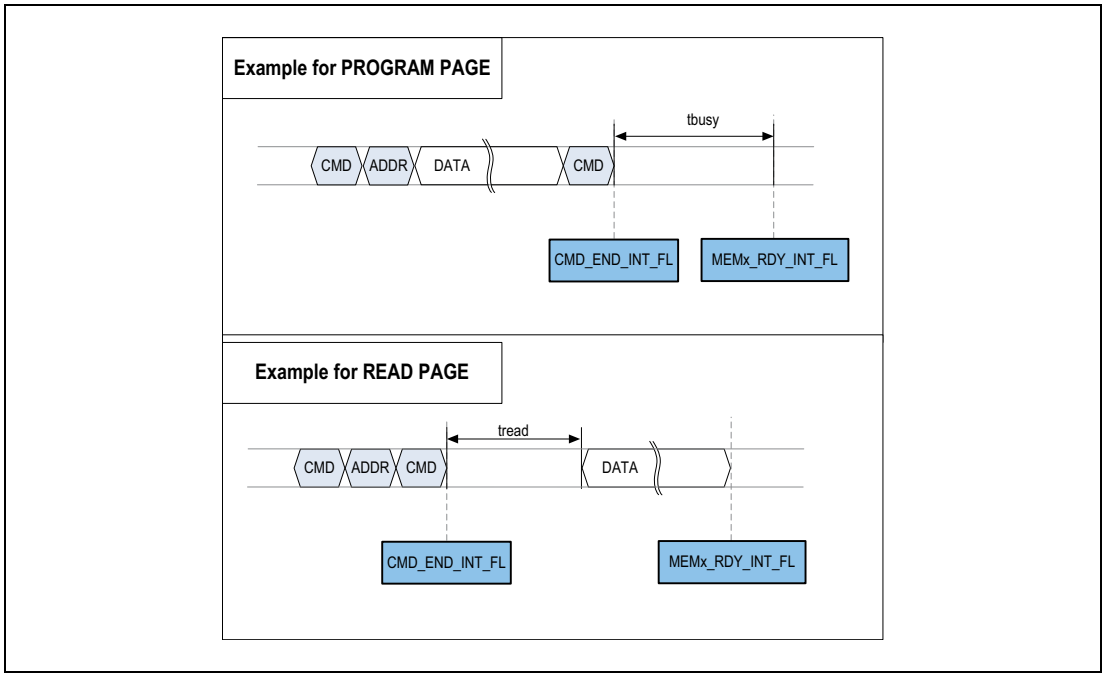


Figure 58.30 Command Sequence End and Memory-Ready Interrupts

58.4.6 Setup and Configuration

The CONTROL register is the main control register in the NAND Flash controller.

The following bits configure basic settings of the controller:

- INT_EN – Bit which enables Global Interrupt.
- ECC_EN – Bit which enables Hardware ECC.
- BLOCK_SIZE[1:0] – Bits which configure block size.
- IO_WIDTH – Bit which configures width of the I/O bus connected to the NAND Flash memory device.
- BBM_EN – Bit which enables remapping process.
- ADDRx_AUTO_INCR - Addresses auto increment for row address register 0 or 1.
- SMALL_BLOCK_EN – Bit which enables Small Block Mode.
- AUTO_READ_STAT_EN – Bit which activates automatic read status after the PROGRAM PAGE and BLOCK ERASE commands.
- READ_STATUS_EN – Bit which chooses whether the controller checks RnB lines or sends READ_STATUS commands.
- ECC_BLOCK_SIZE[1:0] – Bits which define ECC Block Size.

The registers described below are configured according to the settings of other bits in the CONTROL register:

- If INT_EN bit is set, the software must write the mask into the INT_MASK register, which masks the selected interrupts source in the NAND Flash controller.
- If ECC_EN bit is set, the software must correctly configure the ECC module by writing the appropriate configuration into the ECC_CTRL. Additionally, the software configures the offset in the ECC_OFFSET register. In small block mode, the value in the ECC_OFFSET is ignored and the correction words are located in the NAND Flash memory device, right behind the data.
- The write number of the data which will be transferred by the controller (DATA_SIZE register). When ECC is enabled, there are some restrictions to the DATA_SIZE value.
- If the BBM_EN bit is set, the software must initialize the remapping tables (DEVn_PTR and DEVn_SIZE registers).
- Additionally, the software must configure time parameters which can be found in the TIMINGS_ASYN register. Additionally, the software must configure the TIME_SEQ_0 and TIME_SEQ_1 registers.

When the NAND Flash controller uses DMA to transfer data, the software must configure the DMA_ADDR, DMA_CTRL and DMA_CNT registers. The software can modify these registers before any transfer or during the initialization procedure.

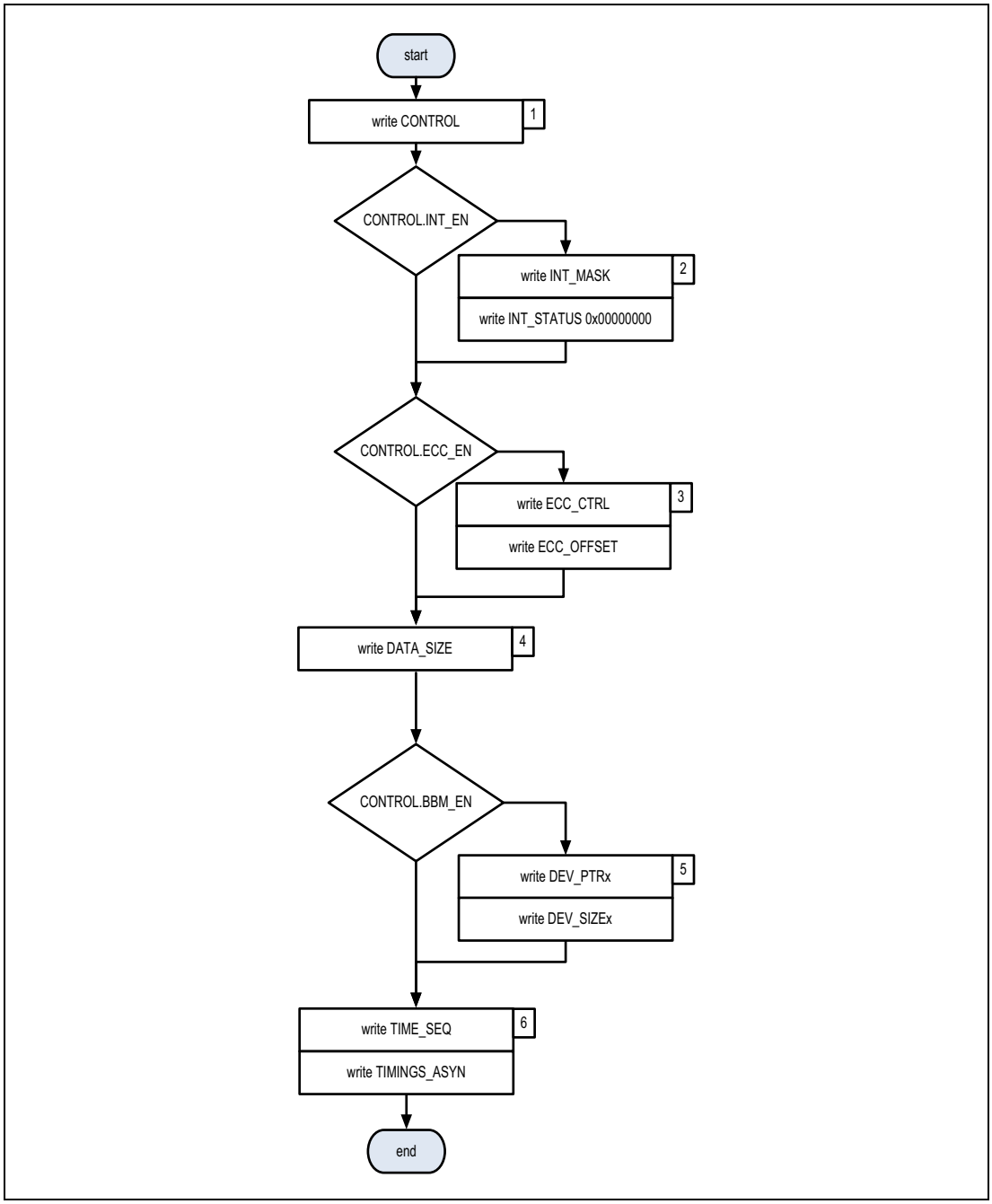


Figure 58.31 Configuration

58.4.6.1 Send Data to NAND Flash via Slave Interface

1. The NAND Flash controller must be correctly configured before sending data to the NAND Flash memory device. The setup process is described in detail in Section 58.4.6, Setup and Configuration and Section 58.3, Register Descriptions.
2. Write the address of the data in NAND Flash memory into the address register 0 (ADDR_COL and ADDR_ROW registers). Write the number of data which you want to read (DATA_SIZE register).
3. To use the simplest program command, write 0x0010800C to the COMMAND register (PROGRAM PAGE command, FIFO module selected, registers as input).
4. Write data to the FIFO using the FIFO_DATA register. Data is sent to the NAND Flash memory device.

When the memory is ready for further work, the appropriate bit MEMx_ST is set. When the software needs to send the command to another memory, it is not necessary to wait for the MEMx_ST bit to be set.

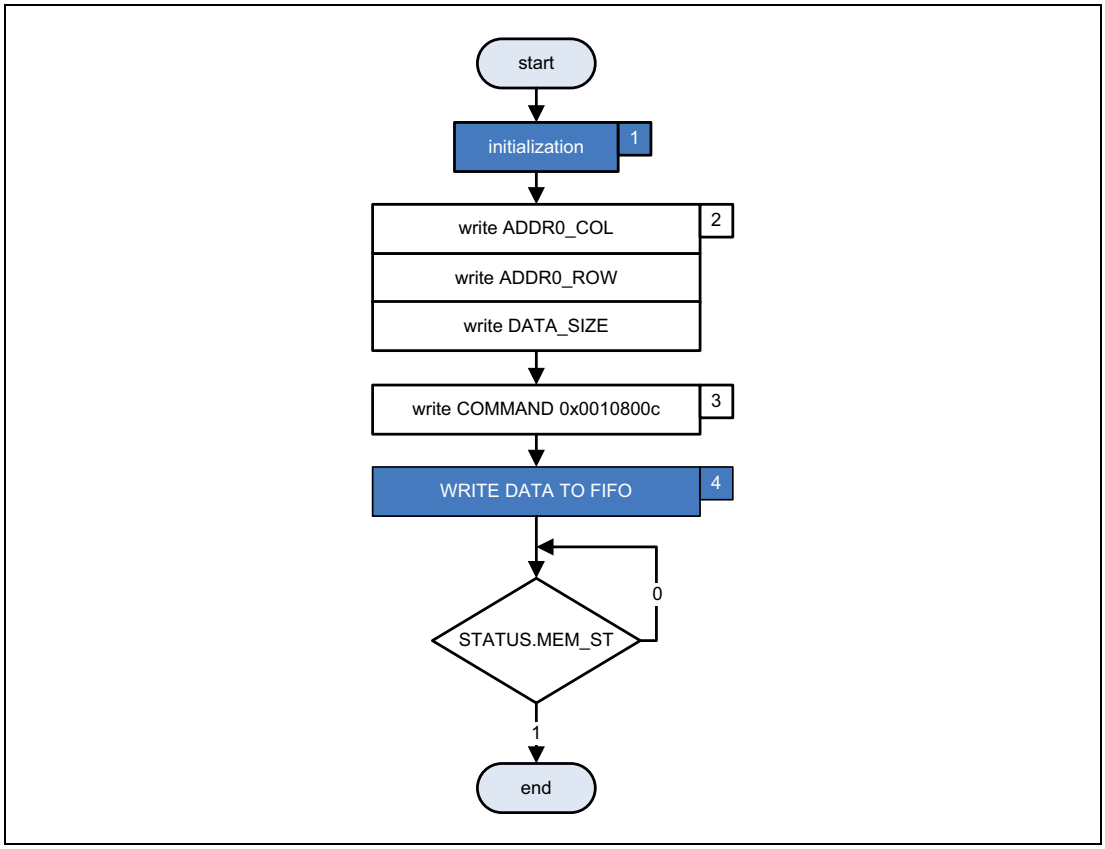


Figure 58.32 Write data to NAND Flash memory via Slave Interface

58.4.6.2 Read Data from NAND Flash via Slave Interface

1. The NAND Flash controller must be correctly configured before sending data to the NAND Flash memory device. The setup process is described in detail in Section 58.4.6, Setup and Configuration and Section 58.3, Register Descriptions.
2. Write the address of the data in NAND Flash memory into the address register 0 (ADDR_COL and ADDR_ROW registers). Write the number of data which you want to read (DATA_SIZE register).
3. To use the simplest read command, write 0x3000002A to the COMMAND register (READ PAGE command, FIFO module selected, registers as input).
4. The application program can read data immediately after sending the command but in this case the NAND flash controller will send a very few WAIT replies.
   If this situation shall be avoided the application program must read FIFO_STATE register and wait when CF_EMPTY bit is set. After that wait for the DF_R_EMPTY bit in the FIFO_STATE register to be clear.
5. Read data from the FIFO using the FIFO_DATA register.

When the memory is ready for further work, the appropriate bit MEMx_ST is set. When the software needs to send the command to another memory, it is not necessary to wait for the MEMx_ST bit to be set.

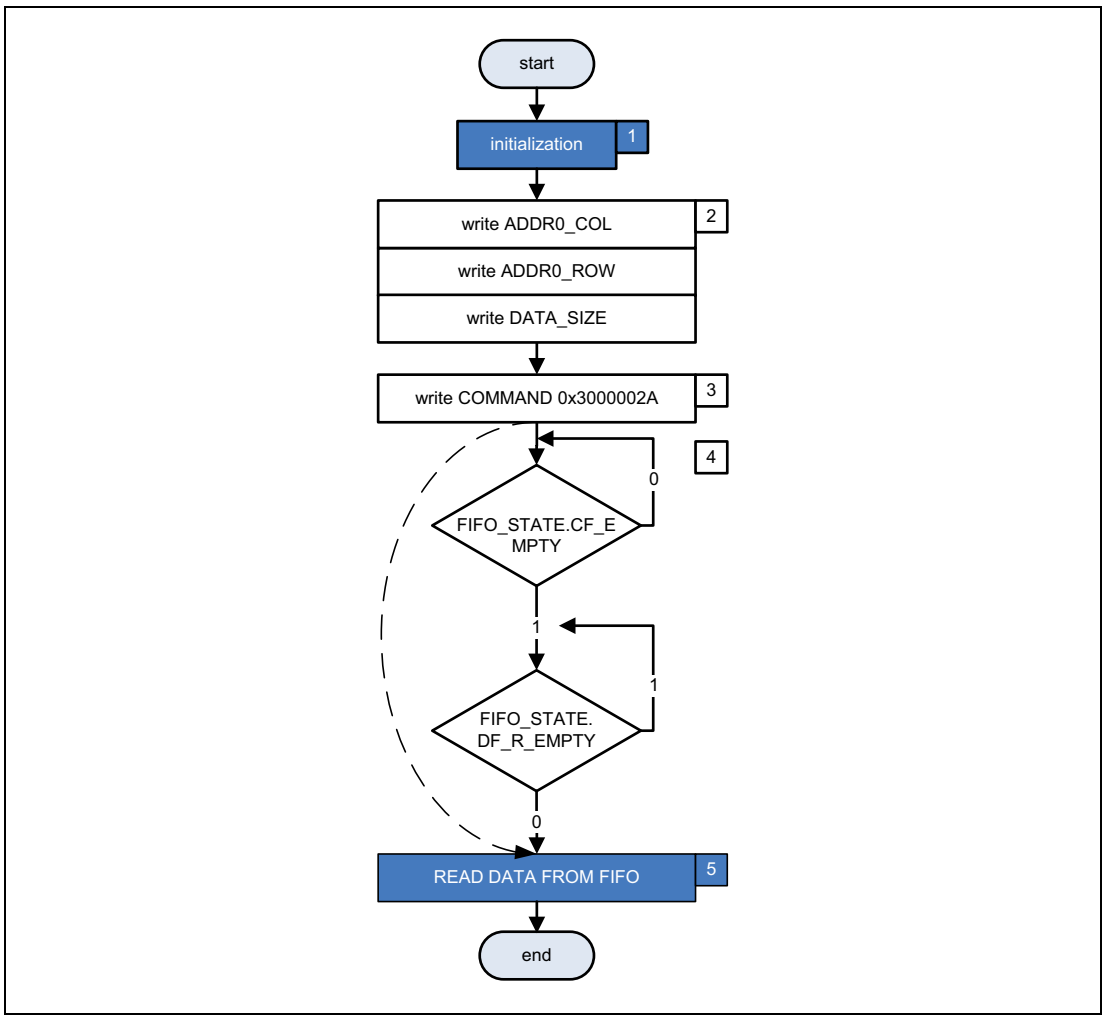


Figure 58.33 Read data from NAND Flash memory via Slave Interface

58.4.6.3 Send Data to NAND Flash via Master Interface (using DMA)

1. The NAND Flash controller must be correctly configured before sending data to the NAND Flash memory device. The setup process is described in detail in Section 58.4.6, Setup and Configuration and Section 58.3, Register Descriptions.
2. Set the INT_EN bit in the CONTROL register to enable global interrupt.
3. Select active interrupt in the INT_MASK register and write 0x00000000 into the INT_STATUS register to clear all interrupts.
4. Select the DMA work mode to configure the DMA module correctly:
   In registers managed mode, the address of the data is written in the system memory (DMA_ADDR_L register). Write the number of the transferred data into the DMA_CNT register. Set bit DMA_START to start DMA when the command sequence is sent to the NAND Flash memory. Bits ERR_FLAG and DMA_READY are read-only. The first indicates the error on the internal bus while DMA is transferring data; the second indicates when DMA is ready (transfer is completed).
5. Write the address of the data in the NAND Flash memory device into the address register 0.
6. To use the simplest program command, write 0x0010804C to the COMMAND register (PROGRAM PAGE command, FIFO module selected, DMA module as input).
7. When the memory is ready for further work, the appropriate bit MEMx_ST is set and interrupt is active. (recommended interrupt is MEM0_RDY_INT).

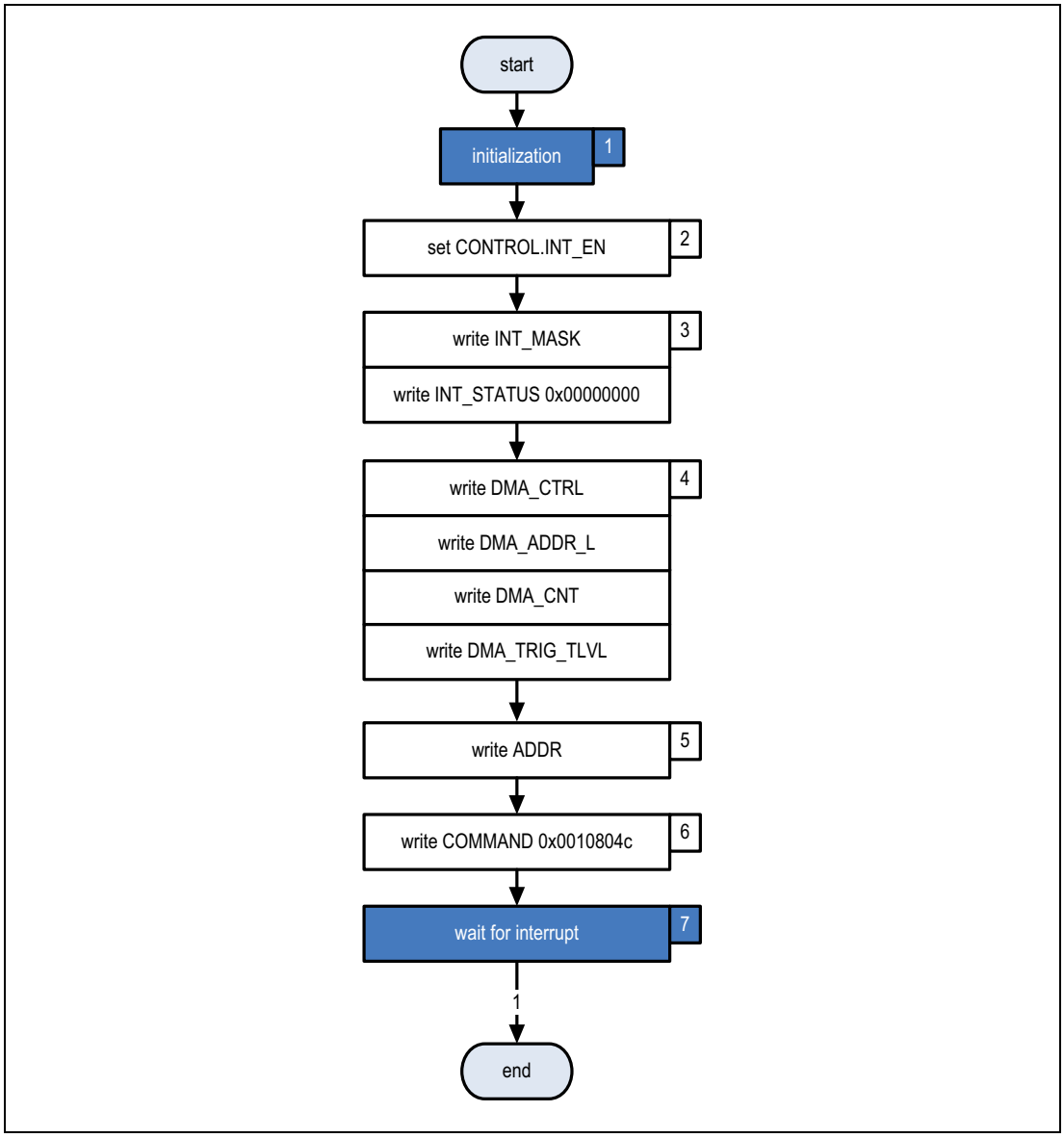


Figure 58.34 Send Data to NAND Flash Using DMA, Interrupt Enable

58.4.6.4 Fast Writing and Reading of Several Pages from the Memory using DMA

1. The NAND Flash controller must be correctly configured before sending data to the NAND Flash memory device. The setup process is described in detail in Section 58.4.6, Setup and Configuration and Section 58.3, Register Descriptions.
2. Set INT_EN bit in CONTROL register to enable global interrupt.
3. Select the active interrupt (CMD_END_INT_EN) in the INT_MASK register and write 0x00000000 into the INT_STATUS register to clear all interrupts.
4. In Scatter-Gather mode, it is necessary to write the descriptors into the system memory.
   Select the DMA work mode to correctly configure the DMA module.
   In Scatter-Gather mode, it is not necessary to configure the DMA_CNT register. For more details, see Section 58.5.2.2, DMA Description, **(b)**
5. Set DMA_START bit to start the DMA when the command sequence is sent to the NAND Flash memory.
   Bits ERR_FLAG and DMA_READY are read-only. The former indicates the error on the internal bus while DMA is transferring data, the latter indicates when the DMA is ready (transfer is completed).
6. Set the ADDR0_INCR bit to auto increment row address 0 register after each command.
   Write address of the data in the NAND Flash memory device into the 0 address register.
7. Write 0x00000000 into the INT_STATUS register to clear all interrupts.
8. Write address of the first descriptor into the DMA_ADDR_L register.
9. Write PROGRAM PAGE CACHE command to the NAND Flash memory device by writing 0x0015804C into the COMMAND register (DMA module as input, FIFO module selected).
   When the controller is ready for further work, the appropriate CMD_END_INT_FL bit is set and the interrupt is activated.
10. When the number of the pages to transfer does not equal one, go to section 7.
11. Write 0x00000000 into the INT_STATUS register to clear all interrupts.
    Write address of the first descriptor into the DMA_ADDR_L register.
    The last command in the sequence of sending data is the PROGRAM PAGE (write 0x0010804C) into the COMMAND register (DMA module as input, FIFO module selected).
12. Wait for interrupts and write 0x00000000 into the INT_STATUS register to clear all interrupts.
    Write the address of the read data to the NAND Flash memory device into address register 0.
13. Write the new descriptors to the system memory.
14. If DMA should work in the same work mode and burst type do not modify DMA_BURST[2:0] and DMA_MODE bits. Set DMA_START bit to start DMA when the command sequence is sent to the NAND Flash memory.
15. Write the READ PAGE command into the NAND Flash memory device by writing 0x00300069 into the COMMAND register (FIFO module selected, DMA module as input).
16. Write 0x00000000 into the INT_STATUS register to clear all interrupts.
17. Write the address of the first descriptor to the DMA_ADDR_L register.
18. Write READ PAGE CACHE SEQUENTIAL command to the NAND Flash memory device by writing 0x0000316B into the COMMAND register (FIFO module selected, DMA module as input).

19. When the controller is ready for further work, the appropriate CMD_END_INT_FL bit is set and the interrupt is activated.

When the number of the pages to transfer does not equal one, go to section 16.

20. Write 0x00000000 into the INT_STATUS register to clear all interrupts.Write the address of the first descriptor to the DMA_ADDR_L register. The last command in the sequence of reading data is READ PAGE CACHE LAST (write 0x00003F6B into the COMMAND register).

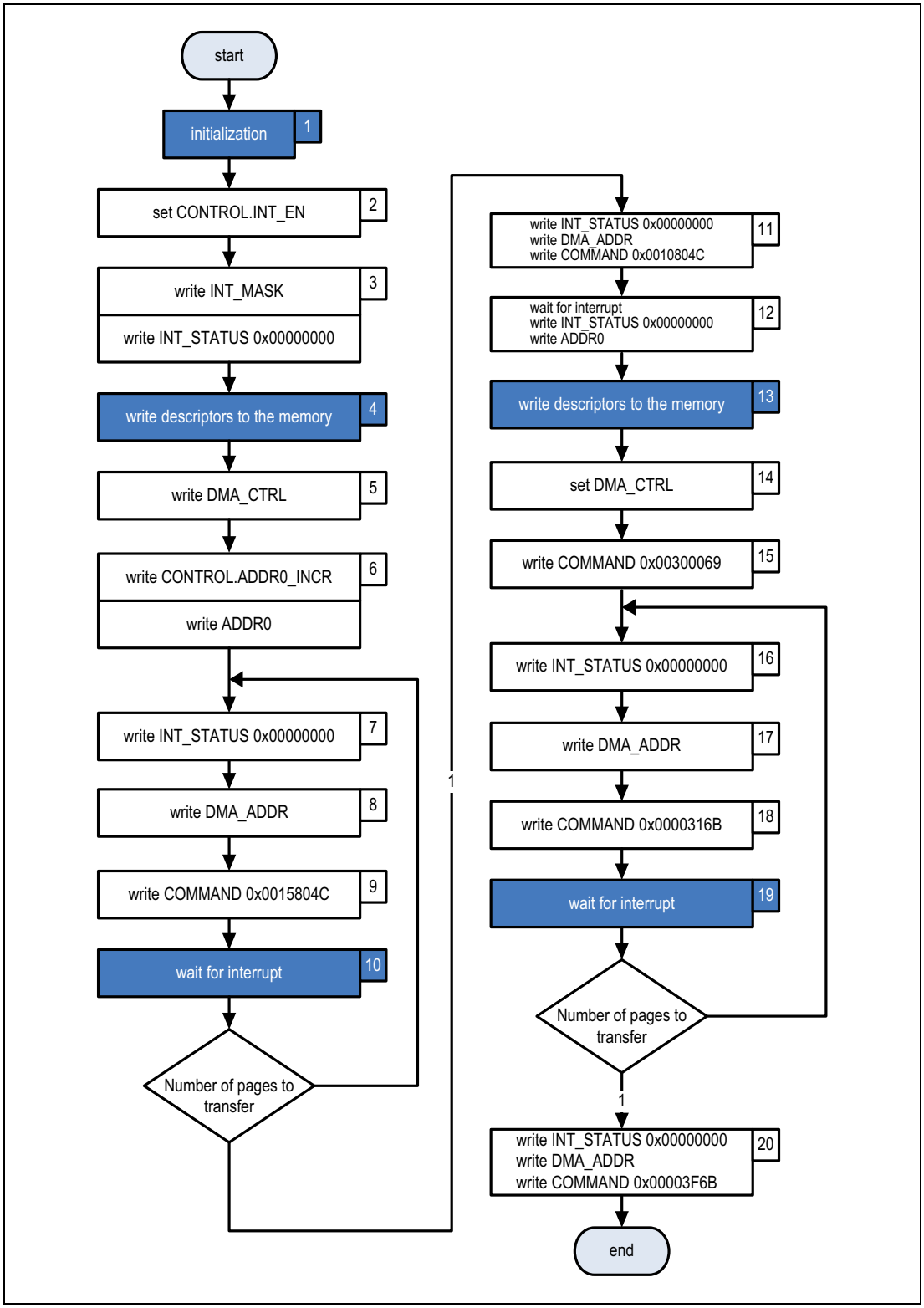


Figure 58.35 Fast Writing and Reading of Several Pages from the Memory Using DMA

58.4.6.5 Writing Partial Pages

The following procedure need to be used to write partial pages with ECC engine enabled:

1. The NAND Flash controller must be correctly configured before sending data to the NAND Flash memory device. The setup process is described in detail in Section 58.4.6, Setup and Configuration and Section 58.3, Register Descriptions.
2. Write the address of the data in NAND Flash memory into the address register 0 (ADDR_COL and ADDR_ROW registers). You need to write partial page offset into the ADDR_COL register. Write the number of data which you want to read (DATA_SIZE register) – in this case you need to set partial sector size. Write the offset value of the ecc data into the ECC_OFFSET register.
3. To use the simplest program command, write 0x0010800C to the COMMAND register (PROGRAM PAGE command, FIFO module selected, registers as input).
4. Write data to the FIFO using the FIFO_DATA register. Data is sent to the NAND Flash memory device.

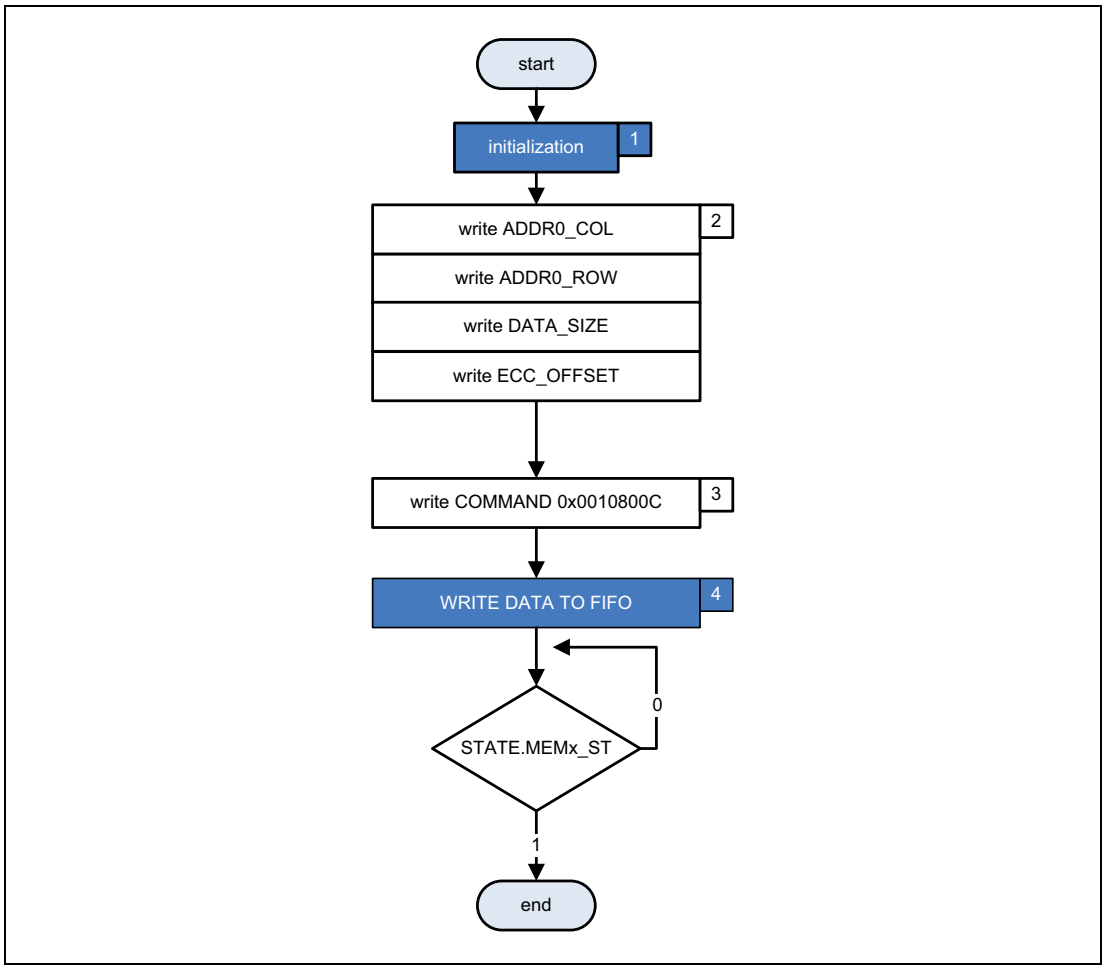


Figure 58.36 Writing Partial Pages

58.4.6.6 Reading Partial Pages

1. The NAND Flash controller must be correctly configured before sending data to the NAND Flash memory device. The setup process is described in detail in Section 58.4.6, Setup and Configuration and Section 58.3, Register Descriptions.

2. Write the address of the data in NAND Flash memory into the address register 0 (ADDR_COL and ADDR_ROW registers). You need to write partial page offset into the ADDR_COL register. Write the number of data which you want to read (DATA_SIZE register) – in this case you need to set partial page size.
3. To use the simplest read command, write 0x3000002A to the COMMAND register (READ PAGE command, FIFO module selected, registers as input).
4. You can read data immediately after sending the command but in this case the NAND flash controller will send a very few WAIT replies.
   If this situation shall be avoided the application program must read FIFO_STATE register and wait when CF_EMPTY bit is set. After that wait for the DF_R_EMPTY bit in the FIFO_STATE register to be clear.
5. Read data from the FIFO using the FIFO_DATA register.

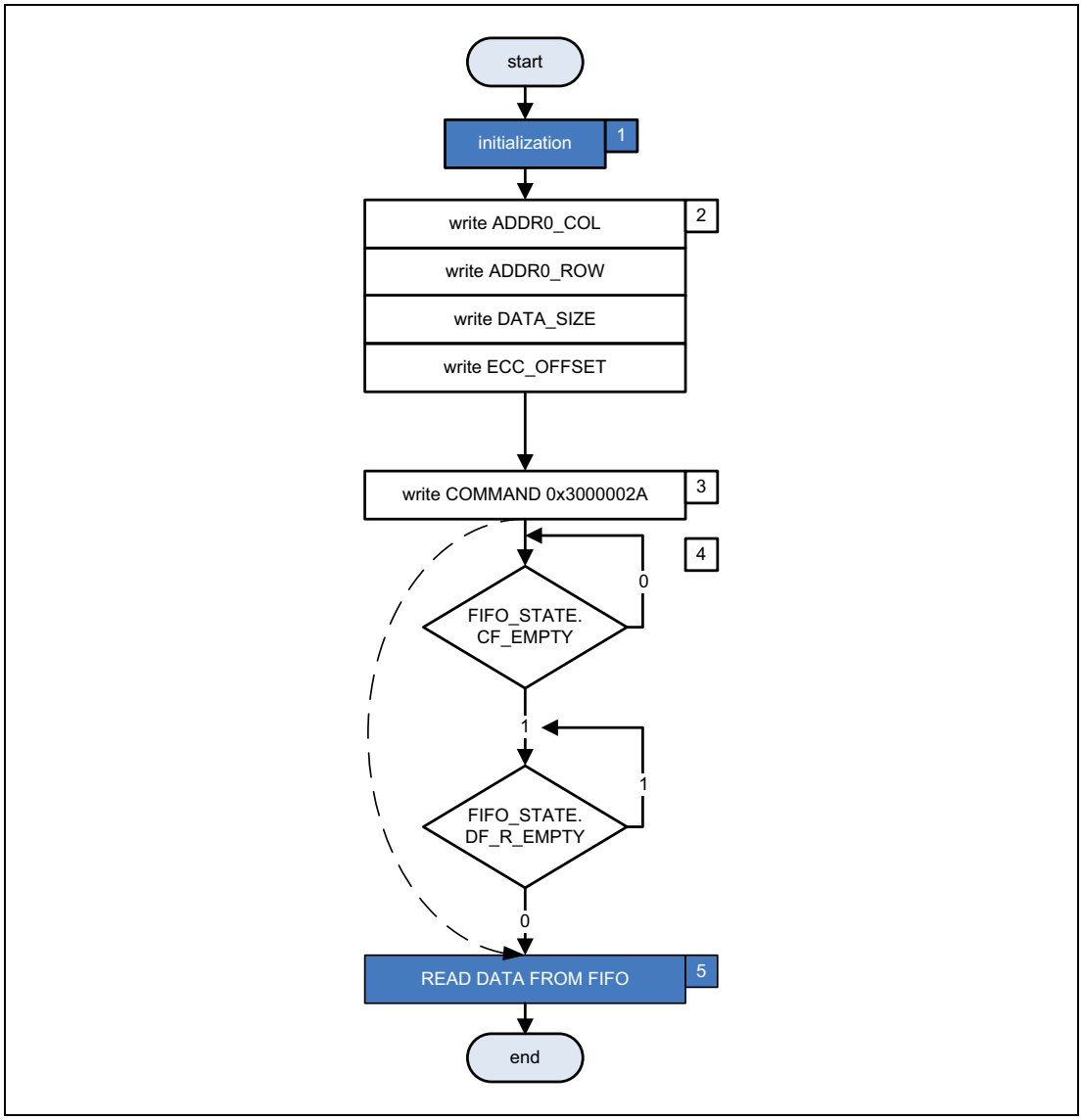


Figure 58.37 Reading Partial Pages

58.5 Functional Details

58.5.1 Block Diagram

The figure below shows the NAND Flash controller block diagram.

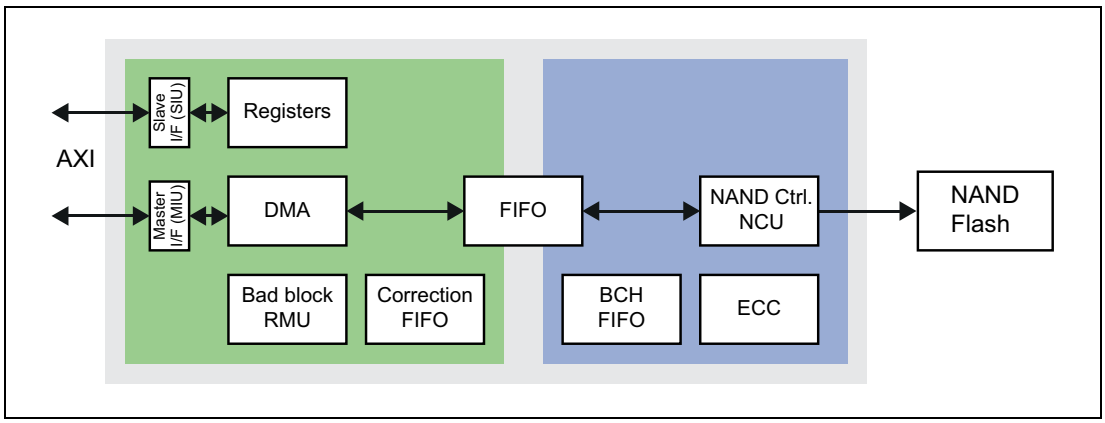


Figure 58.38 Block diagram

The main components of the controller are:

- **FIFO** – This unit provides the FIFO queue interface to the other controller modules. Using the FIFO interface allows closing of the read and write pointers in the same module; thus, two modules can read and write in parallel. Depending on the controller software configuration, one side of the queue is the SIU or DMA module, the second one is always the NAND Control Unit (NCU).
- **DMA** – This unit is responsible for the fast transfer of data between the external memory location and the controller. The unit can works in two modes: the Scatter-Gather mode and the internally triggered single mode. DMA functions as the master side of the OCP socket. This block is optional and can be disabled during the compile process.
- **NAND control unit (NCU)** – This unit is responsible for the generation of the NAND Flash device access sequences.
- **ECC** – An error correction code calculator. A correction word is calculated for each 256B, 512B or 1024B ECC block of the NAND Flash memory page. During the read operation, the unit can automatically correct bad bits without any interaction with the CPU. It has a status register, the bits of which signal errors occurring during a read, and then it inform when errors are corrected. The ECC module has an integrated FIFO that is used to transfer the calculated words to the NCU modules during the encode process and to store the calculated partial syndromes during the decode process. An additional FIFO module is used to transfer correction data between the correction module and the BCH-CHIEN module.
- **Bad block (RMU)** – It is bad block remapping unit. This module remaps blocks excluded from the poll of the usable blocks into the blocks that will be its substitute. Whole process is transparent for the software.

58.5.2 DMA Module

58.5.2.1 DMA Overview

The DMA engine integrated into the NAND Flash controller has the following properties:

- Two work modes:
  - The registers managed mode
  - The Scatter-Gather mode
- Supported burst types:
  - Imprecise bursts
  - Precise bursts
- Supported burst sequences:
  - The stream burst mode
  - The incrementing mode
- The DMA module preserves the configuration registers. These are copied into the DMA working registers.
- DMA trigger level enables DMA to send packet data.

58.5.2.2 DMA Description

DMA_ADDR_L gives a 32-bit address when the DMA is configured as 32-bit.

DMA supports two modes:

- The registers managed mode – the DMA does one transfer depending on the contents of DMA_ADDR, DMA_CNT, and DMA_CTRL registers.
- The Scatter-Gather mode – new DMA transfer algorithm.

The DMA mode for the current transfer is selected via the DMA_CTRL.DMA_MODE bit and cannot be changed during each consecutive data transfer. The application program can change the DMA mode when the transfer is completed and the DMA_CTRL.DMA_READY bit is set.

In order to begin the transfer, the DMA should be in an IDLE state (DMA_CTRL.DMA_READY = 1).

Setting the DMA_CTRL.DMA_START bit triggers the DMA after the commands are sent to the NAND Flash memory (writing CONTROL register). When the DMA_START bit is clear, sending commands to the NAND Flash memory will not trigger the DMA. Because of this solution, it is possible to program the DMA to transfer more data than the page size of the NAND Flash memory (example: the DMA is configured to 32 KB data transfer, the size of the page in NAND Flash memory is 4 KB. In order to send 32 KB of data into the NAND Flash memory, it is necessary to set the DMA_START bit in the register DMA_CTRL, then it is necessary to send the PROGRAM PAGE command to the NAND Flash memory, clear the DMA_START bit in the DMA_CTRL register, and send seven commands to the NAND Flash memory). In this example, the DMA is set free only once at the beginning.
The DMA transfers four 32-bit words in one package when working in a 4-beat incrementing burst, eight words in an 8-beat incrementing burst, and sixteen words in a 16-beat incrementing burst. The burst size indicates the number of 32-bit words in the burst, not the number of bytes transferred.
The DMA trigger level enables the DMA to send packet data. It indicates when the DMA has to start transferring data. The application program must set trigger level depending on arbitration type, system and NAND Flash memory performance. For example:

1. If the DMA trigger level equals 0, the DMA does not wait, but transfers data immediately because the trigger level is always exceeded.
2. For read from the NAND Flash memory: If the DMA trigger level equals 0x40 (256 bytes), the DMA waits until data level reach the previously programmed trigger level (256 bytes). Then the DMA will transfer data until FIFO gets empty status. Next the wait for required data level cycle is repeated as long as remain data size allow to reach programmed trigger level. The trigger level value should be chosen to ensure that only the continuous data block will be transferred and the FIFO will not be overflowed.
3. For write to NAND Flash memory: If the DMA trigger level equals 0x40 (256 bytes), the DMA writes data to the FIFO until the FIFO is full. Then, the DMA waits until 256 bytes are written in to the NAND Flash memory. Next, again the DMA transfers data until FIFO is full. The trigger level value should be chosen to ensure that only the continuous data block will be transferred.

(a) Registers Managed Mode:

When the DMA works in this mode, it transfers only one continuous block (DMA_CNT is block length) to or from the system memory at address DMA_ADDR. Block can be multiple of DATA_SIZE value in the DATA_SIZE register. All the registers which modify this transfer are described in **Table 58.29**, **Table 58.30** and **Table 58.31**.

The DMA_BURST[2:0] bit in register DMA_CTRL defines the main transfer type used by the DMA to precede the requested transfer.

(b) Scatter-Gather Mode:

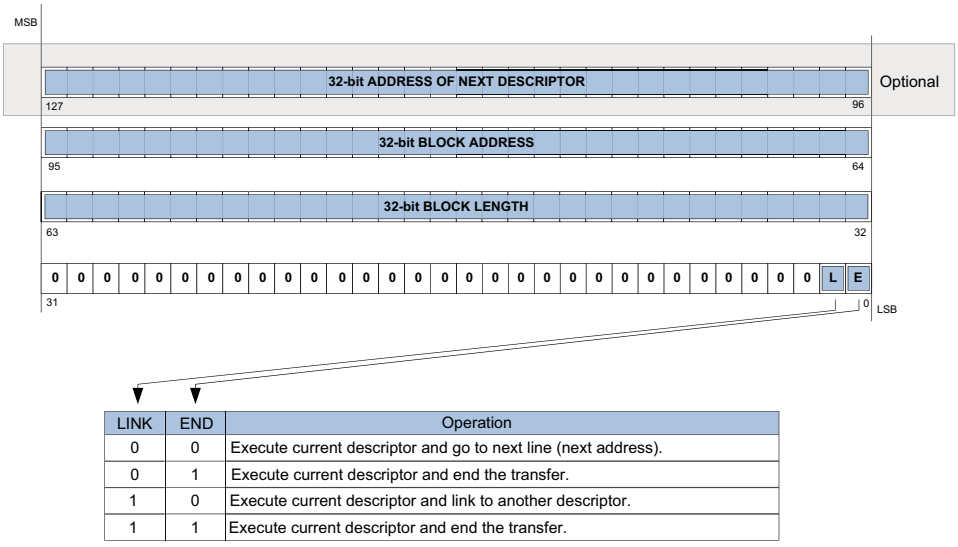
The Scatter-Gather Direct Memory Access (SG-DMA) controller implements high-speed data transfer between two components, when DMA transfers and merges non-contiguous memory to a continuous address space, and vice versa.

The DMA in Scatter-Gather mode uses the Descriptors List to describe data transfers. The NAND Flash registers only points to the base address of the Descriptors List. The base address for this list is set in the DMA_ADDR register whether it is a read or write transfer. The base address, sizes of the data blocks, and flags are defined inside the descriptors.

The DMA_BURST[2:0] bit in register DMA_CTRL defines the main transfer type used by the DMA to precede the requested transfer.

When in Scatter-Gather mode, the DMA transfers data from data blocks. Each descriptor can define only one data block.

The DMA adopts the scatter-gather DMA algorithm so that higher data transfer speed is available. The application program can program a list of data transfers between the system memory and NAND Flash to the Descriptor Table before executing Scatter-Gather mode.



LINK	END	Operation
0	0	Execute current descriptor and go to next line (next address).
0	1	Execute current descriptor and end the transfer.
1	0	Execute current descriptor and link to another descriptor.
1	1	Execute current descriptor and end the transfer.

Figure 58.39 Scatter-Gather DMA Descriptor Fields (32-bit system memory addressing)

The table below defines the SG-DMA descriptor fields and their functions:

Table 58.88 Scatter-Gather DMA Descriptor Fields (32-bit system memory addressing)

Bit	Symbol	Description
[127:96]	ADDRESS	This optional field contains the address of the next descriptor lit when LINK= 1 and END = 0
[95:64]	ADDRESS	The field contains the data block address.
[63:32]	LENGTH	The field contains the data page length in bytes ($0000\ 0004_H$ - $FFFF\ FFFD_H$). The number of the bytes has to be divided by 4 (two least significant bits should equal 00).
[31: 2]	–	Reserved for future use.
[1]	LINK	
[0]	END	

LINK	END	Comment
0	0	Execute current descriptor and go to next descriptor on the list.
0	1	Execute current descriptor and end the transfer.
1	0	Execute current descriptor and link to another descriptor.
1	1	Execute current descriptor and end the transfer.

Table 58.89 DMA Data Counter Length Field

Length Field	Value of Length
$0000\ 0000_H$	Not available
$0000\ 0001_H$	Not available
$0000\ 0002_H$	Not available
$0000\ 0003_H$	Not available
$0000\ 0004_H$	4 bytes
$0000\ 0005_H$	Not available
$0000\ 0006_H$	Not available
$0000\ 0007_H$	Not available
$0000\ 0008_H$	8 bytes
….	….

The Descriptor Table is created in system memory by the application program. SG-DMA can support only 32-bit system memory addressing. Each descriptor line (one executable unit) consists of the address, length and flags. The flags specify operation of the descriptor line.

The figure below shows the example of Scatter-Gather programming:

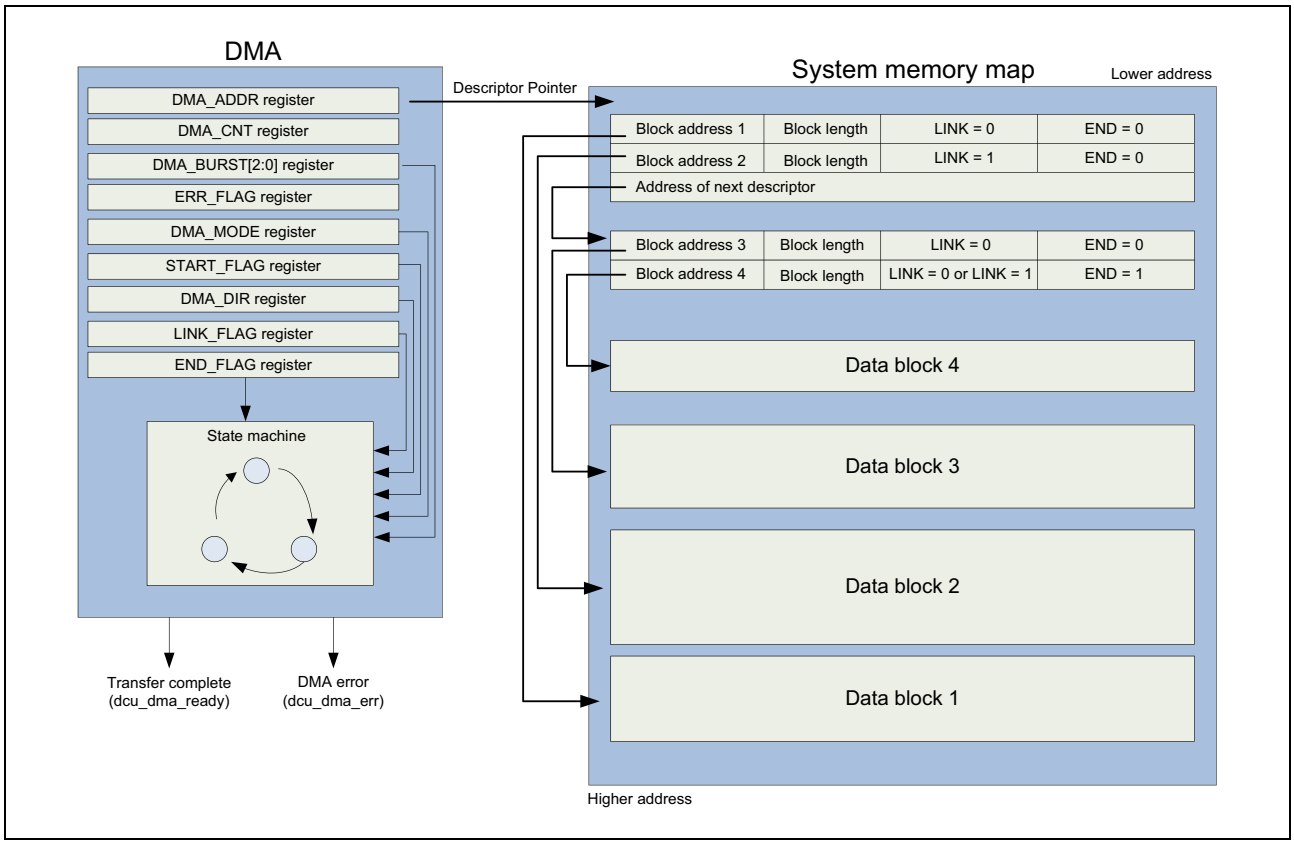


Figure 58.40 Example of Scatter-Gather DMA Data Transfer

58.5.3 ECC Module

58.5.3.1 Module Overview

The ECC module is based on one of the BCH algorithms and allows correction of multiple bit errors. The ECC engine integrated into the NAND Flash controller has the following properties:

- The encoder and decoder works on the 256, 512, 1024 bytes data blocks.
- Programmable correction capability: 2, 4, 8, 16, 24 or 32 errors.
- The corrected data words are aligned to the 32 bits.
- The correction words are aligned to the 32 bits.
- Correction words are placed after the data.

58.5.3.2 Block Diagram

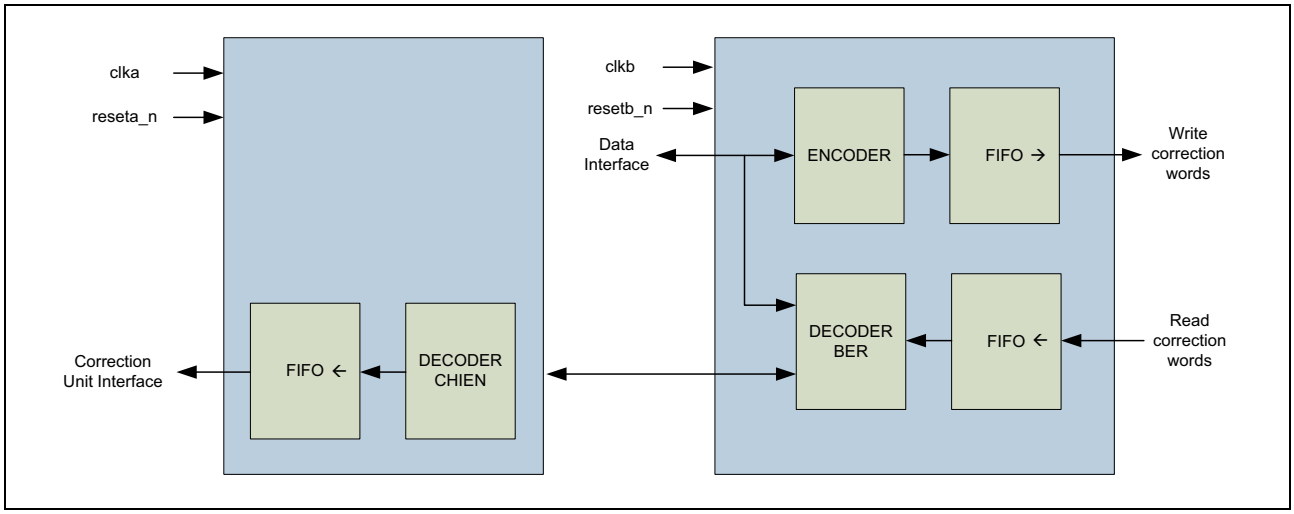


Figure 58.41 ECC Module Block Diagram

58.5.3.3 BCH Algorithm Implementation

- Data block length:
  - 256 (small block memory), 512 bytes or 1024 bytes.
- Programmable memory page length, multiple of the block length by any power of 2 (optional).
- Programmable correction capability: 2, 4, 8, 16, 24 or 32 errors.
- Separate encoder and decoder modules.
- Calculation of correction data performed during write to memory.
- Error detection performed during read data from memory.
- Internal pipeline allows the correction of errors in one data block simultaneously with detection of errors in the following data block.

Table 58.90 Size of Correction Bytes

Correction Capability	ECC Block Size	Size of Correction Bytes per one ECC Block
2 bit	256 / 512 / 1024 byte	4 byte
4 bit	256 / 512 / 1024 byte	7 byte
8 bit	256 / 512 / 1024 byte	14 byte
16 bit	256 / 512 / 1024 byte	28 byte
24 bit	256 / 512 / 1024 byte	42 byte
32 bit	256 / 512 / 1024 byte	56 byte

Revision History

Revision 0.60 History

This revision history list shows all modifications of

Rev. 0.60 compared to Rev. 0.50.

NOTE

The chapter and page numbers in the table below refer to the older Rev. 0.60 and thus may not be valid for the current revision.

Revision 0.60 History, Date Aug 29, 2014 (1/13)

Section	Page	Summary
1	89	Code Flash size of D1L2(H) devices changed to 4 MB
1	89	MainOsc frequency range changed
1	90	name of CAN interface with FD support changed to RSCAN-FD
1	91	RS-CANFD added to D1L1 block diagram
1	92	RS-CANFD added to D1L2(H) block diagram
1	93	MainOsc frequency range changed
1	94	name of CAN interface with FD support changed to RSCAN-FD
1	95	ISO power supply for D1M2 devices corrected
1	95	D1M1 package type changed to HLQFP
1	96	RS-CANFD added to D1M1(H) block diagram
1	97	RS-CANFD added to D1M2(H) block diagram
1	98	part numbers for devices with RS-CANFD I/F added
2	102	D1M1 package type changed to 176 HQFP
2	104	D1M2 pin connection diagram added
2	112	"Module Name" column added to register table
2	113	note added concerning register names in header files
2	114	typo correction (FCLA0CTLm -> FCLAnCTLm)
2	120	caution added concerning usage of alternative functions
2	121	caution added concerning usage of alternative functions
2	137	port initialization flow for alternative ports with PIPC=1 corrected
2	138	PODCEn.PODCEn_m removed from flowchart
2	142	input buffer characteristics tables separated for the different devices
2	142	input characteristic name "CMOS" changed to "CMOS1"
2	148	input buffer characteristic selection of JP0_3 and JP0_4 corrected
2	151	P21 port input buffer characteristics corrected
2	151	P40 port input buffer characteristics corrected
2	152	P44 port input buffer characteristics corrected
2	153	P45 port input buffer characteristics corrected
2	204	XOR Compare Unit removed from SG0FAOL at port P3_3
2	204	XOR Compare Unit removed from SG0AO at port P3_5
2	207	ISM channel connections of P16_[11:8] corrected
2	207	P16 ISM assignment changed (ISM/ZPD3 -> ISM/ZPD5)

Revision 0.60 History, Date Aug 29, 2014 (2/13)

Section	Page	Summary
2	208	ISM channel connections of P17_[11:0] corrected
2	208	P16 ISM assignment changed (ISM/ZPD4 -> ISM/ZPD6)
2	208	ZPD4x changed to ZPD6x
2	208	P16 ISM assignment changed (ISM/ZPD5 -> ISM/ZPD3)
2	208	ZPD5x changed to ZPD3x
2	208	P16 ISM assignment changed (ISM/ZPD6 -> ISM/ZPD4)
2	208	ZPD6x changed to ZPD4x
2	212	VDCE1_VO_TCON3 added as 2nd alternative output of P43_6
2	214	TAUB2O8 exchanged by VDCE0_VO_TCON3
2	218	ports for "Port output check by timers" corrected
2	228	correction: controls digital filter processing -> controls filter processing
2	233	X1/X2 pin states in normal operation mode corrected
2	233	XT1/XT2 pin states in normal operation mode corrected
2	233	signal name corrections (SDRAA[13:9] -> SDRAA[13:11])
2	233	signal name corrections (SDRAA10PC -> SDRAA10)
2	234	signal name corrections (SDRBA[13:9], SDRBA10PC -> SDRBA[13:11], SDRBA10)
2	234	Note1. added
2	234	X1/X2 pin states in normal operation mode corrected
2	234	XT1/XT2 pin states in normal operation mode corrected
2	235	signal name corrections (SDRAA[13:9] -> SDRAA[13:11])
2	235	signal name corrections (SDRAA10PC -> SDRAA10)
2	235	signal name corrections (SDRBA[13:9], SDRBA10PC -> SDRBA[13:11], SDRBA10)
2	235	Note1. added
2	235	Note2. added
2	236	signal name corrections (SDRAA[13:9] -> SDRAA[13:11])
2	236	signal name corrections (SDRAA10PC -> SDRAA10)
2	237	signal name corrections (SDRBA[13:9], SDRBA10PC -> SDRBA[13:11], SDRBA10)
2	238	correction: connect XT1 to OSCVSS when unused
2	238	recommended connection of unused RESETZ corrected
2	238	note added for all unused general purpose I/O ports
3	247	note added concerning register names in header files
3	260	upper window address of cache window example changed
3	262	buffer base and mask registers added
3	268	Index type (IDX_*, CLEAR) register description corrected
3	276	initial values of ACCABASEn corrected
3	277	initial values of ACCAMASKn corrected
3	278	Buffer base n register (AXCBFBASEn) added
3	279	Buffer mask n register (AXCBFMASKn) added
4	282	SWRESA register assigned to control protection cluster 0
4	283	register name IDMODI changed to SELBIDMODI
4	285	"Module Name" column added to register table
4	286	note added concerning register names in header files
4	301	register name IDMODI changed to SELBIDMODI
4	301	register name IDMODI changed to SELBIDMODI

Revision 0.60 History, Date Aug 29, 2014 (3/13)

Section	Page	Summary
4	302	register name IDMODI changed to SELBIDMODI
5	307	D1L2(H) Code Flash size changed to 4 MB
5	309	D1L2(H) Code Flash size changed to 4 MB
7	313	register access size column added
7	313	note added concerning register names in header files
7	363	typo correction (BERR0SSF28 -> BERR0SSF128)
7	364	BERR0CL017 removed
7	364	typos correction in table (BERR0ST1 -> BERR0ST0)
7	364	BERR0CL017 removed
7	366	section "DMA bus error status registers" added
8	381	DMA transfer request acceptance indicators completed
8	382	typo correction (requests -> request)
8	393	VM bit added to table
8	393	typo correction (UM -> VM)
8	397	table column name "Channel" changed to "DMA trigger ID"
8	401	"Module Name" column added to register table
8	401	note added concerning register names in header files
8	407	description of VM bit corrected
8	408	"Module Name" column added to register table
8	409	note added concerning register names in header files
9	428	signal name "PWRGD" added
9	429	Reset controller block diagram corrected
9	430	Low Speed IntOsc removed from table as reset target
9	430	Low Speed and High Speed reset targets separated
9	431	typo correction (POC1RES -> POC0RES)
9	431	external ISO reset control output corrected to PWRCTL
9	431	"Isolated-Area power reset control" in diagram indicated
9	431	correction (ISO power reset flag -> ISOPWRES power reset flag
9	432	typo correction (POC1RES -> POC0RES)
9	432	external ISO reset control output corrected to PWRCTL
9	432	"Isolated-Area poser reset control" in diagram indicated
9	434	description of external reset input/output improved
9	437	"Module Name" column added to register table
9	437	note added concerning register names in header files
9	443	reset sources of MRSTC stated more precisely
9	443	VDCE1RES and VDCE0RES exchanged in register image
10	451	correction: B2VCC/B2VSS only for D1M2(H)
10	452	note concerning relation between ZPDVCC and REG0VCC removed
10	455	note concerning relation between ZPDVCC and REG0VCC removed
11	460	"Module Name" column added to register table
11	460	Temperature Sensor status register (TSNSTAT) address corrected
11	460	note added concerning register names in header files
12	464	MainOsc frequency range changed
12	469	section "CPU Subsystem and bus clock domains" revised

Revision 0.60 History, Date Aug 29, 2014 (4/13)

Section	Page	Summary
12	472	CLMAOTCTL1 register initial value corrected
12	472	note added concerning register names in header files
12	474	PLL0E register added to caution
12	474	PLL1E register added to caution
12	475	MainOsc reset described more precisely
12	475	MainOsc frequency range changed
12	477	SubOsc reset described more precisely
12	477	"SubOsc STOP requests in DEEPSTOP mode" added
12	479	High Speed IntOsc reset corrected in diagram
12	479	CLMA0RES added as reset source in figure
12	479	High Speed IntOsc reset described more precisely
12	484	typo correction (CKCS_IPLL2INS_CTL -> CKSC_IPLL2INS_CTL)
12	484	typo correction (CKDV_IPLL2IND_CLT -> CKDV_IPLL2IND_CTL)
12	484	typo correction (CKCS_IPLL2INS_CTL -> CKSC_IPLL2INS_CTL)
12	487	"Module Name" column added to register table
12	487	SubOsc stop mask register added
12	487	typo correction (CKSC_IPPL0S_CTL -> CKSC_IPLL0S_CTL)
12	487	typo correction (CKSC_IPPL0S_ACT -> CKSC_IPLL0S_ACT)
12	487	typo correction (CKSC_IPPL1S_CTL -> CKSC_IPLL1S_CTL)
12	487	typo correction (CKSC_IPPL1S_ACT -> CKSC_IPLL1S_ACT)
12	488	BUS_CLK_RATIO register removed
12	488	C_AWO_WDTA0 clock divider active register (CKSC_AWDTA0D_STPM) removed
12	489	BUS_CLK_RATIO register removed
12	489	C_AWO_WDTA0 clock divider active register (CKSC_AWDTA0D_STPM) removed
12	490	note added concerning register names in header files
12	493	settings of MOSCAMPSEL[1:0] corrected
12	493	note added concerning oscillators for low power modes
12	498	SubOsc Stop Mask Register (SOSCSTPM) added
12	499	ROSCE reset source corrected
12	500	ROSCS reset source corrected
12	501	ROSCST reset source corrected
12	502	ROSCSTPM reset source corrected
12	511	typo correction (NI[6:0]PLL1 -> NI[6:0])
12	516	table columns names corrected
12	518	"(default)" added
12	531	typo correction (IPLLFIXS ACT[1:0] -> PLLFIXS ACT[1:0])
12	532	typo correction (ISDRBSSTP -> SDRBSSTP)
12	532	"(default)" added
12	533	typo correction (SDRBACT -> SDRBDACT)
12	538	initial value of CKSC_IXCCLKS_CTL corrected
12	538	caution added concerning activation of the cross-connect clock domain C_ISO_XCCLK
12	539	initial value of CKSC_IXCCLKS_ACT corrected
12	540	"(default)" added
12	542	"(default)" added

Revision 0.60 History, Date Aug 29, 2014 (5/13)

Section	Page	Summary
12	545	CKSC_IPCMLBBS_ACT initial value corrected
12	546	default value in IXCMLBBSSTP bit description corrected
12	551	typo correction (WOTSACT[2:0] -> AWOTSACT[2:0])
12	555	caution added concerning CKSC_AWDTA0D_CTL register setting
12	571	note for PLLFIXCLK/12 and /24 removed
12	571	typo correction (RSCANSCSID[2:0] -> RSCANDCSID[2:0])
12	571	correction: selection -> divider
12	574	typo correction (RSCANXINSACT0 -> IRSCANXINSACT0)
12	585	"(default)" added
12	587	type correction (CKLJIT -> CLKJIT)
12	587	caution added concerning C_ISO_ADCE frequency
12	592	section "Clock Controller set-up" added
12	594	"Clock Selection" section added
12	600	"Module Name" column added to register table
12	600	note added concerning register names in header files
12	600	typo correction (R -> R/W)
12	600	FOUTDIV settings corrected
12	601	correction read/written -> read
12	601	typo correction (R/W -> R)
12	603	CLMAn output signal connections corrected
12	605	note added concerning register names in header files
12	606	Clock Monitor block diagram corrected
12	609	figure title corrected
12	609	figure title corrected
12	609	figure title corrected
12	611	"Module Name" column added to register table
12	611	note added concerning register names in header files
13	623	typo correction (INTRTCA0S -> INTRTCA01S)
13	623	ECM signal name corrected (INTECM -> ECMTI)
13	623	ECM signal name corrected (NMIECM -> ECMTNMI)
13	628	"Module Name" column added to register table
13	628	note added concerning register names in header files
13	629	initial value of WUF0 register corrected
13	634	"Preparation for DEEPSTOP" flow improved
14	637	"Bus Systems Clock Supply" section added
14	666	"SPID and PEID assignment table" relocated
14	670	description of PEGGnMK.GnMASK corrected
14	680	"Module Name" column added to register tables
14	680	note added concerning register names in header files
14	680	PBG channel numbers for PBG5A corrected
14	683	"SPID and PEID assignment table" relocated
14	683	"SPID and PEID assignment table" relocated
14	686	VM moved from bit 5 to bit 7
14	694	typo correction (XCGnINT1L -> XCGnINTL1)

Revision 0.60 History, Date Aug 29, 2014 (6/13)

Section	Page	Summary
16	710	SDRBA10PC signal name corrected to SDRBAPC
16	712	typo correction (I/Memory -> Memory)
16	712	"Module Name" column added to register table
16	712	PHY control register 0 added
16	712	PHY control register 1 added
16	712	PHY control register 3 added
16	713	note added concerning register names in header files
16	718	typo correction (ARFEN = -> ARFEN = 1)
16	723	typo correction (DBCONF0.DB0[1:0] -> DBCONF0.DW0[1:0])
16	736	READ/WRITE interval calculation formula corrected
16	741	DQL calculation formula corrected
16	753	description of RODTOUT0 corrected
16	753	description of WODTOUT0 corrected
16	755	PHY control register 0 (DBPDCNT0) added
16	756	PHY control register 1 (DBPDCNT1) added
16	757	PHY control register 3 (DBPDCNT3) added
16	762	typo correction (3FFFFH -> 3FFFH)
16	765	step (3) added to initialization sequence
16	765	step (4) added to initialization sequence
17	787	"Module Name" column added to register table
17	787	note added concerning register names in header files
17	793	correction in register image: BFM[1:0] -> BFM[2:0]
17	793	description of DRCR.BFM[2:0] corrected
17	811	bit position of CKDLYOC[2:0] corrected
17	811	CKDLYOC[2:0]=010B setting added
17	813	typo correction in register image (DRDRD -> DRDRE)
17	829	Output pin (CPHAT = 1) signal corrected
17	830	Output pin (CPHAT = 1) signal corrected
19	896	typo correction (CSIHnCSx -> CSIHnCSLx)
19	901	removed sentence concerning CSIHnCTL0.CSIHnPWR = 0
19	914	typo correction (CSIHnDAPxOSC -> CSIHnDAPx)
19	916	changed "inter-data delay time" into "inter-data time"
19	921	typo correction (CSIHnRX0Hn -> CSIHnRX0H)
19	989	typo correction (INCSIHTIC -> INTCSIHTIJC)
19	989	typo correction (IHTIC -> INTCSIHTIJC)
19	989	typo correction (IHTIC -> INTCSIHTIJC)
20	991	caution added concerning RLIN3 clock frequencies
20	1027	typo correction (LBRP0[15:0] -> BRP[15:0])
20	1043	typo correction (R -> R/W)
20	1090	typo correction (N +1 -> M + 1)
21	1135	write value of unused bits in RIICnDRT corrected
22	1182	"Number of Units" table corrected
22	1182	"Unit Configurations and Channels" table corrected
22	1183	number range of index r corrected

Revision 0.60 History, Date Aug 29, 2014 (7/13)

Section	Page	Summary
22	1207	RAM test page access registers 30 to 63 removed
22	1240	typo correction (GAFLRVM -> GAFLRMV)
22	1243	NRXMB[7:0] number range corrected
22	1312	RTMPS[6:0] number range corrected
22	1312	RTMPS[6:0] number range corrected
22	1316	number range of index r corrected
22	1322	max. transition time Global test -> Global reset corrected
22	1322	max. transition time Global operation -> Global reset corrected
22	1325	max. transition time Channel reset -> Channel communication corrected
22	1325	max. transition time Channel halt -> Channel reset corrected
22	1325	max. transition time Channel halt -> Channel reset corrected
22	1329	number of Rx buffers corrected
22	1329	number of receive rules corrected
22	1329	maximum number of pages corrected
22	1338	formatting of bulleted list corrected
22	1344	number range of index r corrected
22	1344	total RAM size corrected
22	1345	RAM initialization time corrected
22	1348	maximum number of Rx rule pages corrected
22	1349	number of buffers corrected
22	1350	maximum number of Rx buffers corrected
22	1368	number range of index r corrected
23	1381	RS-CANFD description added
24	1715	"Module Name" column added to register table
24	1716	"Module Name" column added to register table
24	1716	note added concerning register names in header files
24	1777	"QFF0 to 17 Receive 0 to 17 Full Interrupt Status Bits" description corrected
24	1881	"Flow of Transmission Time Stamping" corrected
24	1919	"Rx-FIFO read error may not be flagged when using FEMPTY_ND descriptor" usage note added
24	1919	"When trying to release non-existing timestamp FIFO entry, new FIFO update flag may be lost: usage note added
24	1919	"gPTP compare may fail for range of compare values" usage note added
24	1919	"UFC stop level triggers RIS2.QFFr even no received frame is lost" usage note added
24	1920	"RIS0.FRFr may lost when data processing stops close below configured warning level" usage note added
25	1925	"Module Name" column added to register table
25	1926	note added concerning register names in header files
26	1929	caution concerning reset conflictions added
26	1931	typo corrections (WDT0. -> WDT0, WDTA1. -> WDTA1)
26	1940	added "Note 2" for reference to reset conflict cases
28	1971	wording and format improved
28	1973	formula for AWOTOUT cycle duration calculation added
28	1973	note added
28	1974	formula for capture period calculation added

Revision 0.60 History, Date Aug 29, 2014 (8/13)

Section	Page	Summary
28	1978	explanation of "Counter start delay" improved
29	1988	TAUBn signals interrupt names corrected
29	1993	TAUB2O15 declared as "Error Control Module timer input"
32	2380	correction: no A/D conversion trigger to ADCE1
32	2388	condition for writing to PWGAnCTL corrected
33	2408	DMA trigger ID of INTSG0TI corrected
33	2408	DMA trigger ID of INTSG1TI corrected
33	2408	DMA trigger ID of INTSG2TI corrected
33	2408	DMA trigger ID of INTSG3TI corrected
33	2408	DMA trigger ID of INTSG4TI corrected
33	2409	"XOR Compare Unit check of SG output signals" added
35	2473	DMA trigger ID of INTSSIF0TX corrected
35	2473	DMA trigger ID of INTSSIF0RX corrected
35	2473	DMA trigger ID of INTSSIF1TX corrected
35	2473	DMA trigger ID of INTSSIF1RX corrected
35	2476	"Module Name" column added to register table
35	2476	notes added concerning register names in header files
36	2554	access width of all registers corrected to 32-bit
36	2556	access width of all registers corrected to 32-bit
37	2562	note added concerning concurrent usage of video 0 downscaler and 00 upscaler
37	2563	note added concerning concurrent usage of video 1 downscaler and 10 upscaler
37	2565	four graphics layers configuration 2 added
37	2566	note added concerning concurrent usage of video 0 downscaler and video 00 upscaler
37	2567	one video + up to three graphics layers configuration 2 added
37	2567	note added concerning concurrent usage of video 0 downscaler and video 00 upscaler
37	2568	note added concerning concurrent usage of video 0 downscaler and video 00 upscaler
37	2568	note added concerning concurrent usage of video 1 downscaler and video 01 upscaler
37	2569	cross-reference in note corrected
37	2570	clock name C_ISO_VOnPIXCLK correct to C_ISO_VDCEnCLK
37	2570	clock name C_ISOVO0PIXCLK correct to C_ISO_VDCE0CLK
37	2570	clock name C_ISOVO0PIXCLK in figure correct to C_ISO_VDCE0CLK
37	2570	clock name C_ISOVOnPIXCLK correct to C_ISO_VDCEnCLK
37	2571	C_ISO_VI1PIXCLK selector VDCECTL.VI1CTL added
37	2572	section "Clock selections for different layer configurations" added
37	2575	typo correction (CKSC_IPLL2IND_CTL -> CKDV_IPLL2IND_CTL)
37	2575	typo correction (PLL0PIXCLK -> PLL0CLK)
37	2608	typo correction (channel10 -> channel 1)
37	2608	typo correction (PFCEn_m=01 -> PFCEn_m =0)
37	2612	typo correction (Manximum -> Maximum)
37	2614	Reserved data type error check register (MIPInRDT_ERR_ON) added
37	2617	DT_EN bit added
37	2620	MIPInBUF_CTL register description corrected
37	2629	LP11 reset delay adjustment register (MIPInRESET_DLY_CTL0) description added
37	2630	Reserved data type error check register (MIPIn_RDT_ERR_ON) added

Revision 0.60 History, Date Aug 29, 2014 (9/13)

Section	Page	Summary
37	2642	"MIPI port setup" corrected
37	2642	"Start sequence" corrected
37	2642	"Example start sequence" added
37	2643	"Stop sequence" corrected
37	2643	"Interrupt handling sequence" added
37	2644	"Video Input Setup for MIPI" added
37	2645	"Conditions for MIPI video clock selection" added
37	2647	signal name corrected (VDCE0_VO_TCON0_VS -> VDCE0_VO_TCON0)
37	2647	signal name corrected (VDCE0_VO_TCON2_HS -> VDCE0_VO_TCON2)
37	2647	signal name corrected (VDCE0_VO_TCON3_DE -> VDCE0_VO_TCON3)
37	2649	signal name corrected (VDCE0_VO_TCON0_VS -> VDCE0_VO_TCON0)
37	2649	signal name corrected (VDCE0_VO_TCON2_HS -> VDCE0_VO_TCON2)
37	2649	signal name corrected (VDCE0_VO_TCON3_DE -> VDCE0_VO_TCON3)
37	2650	section "Video output clock inversion in LVTTL mode" added
37	2653	"Module Name" column added to register table
37	2653	note added concerning register names in header files
37	2654	typo correction (VDCE control registers -> VDCE control register)
37	2654	VDCECTL.VI1CTL bit added
37	2655	typo correction (RHSL[1:0] -> RPHSL[1:0])
37	2657	arrow direction between OIR and Video Output Module 0 corrected
37	2658	"Module Name" column added to register table
37	2658	note added concerning register names in header files
37	2660	typo correction (initial value R/W -> 0)
37	2660	typo correction (R -> R/W)
38	2665	TCON0, TCON2 and TCON3 port signal names corrected
38	2665	port signal name corrected (VDCE0_VO_TCON0_VS -> VDCE0_VO_TCON0)
38	2665	port signal name corrected (VDCE0_VO_TCON2_HS -> VDCE0_VO_TCON2)
38	2665	port signal name corrected (VDCE0_VO_TCON3_DE -> VDCE0_VO_TCON3)
38	2665	port signal name corrected (VDCE1_VO_TCON0_VS -> VDCE1_VO_TCON0)
38	2665	port signal name corrected (VDCE1_VO_TCON2_HS -> VDCE1_VO_TCON2)
38	2665	port signal name corrected (VDCE1_VO_TCON3_DE -> VDCE1_VO_TCON3)
38	2668	maximum output video image size corrected
38	2669	Image quality improver blocks corrected
38	2669	correction: "Image Renderer for display (IMR-LSD)" changed to "Video Output Warping Engine (VOWE)"
38	2700	"Module Name" column added to register table
38	2700	"Module Name" column added to register table
38	2701	"Module Name" column added to register table
38	2701	"Module Name" column added to register table
38	2701	note added concerning register names in header files
38	2745	"Module Name" column added to register table
38	2749	"Module Name" column added to register table
38	2753	note added concerning register names in header files
38	2840	"Module Name" column added to register table

Revision 0.60 History, Date Aug 29, 2014 (10/13)

Section	Page	Summary
38	2841	"Module Name" column added to register table
38	2841	note added concerning register names in header files
38	2873	"Module Name" column added to register table
38	2874	"Module Name" column added to register table
38	2875	"Module Name" column added to register table
38	2875	"Module Name" column added to register table
38	2876	"Module Name" column added to register table
38	2877	"Module Name" column added to register table
38	2878	"Module Name" column added to register table
38	2878	"Module Name" column added to register table
38	2879	note added concerning register names in header files
38	2935	"Image renderer for display (IMR-LSD)" corrected to "Video Output Warping Engine (VOWE)"
38	2944	"Module Name" column added to register table
38	2945	note added concerning register names in header files
38	3002	"Module Name" column added to register table
38	3005	"Module Name" column added to register table
38	3006	"Module Name" column added to register table
38	3006	"Module Name" column added to register table
38	3009	"Module Name" column added to register table
38	3010	"Module Name" column added to register table
38	3010	note added concerning register names in header files
38	3072	"Module Name" column added to register table
38	3072	"Module Name" column added to register table
38	3073	note added concerning register names in header files
39	3083	Video Output Warping Engine product specific information added
40	3086	signal name correction (VDCE0_VO_TCON0_VS -> Port VDCE0_VO_TCON0)
40	3086	signal name correction (VDCE0_VO_TCON2_HS -> Port VDCE0_VO_TCON2)
40	3086	signal name correction (VDCE1_VO_TCON0_VS -> Port VDCE1_VO_TCON0)
40	3086	signal name correction (VDCE1_VO_TCON2_HS -> Port VDCE1_VO_TCON2)
40	3099	note 3 added concerning ACTMON_INT error interrupt
40	3112	typo correction (VOCAnDIFF[17:0] -> VOCAnDIFF0[17:0])
41	3127	"Module Name" column added to register table
41	3127	note added concerning register names in header files
42	3158	typo correction (GPU2DCOL_ARGB_OUT_0 -> GPU2D0COL_ARGB_OUT_0)
42	3158	typo correction (GPU2DCOL_ARGB_OP1B_1 -> GPU2D0COL_ARGB_OP1B_1)
42	3162	typo correction (GPU2DCOL_ARGB_OUT_0 -> GPU2D0COL_ARGB_OUT_0)
42	3162	typo correction (GPU2DCOL_ARGB_OP1B_1 -> GPU2D0COL_ARGB_OP1B_1)
43	3164	typo correction (m = 00 to 15 -> m = 0 to 15)
43	3164	Sprite Engine base address corrected
43	3185	description of SPEAnVDCi registers improved
44	3197	typo correction (Code -> Codec)
44	3200	"Module Name" column added to register table
44	3201	JCSWRST register added

Revision 0.60 History, Date Aug 29, 2014 (11/13)

Section	Page	Summary
44	3201	note added concerning register names in header files
44	3202	note changed concerning JCUA reset for mode selection
44	3203	bit 7 of JCCMD register removed
44	3232	JPEG Codec Unit A Software reset register (JCSWRST) register added
44	3234	"Software reset" description changed.
44	3239	"Software reset" description changed.
44	3250	section "Procedure of checking JINTS1" added
44	3250	section name "Bus Reset Processing" changed to "Software Reset Processing"
44	3250	text concerning bus reset changed
45	3252	number range of index j corrected
45	3253	correction: scan group 2 issues ADCE0TRGI2
45	3253	correction: scan group 3 issues ADCE0TRGI3
45	3258	reference to "Self-Diagnostic Function" corrected
45	3260	typo corrections in figure (ADCA -> ADCE)
45	3261	"Module Name" column added to register table
45	3261	addresses of Data register j corrected
45	3262	note added concerning register names in header files
45	3267	address of ADCEnDRj registers corrected
45	3284	caution changed concerning ADCEnSGCRx
45	3307	note added concerning start trigger during scanning
45	3308	typo corrections in figure (ADCA -> ADCE)
45	3309	typo corrections in figure (ADCA -> ADCE, INTADCEnI0 -> INTADCEnI1)
45	3309	typo corrections in figure (ADCA -> ADCE, INTADCEnI0 -> INTADCEnI1)
45	3310	typo corrections in figure (ADCA -> ADCE, INTADCEnI0 -> INTADCEnI1)
45	3311	typo corrections in figure (ADCA -> ADCE, INTADCEnI0 -> INTADCEnI1 or ADCEnI2)
45	3311	typo corrections in figure (ADCA -> ADCE, INTADCEnI0 -> INTADCEnI1)
45	3312	typo corrections in figure (ADCA -> ADCE, INTADCEnI0 -> INTADCEnI1)
45	3313	SG4 interrupt INTADCEnI0 remove from diagram
45	3314	typo correction in diagram (Simultaneous T&H -> A/D conversion)
45	3315	typo correction in diagram (Simultaneous T&H -> A/D conversion)
45	3317	typo corrections in figure (ADCA -> ADCE, INTADCEnI0 -> INTADCEnI1)
45	3317	typo corrections in figure (ADCA -> ADCE, INTADCEnI0 -> INTADCEnI1)
45	3318	typo corrections in figure (ADCA -> ADCE, INTADCEnI0 -> INTADCEnI1)
45	3318	typo corrections in figure (ADCA -> ADCE, INTADCEnI0 -> INTADCEnI1 or ADCEnI2)
45	3319	typo corrections in figure (ADCA -> ADCE, INTADCEnI0 -> INTADCEnI1 or ADCEnI2)
45	3319	typo corrections in figure (ADCA -> ADCE, INTADCEnI0 -> INTADCEnI1 or ADCEnI2)
45	3322	"A/D Error Interrupt Request" description corrected
45	3323	correction in figure: ADCEnDGCTL1.DC14 connected to DIAGOUT2
45	3324	meaning of ADCEnSGVCEPx in diagnostic procedure corrected
45	3325	step 9 of diagnostic procedure corrected
45	3330	usage note added concerning "Analog input (ADCEnIm) pins can be used as port pins"
46	3333	port assignment corrected: motor drivers 2 -> ports ISM5
46	3333	port assignment corrected: motor drivers 3 -> ports ISM6
46	3333	port assignment corrected: motor drivers 4 -> ports ISM3

Revision 0.60 History, Date Aug 29, 2014 (12/13)

Section	Page	Summary
46	3333	port assignment corrected: motor drivers 5 -> ports ISM4
46	3334	note added concerning register names in header files
46	3373	ZIS[1:0] settings corrected
46	3412	ISMnCZISm settings corrected
48	3432	note added concerning NMI for ECM
48	3433	note added concerning CLMA3 error for ECM
48	3433	note added concerning CLMA4 error for ECM
48	3434	note 1. added
48	3434	note 2. added
48	3438	correction: ECMMESET register is W
48	3438	correction: ECMMECLR register is W
48	3438	correction: ECMCESET register is W
48	3438	correction: ECMCECLR register is W
48	3439	initial value of ECMPS corrected
48	3440	typo correction (ECMmEST -> ECMmESET)
48	3442	typo correction (R/W -> R)
48	3446	access mode of ECMMICFG0 register corrected
48	3447	ECMMICFG1 register description modified
48	3447	access mode of ECMMICFG1 register corrected
48	3448	ECMNICFG0 register description modified
48	3448	access mode of ECMNMICFG0 register corrected
48	3449	ECMNICFG1 register description modified
48	3449	access mode of ECMNMICFG1 register corrected
48	3450	access mode of ECMIRCFG0 register corrected
48	3451	access mode of ECMIRCFG1 register corrected
48	3452	access mode of ECMEMK0 register corrected
48	3453	access mode of ECMEMK1 register corrected
48	3458	ECMPE0[31:00] bit description corrected
48	3459	correction: R -> R/W
48	3462	access mode of ECMDTMCFG0 register corrected
48	3463	access mode of ECMDTMCFG1 register corrected
48	3464	access mode of ECMDTMCFG2 register corrected
48	3465	access mode of ECMDTMCFG3 register corrected
48	3465	typos correction (ECMTE[305:301] -> ECMTE[315:311])
50	3475	section "Intelligent Cryptographic Unit (ICU-S2)" completed
51	3478	typo correction (ISMnSVSDIS -> ISMnSVDIS)
51	3481	remark concerning IDMODI write protection added
51	3481	note added concerning register names in header files
51	3482	caution added concerning debugger usage and DEEPSTOP mode
52	3483	D1L2(H) Code Flash size changed to 4 MB
52	3485	start address of block125 and 165 corrected
52	3485	address map for 4 MB devices included
52	3498	note added concerning register names in header files
52	3500	note added concerning register names in header files

Revision 0.60 History, Date Aug 29, 2014 (13/13)

Section	Page	Summary
52	3501	typo corrections (WDT0. -> WDT0)
52	3501	typo correction (WDT1OPWDTRP -> WDT1OPWDTPR)
52	3502	typo correction (WDT1.OPWDOFV[2:0] -> WDT1OPWDOVF[2:0])
52	3502	typo corrections (WDT0. -> WDT0)
52	3502	typo correction (WDT0OPWDTRP -> WDT0OPWDTPR)
52	3502	typo correction (WDT0.OPWDOFV[2:0] -> WDT0OPWDOVF[2:0])
52	3502	note added concerning register names in header files
52	3503	note added concerning register names in header files
52	3504	note added concerning register names in header files
52	3506	"Module Name" column added to register table
52	3506	note added concerning register names in header files
52	3507	typo correction (R -> R/W)
52	3514	"Module Name" column added to register table
52	3514	note added concerning register names in header files
52	3515	bit name corrected: SECDDIS -> SECDIS
52	3515	bit name corrected: SECDDIS -> SECDIS
53	3522	Local RAM described more precisely
53	3522	section "Local RAM initialization" added
53	3524	"Module Name" column added to register table
53	3524	note added concerning register names in header files
53	3529	typo correction (GRAM -> Local RAM)
54	3545	"VRAM Transaction Restrictor" section added
55	3568	description of manufacturer ID by DID.MD[10:0] corrected

Revision 1.00 History

This revision history list shows all modifications of

Rev. 1.00 compared to Rev. 0.60.

NOTE

The chapter and page numbers in the table below refer to the older Rev. 1.00 and thus may not be valid for the current revision.

Revision 1.00 History, Date Feb 27, 2015 (1/37)

Section	Page	Summary
1	91	correction: SSCG0 -> PLL0
1	91	"Operating ambient temperature" removed
1	94	D1M1H CPU frequency increased to 200 MHz
1	95	correction: RS-CANFD available for all products
1	96	correction: SSCG0 -> PLL0
1	96	"Operating ambient temperature" removed
2	100	"Tentative" removed from all pin connection diagrams
2	100	correction in figure: RESETZ is bi-directional
2	101	correction in figure: RESETZ is bi-directional
2	102	correction in figure: RESETZ is bi-directional
2	103	correction in figure: RESETZ is bi-directional
2	104	D1M1H pin connection diagram added
2	107	number of D1M2 port groups corrected
2	107	P40 removed from D1M2 port group list
2	108	note 2 added
2	110	table corrected for PMCn_m=1, PMn_m=1 and PIBCn_m=0
2	119	note 4 added
2	138	Example of Port Configuration Flow (in alternative mode) modified
2	148	TTL(MLB) removed from P21
2	152	correction: P21 TTL(MLB) only for D1M2H
2	163	PDSC21 and PISA21 removed
2	164	correction: bit 8 of PIPC42 available
2	166	PIS44/PISA44 registers added
2	167	PIS45/PISA45 registers added
2	176	PDSC21 and PISA21 removed
2	177	correction: bit 8 and bit 3 of PIPC42 available
2	179	PIS44/PISA44 registers added
2	180	PIS45/PISA45 registers added
2	195	correction: bit 8 and bit 3 of PIPC42 available
2	202	CAN0DREN signal added as 4th alternative output of P0_3
2	202	CAN2DREN signal added as 4th alternative output of P0_7
2	202	CAN1DREN signal added as 4th alternative output of P0_9
2	214	correction: VDCE0 also for D1L2 products
2	222	caution added
2	223	cross references changed

Revision 1.00 History, Date Feb 27, 2015 (2/37)

Section	Page	Summary
2	224	control bit names of Digital Noise Elimination Enable Registers corrected
2	226	correction: DNFA6NFEN1 only for CSIH0SI at port P0_2
2	226	addition: DNFA6NFEN8 for CSIH0SI at port P1_4
2	227	in the table: control bit names of Digital Noise Elimination Enable Registers corrected
2	227	XOR compare unit filters for SG0FAO, SG0FAOL (P3_7), SG0AO (P3_7 corrected
2	228	"Filter Type" section introduced
2	229	"Port Filter Registers" section introduced
2	229	cross references to FCLAnCTLm assignment tables completed
2	229	correction: "7 to 0" -> "7 to 3"
2	229	table title added
2	229	cross reference to filter allocations corrected
2	230	cross references to DNFAnCTLm assignment tables completed
2	230	cross reference to filter allocations corrected
2	231	cross references to DNFAnEN assignment tables completed
2	231	cross reference to filter allocations corrected
2	232	category for P43_[2:0] corrected
2	232	correction: RESET pin has no internal pull-down
2	232	correction: PWRCTL = L while RESET pin = L
2	232	ADCE0Im in DEEPSTOP corrected
2	232	correction: SDRAA[13:11] -> SDRAA[12:11]
2	232	SDRADQ[31:0] pin states corrected
2	232	SDRADQS[3:0] pins removed
2	232	correction: SDRABA[2:0] -> SDRABA[1:0]
2	232	inverted SDRACLK clock removed
2	232	SDRADM[3:0] pin states corrected
2	232	SDRAODT removed
2	232	SDRACS, SDRACAS, SDRARAS, SDRAWE pin states corrected
2	233	note 2 added
2	233	category for P43_[2:0] corrected
2	233	correction: RESET pin has no internal pull-down
2	233	correction: PWRCTL = L while RESET pin = L
2	233	ADCE0Im in DEEPSTOP corrected
2	233	correction: SDRAA[13:11] -> SDRAA[12:11]
2	233	SDRADQ[31:0] pin states corrected
2	233	SDRADQS[3:0] pins removed
2	233	correction: SDRABA[2:0] -> SDRABA[1:0]
2	233	inverted SDRACLK clock removed
2	233	SDRADM[3:0] pin states corrected
2	233	SDRAODT removed
2	233	SDRACS, SDRACAS, SDRARAS, SDRAWE pin states corrected
2	234	note 3 added
2	234	category for P43_[2:0] corrected
2	234	correction: RESET pin has no internal pull-down
2	234	correction: PWRCTL = L while RESET pin = L

Revision 1.00 History, Date Feb 27, 2015 (3/37)

Section	Page	Summary
2	234	ADCE0Im in DEEPSTOP corrected
2	235	correction: SDRAA[13:11] -> SDRAA[12:11]
2	235	SDRADQ[31:0] pin states corrected
2	235	SDRADQS[3:0] pins removed
2	235	correction: SDRABA[2:0] -> SDRABA[1:0]
2	235	inverted SDRACLK clock removed
2	235	SDRADM[3:0] pin states corrected
2	235	SDRAODT removed
2	235	SDRACS, SDRACAS, SDRARAS, SDRAWE pin states corrected
2	235	note 1 added
2	236	handling recommendation of PWRGD pin corrected
2	236	recommendations for unused SDRA pins included
2	237	recommendation for unused SDRAVCC pin included
2	237	recommendation for unused ZPDREF changed
3	239	CPU Subsystem block diagram corrected
3	241	correction: PBUS modules in 4 functional segments, distributed to five separate PBUSes.
3	242	SDR-SDRAM Memory Controller (SDRA) added to caution list
3	243	"Retention RAM (RRAM)" added
3	246	SECDIS (bit 1) removed
3	246	SECDIS bits
3	251	typo correction (CF1STERSTR -> ID1STERSTR)
3	254	correction: "(for bank 0)" removed
3	255	correction: "(for bank 0)" removed
3	256	correction: "error status clear register" -> "error status register"
3	257	"or an address parity error" removed
3	258	"Retention RAM (RRAM)" access through XC Cache added
3	258	correction: "Granularity is 1 MB" -> "Minimum granularity is 1 MB"
3	258	XC cache buffered mode added
3	259	"Cache/Buffer window set-up" corrected
3	259	"XC Cache latency" added
3	260	devices with XC cache control registers specified
3	260	"Buffer flush register" added in table
3	260	note added concerning XC cache control registers
3	261	BUF0 and BUF1 bits added
3	262	AXCSYSERR initial value corrected
3	262	ETYPE[11:8] added
3	265	"AXCBUFFLUSH — Buffer flush register" added
3	277	XC cache ECC control registers added
3	293	"Usage Notes" added
4	297	change: "(indicated by the error monitor bit set to “1”)" -> "(the error monitor bit set to “1”)"
4	297	"Emulation break during the protection unlock sequence" modified
4	298	table title changed
4	299	APB_CLK_RATIO register added

Revision 1.00 History, Date Feb 27, 2015 (4/37)

Section	Page	Summary
4	299	correction: SELBIDMODI -> IDMODI
4	304	PROTERR description changed
4	306	PROTERR description changed
4	307	PROTDERR description changed
4	308	CLMAnPRERR description changed
4	309	CLMATPRERR description changed
4	310	CLMATPRERR2 description changed
4	311	CLMATPRERR3 description changed
4	313	bit name corrected: MRSTCPRERR -> MRSTCPERR
4	313	MRSTCPERR description changed
4	313	APB_CLK_RATIO register added
4	313	APB_CLK_RATIO register added
4	314	APB_CLK_RATIO register added
4	314	PWRGDPRERR description changed
4	314	correction: SELBIDMODI -> IDMODI
4	314	correction: SELBIDMODI -> IDMODI
4	315	correction: SELBIDMODI -> IDMODI
4	315	bit name corrected: IDMODIPRERR -> IDMODIPERR
4	315	IDMODIPERR description changed
4	317	PPROTSnPRERR description changed
4	319	FLMDPRERR description changed
5	321	CPU address map corrected
5	321	note added concerning initial value of CPU reset vector
5	324	corrected start address of D1L1 Local RAM and exchanged VRAM0 and VRAM1 in D1M2(H) address map
6	325	note 1 added
6	325	notes 1 and 2 added
7	334	caution concerning tentativeness of interrupt tables removed
7	336	correction: INTDCUTDI is level interrupt
7	338	Flash sequencer ready (INTFLENDNM0 interrupt added
7	343	correction: INTDCUTDI is level interrupt
7	345	Flash sequencer ready (INTFLENDNM0 interrupt added
7	350	correction: INTDCUTDI is level interrupt
7	352	Flash sequencer ready (INTFLENDNM0 interrupt added
7	357	correction: INTDCUTDI is level interrupt
7	359	Flash sequencer ready (INTFLENDNM0 interrupt added
7	360	Underflow of auto refresh - level interrupt (INTSDRA) added
7	364	correction: INTDCUTDI is level interrupt
7	366	Flash sequencer ready (INTFLENDNM0 interrupt added
7	371	correction: INTDCUTDI is level interrupt
7	373	Flash sequencer ready (INTFLENDNM0 interrupt added
7	376	typo correction (indicated -> indicate)
7	377	BERR0ST1 register description improved
7	379	typo correction (BUERR0ST0 -> BERR0ST0)

Revision 1.00 History, Date Feb 27, 2015 (5/37)

Section	Page	Summary
7	380	typo correction (BUERR0ST1 -> BERR0ST1)
7	386	typo correction ("BECM0BUSERRINF0 and BECM0BUSERRINF1" -> "BECMBUSERRINFO0 and BECMBUSERRINFO1")
8	418	DMnnCM initial value corrected
8	418	PEID[2:0] description corrected
8	424	typo correction (SHNSEL0 -> CHNSEL0)
9	438	"Clock Supply" added
9	439	abbreviations "RESC" and "RESCR" removed from diagram
9	439	modification in diagram: circle instead of solid point as signal inverter
9	439	modification in diagram: SDRB0RES -> SDRA0RES/SDRB0RES
9	439	typo correction (RESET -> RESET with overbar)
9	439	note 5 added
9	440	ISOPWRES added as reset source
9	440	"Retention RAM RRAM" removed from table
9	440	DBRES effect on Low Speed IntOsc corrected
9	440	DBRES effect on High Speed IntOsc corrected
9	441	correction: POC0RES does not assert ISOPWRES
9	441	modification in diagram: circle instead of solid point as signal inverter
9	442	D1M2(H) Isolated-Area power reset description corrected
9	442	modification in diagram: circle instead of solid point as signal inverter
9	443	typo correction ("Wake-up from DEEPSTOP PWRCTL/" -> "Wake-up from DEEPSTOP")
9	444	typo correction (programing -> programming)
9	445	note 1 added
9	445	typo correction (SGnRES -> SGn)
9	445	correction: "Memory Controller SDRB0" -> "Memory Controller SDRB0 PHY"
9	445	note added concerning SDRA0 reset
9	445	typo correction (cross-conenct -> cross-connect)
9	445	(RESC) removed
9	445	(RESCR) removed
9	445	correction: RESC -> Reset Controller
9	445	correction: RESCR -> Reset Controller
9	447	PWRG protection command and status registers added
9	448	note 2 added
9	448	note 2 about initial value for RESF/RESFR added
9	448	RESF8 initial value corrected
9	448	typo correction (13 to 12 -> 13 to 11)
9	449	note changed
9	450	rephrased RESFC, RESFCR registers explanation
9	450	typo correction ("13 to 12" -> "13 to 11")
9	453	note added concerning reset of unused modules
9	453	typo correction (PROTCMDMRSTC -> PROTCMDMRST)
9	453	XC0RES description corrected
9	453	addition: SDRB0RES not for all products available
9	453	correction: "Memory Controller SDRB0" -> "Memory Controller SDRB0 PHY"

Revision 1.00 History, Date Feb 27, 2015 (6/37)

Section	Page	Summary
9	453	note added concerning SDRA0 reset
9	453	addition: MLBB0RES not for all products available
9	454	SG3RES description corrected
9	454	SG2RES description corrected
9	454	addition: VDCE1RES not for all products available
9	454	addition: VDCE0RES not for all products available
9	455	reset source for register initialization added
9	456	(RESC) removed
9	456	(RESCR) removed
9	456	description about REGVCC < VPOC modified
9	458	"Overview of CPU System Startup after Power-On-Clear" added
9	458	correction in note 1: "RESFR.RESFR8 is" -> "RESF8.RESF8 and RESFR.RESFR8 are"
9	461	typo correction (CLM0RES -> CLMA0RES)
9	461	typo correction (CLM0RES -> CLMA0RES)
9	461	additional note concerning DBRES and IntOsc
10	462	B0VCC supply for D1L2 port group P42 corrected
10	462	typo correction ("port groups 46, 47" -> "port groups P46, P47")
10	463	correction: REG1VCC for D1M1(H), but not for D1M2 devices
10	466	correction in diagram: "VI0 MIPI I/F port buffers" assigned to note 2 instead of note 3
12	473	PLL1 nominal frequencies corrected
12	473	typo correction (VDCE1_VI-CLK -> VDCE1_VI_CLK)
12	476	change: "If the High Speed IntOsc clock ..." -> "Only if the High Speed IntOsc clock ..."
12	476	correction: High Speed IntOsc clock fRH stop is indicated by ROSC.ROSCCLKACT = 0
12	477	D1M1H Clock Controller block diagram separated
12	478	CKSC_IPLL1S_CTL location corrected
12	478	change: "If the High Speed IntOsc clock ..." -> "Only if the High Speed IntOsc clock ..."
12	478	correction: High Speed IntOsc clock fRH stop is indicated by ROSC.ROSCCLKACT = 0
12	479	typo correction (comain -> domain)
12	480	APB_CLK_RATIO selector included in diagram
12	481	typo correction (CLMA3RES -> INTCLMA3TI)
12	481	typo correction (CLMA4RES -> INTCLMA4TI)
12	481	typo correction ("CLMA3RES or CLMA4RES" -> "INTCLMA3TI or INTCLMA4TI")
12	481	typo corrections in diagram 9 (CLMA3RES -> INTCLMA3TI, CLMA4RES -> INTCLMA4TI)
12	482	typo correction ("CLMA3RES or CLMA4RES" -> "INTCLMA3TI or INTCLMA4TI")
12	482	caution added
12	483	correction: "It can not be stopped by software, but can be automatically ..." -> "It can be automatically ..."
12	484	typo correction ("stop mask change" -> "clock selector change")
12	485	"when a waveform is output from the MainOsc" added
12	486	"and the oscillation stabilization time is counted" added
12	486	typo correction (MOSCSTPM.MOSCSTPMSK = 1: STOP request signal is masked)
12	487	"SubOsc stabilization" corrected
12	487	"SubOsc STOP requests in DEEPSTOP mode" removed
12	488	exchanged "SubOsc enable trigger" with "SubOsc enable trigger/disable trigger"

Revision 1.00 History, Date Feb 27, 2015 (7/37)

Section	Page	Summary
12	489	correction in diagram: POC0RES -> PURES
12	489	typo correction ("After reset release after release from Power-On-Clear 0 reset (POC0RES)" -> "After reset release from Power-up reset (PURES)"
12	489	note concerning High Speed IntOsc start/stop changed
12	495	typo correction (fPPLkCLK -> fPLLkCLK)
12	495	fractional part PLLkCNF removed from PLL0 and PLL1
12	495	PLL1 correction: PLL1 settings are product dependend
12	495	typo correction (PLL1C.NI[5:0] -> PLL2C.NI[5:0])
12	495	Pr value tables separated
12	496	PLL1 frequency ranges corrected
12	497	SubOsc stop mask register removed from table
12	497	SubOsc stabilization time register added in table
12	498	PBUS clock ratio selection register added
12	498	typo correction (CKSC_AAWOTD_STMP -> CKSC_AAWOTD_STPM)
12	498	CKSC_AWDTA0D_ACT register added
12	499	Write protection registers list added
12	500	note 1 concerning stopping MainOsc changed
12	502	addition concerning MOSCC writing
12	503	addition concerning MOSCST writing
12	503	typo correction (ROSCCLKSCT -> ROSCCLKST)
12	503	typo correction (ROSCCLKSCT -> ROSCCLKST)
12	504	typo correction (MOSCSTMP -> MOSCSTPM)
12	505	note 1 concerning stopping SubOsc changed
12	506	section "SOSCSTPM — SubOsc Stop Mask Register" removed
12	507	"SOSCST — SubOsc stabilization time register" added
12	508	reset sources of ROSCE corrected
12	508	note 1 concerning stopping High Speed IntOsc changed
12	509	reset sources of ROSCS corrected
12	509	section "ROSCST — High Speed IntOsc stabilization time register" removed
12	510	reset sources of ROSCSTPM corrected
12	511	note 1 concerning stopping PLL0 changed
12	513	typo correction (fPPL0CLK -> fPLL0CLK)
12	513	fractional part PLL0.NF[3:0] removed
12	513	frequency range for FVV[2:0] settings corrrected
12	515	frequency range for P[2:0] settings corrrected
12	516	NI[6:0] settings corrected
12	519	PLL1C description separated for different devices and corrected all settings
12	519	typo correction (fPPL1CLK -> fPLL1CLK)
12	519	fractional part PLL1.NF[3:0] removed
12	525	typo correction (fPPL0CLK -> fPLL2CLK)
12	526	NI[5:0] settings corrected
12	528	note added concerning CKSC_IPLL0S_ACT register
12	530	note added concerning CKSC_IPLL1S_ACT register
12	532	CKDV_ICLKJITD_STAT initial value corrected

Revision 1.00 History, Date Feb 27, 2015 (8/37)

Section	Page	Summary
12	536	CKDV_ICLKFIXD_STAT initial value corrected
12	539	addition: PLL2CLK selection only for D1M2(H)
12	541	addition: CKSC_ISDRBS_CTL not for all products available
12	542	addition: CKSC_ISDRBS_ACT not for all products available
12	543	note 1 added
12	543	caution added
12	551	addition: CKSC_IXCETNBS_CTL not for all products available
12	552	addition: CKSC_IXCETNBS_ACT not for all products available
12	553	addition: CKSC_IPCMLBBS_CTL not available for all products
12	554	addition: CKSC_IPCMLBBS_ACT not available for all products
12	555	addition: CKSC_IXCMLBBS_CTL not available for all products
12	556	addition: CKSC_IXCMLBBS_ACT not available for all products
12	556	CKSC_IXCMLBBS_ACT initial value corrected
12	559	APB_CLK_RATIO register description added
12	564	AWOTDSTPMSK description changed
12	567	typo correction ("High Speed IntOsc fRL" -> "High Speed IntOsc fRH"
12	567	note added
12	571	RTCADSTPMSK description changed
12	572	addition: PLL2CLK/4 selection only for D1M2(H)
12	572	caution added concerning PLL0CLK and PLL1CLK FOUT
12	575	addition: CKSC_IMLBBS_CTL not for all products available
12	576	addition: CKSC_IMLBBS_ACT not for all products available
12	578	CKSC_ISFMAS_ACT initial value corrected
12	579	SFMADCSID[2:0] description corrected
12	580	typo correction (ISFMAD -> SFMAD)
12	581	CKSC_IRSCAND_CTL initial value corrected
12	582	CKSC_IRSCAND_ACT initial value corrected
12	582	correction: RSCANDACT[1:0] -> RSCANDACT[2:0]
12	582	note added concerning CKSC_IRSCAND_ACT register
12	584	CKSC_IRSCANXINS_ACT initial value corrected
12	596	typo correction ("C_ISO_LCBI clock active register" -> "C_ISO_LCBI source clock active register")
12	596	CKSC_ILCBIS_ACT initial value corrected
12	598	note added concerning CKSC_IADCED_ACT register
12	601	CKSC_IRLINS_CTL initial value corrected
12	601	RLINSCSID[1:0] description corrected
12	603	PLLk and SubOsc configuration order swapped
12	603	typo correction (CKSC_IPLLkS_CTL.PLLkSTP -> CKSC_IPLLkS_CTL.PLLkSSTP)
12	603	item added in "Configure the SubOsc"
12	603	actions order to activate C_ISO_XCCLK changed
12	603	typo correction ("the the" -> "the")
12	605	typo correction (regsiters -> registers)
12	605	cautions added concerning changing the source clock via the clock selectors and clock supply for registers access
12	606	targets for SDRBCLK clock corrected

Revision 1.00 History, Date Feb 27, 2015 (9/37)

Section	Page	Summary
12	606	typo correction (regsiters -> registers)
12	606	"Base clocks maximum frequencies" table added
12	606	order of divisors changed
12	606	addition: ETNBXCCLK not for all products available
12	606	addition: MLBBCLK not for all products available
12	606	addition: MLBBXCCLK not for all products available
12	607	"CPU and buses subsystems clocks maximum frequencies" table added
12	607	correction "Each AWO clock domain ..." -> "Each AWO clock domain except for C_AWO_WDTA0 ..."
12	607	order of selections changed
12	607	divisor options corrected
12	608	order of selections changed
12	608	addition: PLL2CLK/4 not for all products available
12	608	typo correction (regsiters -> registers)
12	608	"Always-On-Area clock domain clocks maximum frequencies" table added
12	608	CKSC_IRSCAND_CTL default value corrected
12	609	typo correction (C_ISO_SOSTM -> C_ISO_OSTM)
12	609	typo correction (regsiters -> registers)
12	609	note 3 added
12	610	"Isolated-Area clock domain clocks maximum frequencies" table added
12	611	divider range corrected: "1 to 511" -> "1 to 512"
12	611	FOUTDIV.FOUTDIV[x:x] corrected to FOUTDIV.FOUTDIV[9:0]
12	612	FOUTDIV reset source corrected
12	612	note *1 added and 000H declared as default
12	613	correction: "This register defines the clock divisor." -> "This register defines the clock divider status."
12	613	FOUTSTAT reset source corrected
12	613	FOUTCLKACT description corrected
12	613	FOUTSYNC description corrected
12	614	number of clock monitor units corrected
12	614	CLMAn clock supply table completed
12	617	addition: CLMAOTCTL0 not for all product available
12	620	note added concerning PLL jitter
12	621	"Notification of Abnormal Clock Frequency" corrected
12	621	"CLMAn Enable (Write to CLMAnCTL0)" description changed
12	621	"(1) Initial value of CLMAnCTL0" and "(2) Procedures to enable CLMAn" removed
12	622	"Write protection registers" added in table
12	622	reference to register base addresses table corrected
12	623	reference to "Write-Protected Registers" section corrected
12	623	change: readable/writable -> read/written
12	623	note added concerning reset source of CLMA[2:0]CTL
12	624	reference to calculation methods corrected
12	624	change: readable/writable -> read/written
12	624	note added concerning reset source of CLMA[2:0]CMPH
12	624	CLMAnCMPH[11:0] formula corrected

Revision 1.00 History, Date Feb 27, 2015 (10/37)

Section	Page	Summary
12	625	reference to calculation methods corrected
12	625	change: readable/writable -> read/written
12	625	note added concerning reset source of CLMA[2:0]CMPL
12	626	explanation concerning register write protection changed
12	626	reset sources for CLMATEST corrected
12	626	typo correction ("Value after reset" -> "Initial value")
12	626	typo correction ("Value after reset" -> "Initial value")
12	626	typo correction ("1:" -> "1")
12	626	RESCLM description corrected
12	627	reset sources for CLMATESTS corrected
12	627	typo correction ("Value after reset" -> "Initial value")
12	627	typo correction ("Value after reset" -> "Initial value")
12	628	explanation concerning register write protection changed
12	628	reset sources for CLMATEST2 corrected
12	628	typo correction ("Value after reset" -> "Initial value")
12	628	typo correction ("Value after reset" -> "Initial value")
12	628	typo correction ("1:" -> "1")
12	628	RESCLM description corrected
12	629	reset sources for CLMATESTS2 corrected
12	629	typo correction ("Value after reset" -> "Initial value")
12	629	typo correction ("Value after reset" -> "Initial value")
12	630	explanation concerning register write protection changed
12	630	reset sources for CLMATEST3 corrected
12	630	typo correction ("Value after reset" -> "Initial value")
12	630	typo correction ("Value after reset" -> "Initial value")
12	630	addition: CLMA6TESEN not for all products available
12	630	addition: CLMA5TESEN not for all products available
12	630	typo correction ("1:" -> "1")
12	630	RESCLM description corrected
12	631	reset sources for CLMATESTS3 corrected
12	631	typo correction ("Value after reset" -> "Initial value")
12	631	typo correction ("Value after reset" -> "Initial value")
12	631	addition: CLMA6ERRS not for all products available
12	631	addition: CLMA5ERRS not for all products available
12	632	change: readable/writable -> read/written
13	633	"Clock Supply" added
13	634	caution added concerning usage of external interrupts for DEEPSTOP wake-up
13	634	note added concerning OCD wake-up by breakpoint hit
13	635	correction: "In order to clear a wake-up factor flag of the wake-up factor register (WUF0), its assigned bit has to be set to 1." moved under previous bullet paragraph
13	635	typo correction ("does not mean, that the" -> "does not mean that the")
13	635	improved description concerning setting of wake-up factor bits
13	635	sentence concerning I/O buffers in DEEPSTOP mode rephrased
13	636	notes concerning I/O buffers in DEEPSTOP mode corrected

Revision 1.00 History, Date Feb 27, 2015 (11/37)

Section	Page	Summary
13	636	correction: "Thus it is recommended to switch off the the DDR2-SDRAM I/F supply SDRBVCC." -> "Thus the DDR2-SDRAM I/F supply SDRBVCC must be switched off."
13	636	"SOSCSTPM: SubOsc" removed
13	637	note 6 added
13	639	"Stand-by Controller Registers overview" moved in separate section
13	639	correction: WUFMSK0 initial value is FFFF FFFFH
13	641	Initial value corrected
13	641	correction: bits 31 to 19 are reserved
13	642	correction: bits 31 to 19 are reserved
13	643	typo correction ("WUFC0:" removed)
13	643	correction: bits 31 to 19 are reserved
14	651	"List of XC1 QoS Register" added
14	658	SFMA and module registers access options corrected
14	660	SFMA and module registers access options corrected
14	663	CPU-Subsystem, SFMA, SDRA and module registers access options corrected
14	668	CPU-Subsystem, SFMA and module registers access options corrected
14	668	correction: GPU2D and MLBB can write to CPU Subsystem
14	668	Note 5 for Sprite Unit 2 corrected
14	671	JCUA latency numbers corrected
14	675	correction: "ECCRAN0" -> "ECCRCAN0 / ECCRCFD0,1*1"
14	675	correction: "RSCAN" -> "RSCAN / RSCFD0*1"
14	678	correction: P50A_ERR_SLV -> P32A_ERRSLV
14	678	correction: P50B_ERR_SLV -> P32B_ERRSLV
14	681	PEID[2:0] description corrected
14	682	note added
14	684	addition concerning PEG Protection set up
14	685	typo correction (GnMask -> PEGGnMK)
14	686	note added
14	687	"Internal Peripheral Guard error signals" section removed
14	703	note added concerning XCG for SDR-SDRAM Memory Controller (SDRA)
14	708	note added concerning accesses after register modifications
14	716	typo correction ("Read error address register" -> "Write error address register")
14	718	typo correction ("Read error type register" -> "Write error type register")
15	719	"SDR-SDRAM Memory Controller (SDRA)" completed
15	720	Underflow flag of auto refresh cycle interrupt (INTSDRA) added
15	736	LV1TH condition corrected
15	740	"xx = 00B: 8 bit memory width" removed
15	740	"y = 1: 8 banks" removed
16	755	note added concerning accesses after register modifications
16	797	PHYBUS_ WIDTH description corrected
17	825	typo correction (SPBMO0 -> SPBMO1)
17	825	typo correction (SPBMI0 -> SPBMI1)
17	827	SPODLY and CLKDLY registers in block diagram included
17	828	SMWDR0/1 access size corrected

Revision 1.00 History, Date Feb 27, 2015 (12/37)

Section	Page	Summary
17	828	SPODLY initial value corrected
17	830	CPHAT-CPHAR settings corrected
17	830	typo correction (COHAR -> CPHAR)
17	834	BFM[2:0] description corrected
17	834	typo correction (Furhter -> Further)
17	835	RBE decription corrected
17	857	SPODLY initial value corrected
17	860	typo correction (DRECR -> DRCR)
17	862	data width corrected in figure
17	865	SPI command issue and AC characteristics set up included in flow
17	866	correction: "purpose" -> "tag address"
17	867	correction: "set to 1 to when the TEND bit is set to 1" -> "when the TEND bit in CMNSR is set to 1"
17	868	AC characteristics set up included in flow
17	881	correction: "SPBCLK is maximum 40 MHz." -> "SPBCLK is maximum 40 MHz by SPBR[7:0] = 1."
17	882	typo correction (CKLDYOC[2:0] -> CKDLYOC[2:0])
17	882	typo correction (CKLDYRX[2:0] -> CKDLYRX[2:0])
17	882	typo correction (CKLDY.CKLDYRX[2:0] -> CKDLY.CKDLYRX[2:0])
18	886	CSIGTSSO signals removed from table
18	889	"Functional Overview Description" modified
18	890	"Main Registers of CSIG" table title added
18	891	note added
18	896	"over clearing to 0" removed
18	897	"over clearing to 0" removed
18	897	"over clearing to 0" removed
18	899	reference to "Section 18.5.7, Communication in Slave Mode" removed
18	903	change: "This register is initialized when ..." -> "These bits are initialized when ..."
18	903	change: "values of the register must be read" -> "values of these bits must be read"
18	905	correction: "When writing, always write 0." -> "When writing, always write 1."
18	905	CSIGnBCTL0.CSIGnSCE write caution changed
18	910	"When the BRG is disabled, CSIGTSCK stays on the level specified by CSIGnCTL1.CSIGnCKR." removed
18	911	"in transmit-only mode or transmit/receive mode (CSIGnCTL0.CSIGnTXE = 1)" added
18	913	text order changed
18	913	correction: "it is not necessary to write data to the CSIGnTX0W or CSIGnTX0H register of the slave" -> "it is not necessary to read data from the CSIGnRX0 register of the slave"
18	913	correction: "In receive-only mode, any previously ..." -> "In slave mode, any previously ..."
18	916	figure corrected
18	916	figure corrected
18	920	diagram corrected
18	921	change: "Overrun error" -> "Overrun error (received data)"
18	922	addition concerning parity check in EDL
18	924	diagram corrected
18	925	typo correction (INTCSIGTIC -> INTCSIGTIR)

Revision 1.00 History, Date Feb 27, 2015 (13/37)

Section	Page	Summary
18	925	step 6 changed
19	930	CSIH FIFO size corrected to 16
19	931	"Main Registers of CSIH" table title added
19	932	RAM size in diagram corrected to "16 x 32 bit"
19	932	note added concerning FIFO buffer size
19	934	note added
19	937	change: "handshake mode" -> "handshake function"
19	937	change: "Input signal CSIHTSSI is ignored" -> "Input signal CSIHTSSI is disabled"
19	940	text added concerning time-out error in slave mode
19	942	CSIHnTSF description changed
19	943	change: "Transmission is in progress or being prepared" -> "Communication is in progress or being prepared"
19	954	"requires that the EDL function is used (see also Section 19.5.8.2, Data Length Greater than 16 Bits)." removed
19	965	"Transmit-only buffer" and "Dual buffer" separated
19	966	"(The setting of the CSIHnSIT bit is invalid in slave mode.)" added
19	974	change "The example below shows the communication in master mode for 8 data bits" -> "The example below shows the communication in master mode for 8-bit data"
19	975	addition to note
19	976	change: "by using the bit CSIHnCTL1.CSIHnSSE" -> "by using the CSIHnCTL1.CSIHnSSE bit"
19	977	"in transmit-only mode or transmit/receive mode (CSIHnCTL0.CSIHnTXE = 1)" added
19	980	correction in diagram: "Inter-data time" -> "Idle time"
19	986	"The size is 128 words." removed
19	987	typo correction (CSIHnMCTL2.CSIHnND[6:0] -> CSIHnMCTL2.CSIHnND[7:0])
19	987	ranges corrected
19	987	typo correction (CSIHnMCTL2.CSIHnND[6:0] -> CSIHnMCTL2.CSIHnND[7:0])
19	988	change: "when bit CSIHnMCTL2.CSIHnBTST is set" -> "when the CSIHnMCTL2.CSIHnBTST bit is set"
19	988	change: "when bit CSIHnMCTL2.CSIHnBTST is set" -> "when the CSIHnMCTL2.CSIHnBTST bit is set"
19	988	change: "when bit CSIHnMCTL2.CSIHnBTST is set" -> "when the CSIHnMCTL2.CSIHnBTST bit is set"
19	989	change: "setting bit CSIHnCTL1.CSIHnEDLE to 1" -> "setting the CSIHnCTL1.CSIHnEDLE bit to 1"
19	991	diagram corrected
19	992	change: "CSIH is set to transmit/receive mode" -> "The transmit/receive mode is set"
19	992	change: "The “transmission in progress” flag" -> "The transfer status flag"
19	993	change: "by bit CSIHnCTL1.CSIHnHSE" -> "by the CSIHnCTL1.CSIHnHSE bit"
19	993	change: "The timing depends on the data phase selection bit, CSIHnCFGx.CSIHnDAPx setting." -> "The busy timing depends on the data phase selection bit CSIHnCFGx.CSIHnDAPx."
19	996	"while CSIHnCTL1.CSIHnHSE = 1" added
19	997	change: "by bit CSIHnCTL1.CSIHnDCS" -> "by the CSIHnCTL1.CSIHnDCS bit"
19	997	change: "Bit CSIHnSTR0.CSIHnDCE is set." -> "The CSIHnSTR0.CSIHnDCE bit is set."
19	998	change: "Bit CSIHnSTR0.CSIHnPE is set." -> "The CSIHnSTR0.CSIHnPE bit is set."
19	1000	change: "Bit CSIHnSTR0.CSIHnPCT is set." -> "The CSIHnSTR0.CSIHnPCT bit is set."

Revision 1.00 History, Date Feb 27, 2015 (14/37)

Section	Page	Summary
19	1000	change: "Bit CSIHnSTR0.CSIHnTMOE is set." -> "The CSIHnSTR0.CSIHnTMOE bit is set."
19	1000	addition concenring CSIH0 dedicated time-out counter
19	1000	addition concenring CSIH1 dedicated time-out counter
19	1001	change: "Bit CSIHnSTR0.CSIHnOFE is set." -> "The CSIHnSTR0.CSIHnOFE bit is set."
19	1004	change: "Bit CSIHnSTR0.CSIHnOVE is set." -> "The CSIHnSTR0.CSIHnOVE bit is set."
19	1005	explanation concerning loop-back mode changed
19	1005	diagram corrected
19	1008	typo correction (CSIHTIJC -> INTCSIHTIJC)
19	1011	typo correction ("CS0 to CS3" -> "CSIHTCSS0 to CSIHTCSS3"
19	1011	"bits" removed
19	1011	change: "CSIHnRXE = 1 (permits reception), CSIHnMBS = 1" -> "CSIHnRXE = 1 (permits reception), and CSIHnMBS = 1"
19	1014	CSIHnMCTL2.CSIHnSOP[6:0] values in diagram corrected
19	1014	typo correction ("CS0 to CS3" -> "CSIHTCSS0 to CSIHTCSS3"
19	1014	"the bits" removed
19	1014	change: "by setting bits CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0]" -> "by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits"
19	1014	change: "Bit CSIHnMCTL2.CSIHnSOP[6:0] is automatically" -> "The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically"
19	1014	change: "and bit CSIHnMCTL2.CSIHnND[7:0] is decremented" -> "and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented"
19	1015	CSIHnMCTL2.CSIHnSOP[6:0] values in diagram corrected
19	1016	typo correction ("CS1, CS3 and CS7" -> "CSIHTCSS0, CSIHTCSS3, and CSIHTCSS7"
19	1016	"bits" removed
19	1016	change: "by setting bits CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0]" -> "by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits"
19	1016	change: "Bit CSIHnMCTL2.CSIHnSOP[6:0] is automatically incremented and bit CSIHnMCTL2.CSIHnND[7:0] is decremented after each data transmission" -> "The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission"
19	1017	CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMRWP0.CSIHnRRA[6:0] values in diagram corrected
19	1018	typo correction ("CS0 to CS3" -> "CSIHTCSS0 to CSIHTCSS3"
19	1018	"bits" removed
19	1018	change: "by setting CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0]" -> "by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits"
19	1018	change: "Bit CSIHnMCTL2.CSIHnSOP[6:0] is automatically incremented and bit CSIHnMCTL2.CSIHnND[7:0] is decremented after each data transmission." -> "The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission."
19	1019	CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMRWP0.CSIHnRRA[6:0] values in diagram corrected
19	1019	typo correction ("CS0 to CS7" -> "CSIHTCSS0 to CSIHTCSS7"
19	1019	"bits" removed
19	1019	change: "permits reception" -> "permits the reception"

Revision 1.00 History, Date Feb 27, 2015 (15/37)

Section	Page	Summary
19	1019	change: "by setting bits CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0]" -> "by setting the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits"
19	1019	change: "Bit CSIHnMCTL2.CSIHnSOP[6:0] is automatically incremented, and bits CSIHnMCTL2.CSIHnND[7:0] is decremented" -> "The CSIHnMCTL2.CSIHnSOP[6:0] are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented"
19	1021	CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMRWP0.CSIHnRRA[6:0] values in diagram corrected
19	1021	"bits" removed
19	1021	change: "permits reception" -> "permits the reception"
19	1021	change: "by setting CSIHnMCTL2.CSIHnSOP[6:0] and the number of data by setting CSIHnMCTL2.CSIHnND[7:0]." -> "by setting the CSIHnMCTL2.CSIHnSOP[6:0] bits and the number of data by setting the CSIHnMCTL2.CSIHnND[7:0] bits."
19	1021	change: "Bit CSIHnMCTL2.CSIHnSOP[6:0] is automatically incremented and bit CSIHnMCTL2.CSIHnND[7:0] is decremented after each data transmission" -> "The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented after each data transmission"
19	1023	typo correction ("CS0 to CS3" -> "CSIHTCSS0 to CSIHTCSS3"
19	1025	CSIHTSOE signal removed from diagram
19	1025	typo correction ("CS0 to CS7" -> "CSIHTCSS0 to CSIHTCSS7"
19	1025	"bit" removed
19	1025	change: "Make sure CSIHnSTR0.CSIHnFLF" -> "Make sure that CSIHnSTR0.CSIHnFLF"
19	1025	"bits" removed
20	1027	typo correction (RLIN3nLCUC -> RLN3nLCUC)
20	1027	typo correction (RLIN3nLDBR1 -> RLN3nLDBRb)
20	1034	correction: "LIN / UART ID buffer register" -> "LIN ID buffer register"
20	1045	correction in entire table: "interrupt" -> "interrupt request"
20	1045	typo correction ("interrupt" -> "interrupt request")
20	1045	correction: "Interrupt" -> "Interrupt Request"
20	1045	correction: "Interrupt" -> "Interrupt Request"
20	1045	correction: "Interrupt" -> "Interrupt Request"
20	1046	correction: "Interrupt" -> "Interrupt Request"
20	1066	"to be received as expected value" removed
20	1073	"However, when this bit is 1, an interrupt is not generated upon completion of transmission from the UART buffer." removed
20	1078	"Setting range: 00H to FFH." added
20	1078	explanation concerning bit arrangement in RLN3nLUDB0 added
20	1079	"Setting range: 00H to FFH." added
20	1079	explanation concerning bit arrangement in RLN3nLUDB0 added
20	1080	"UROE Bit (Reception Enable)" explanation improved
20	1089	paragraph concerning prescaler settings removed
20	1095	correction in diagram: Synchronized RLIN3RX adjusted
20	1098	typo correction in diagram (RLIN3nLST -> RLN3nLST)
20	1108	"Synchronized RLIN3nRX" added in diagram
20	1110	correction in diagram: "1 or 2 stop bits" -> "stop bit"
20	1111	timing relation of signals in diagram adjusted

Revision 1.00 History, Date Feb 27, 2015 (16/37)

Section	Page	Summary
20	1112	correction in diagram: RLIN3LUOR1 -> RLN3LUOR1
20	1112	correction in diagram: RLIN3nLEST -> RLN3nLEST
20	1113	correction in diagram: RLIN3nLEST -> RLN3nLEST
20	1114	corrections in diagram: RLIN3nLEST -> RLN3nLEST, RLIN3nLIDB -> RLN3nLIDB
20	1114	note added
20	1119	explanation added
20	1120	explanation added
20	1122	typo correction in diagram (RLIN3nLBRPx -> RLN3nLBRPx)
20	1123	formula for baud rate removed
20	1124	figure corrected
20	1125	diagram improved
21	1126	note added concerning RIIC clock supply
21	1127	addition to RIIC I/O signals
21	1136	BBSY explanation changed
21	1137	correction: "master transmit mode" -> "master mode"
21	1138	TRS Bit [Setting conditions] changed
21	1138	TRS Bit [Clearing conditions] changed
21	1139	caution added concerning BBSY Flag
21	1140	access type for MTWP bit corrected
21	1140	BCWP explanation corrected
21	1141	BC[2:0] Bits explanation corrected
21	1142	TMOH explanation changed
21	1143	TMOL explanation changed
21	1143	typo correction ([fm] -> [Fm])
21	1145	ACKBT Bit description changed
21	1147	MALE description changed
21	1148	MALE Bit description changed
21	1148	NALE Bit changed
21	1148	NACKE Bit description changed
21	1148	SCLE Bit description changed
21	1152	RIE Bit description changed
21	1153	R/W indicator of bit 2 and 3 corrected
21	1155	DID Flag [Setting condition] changed
21	1157	TMOF Flag [Setting condition] changed
21	1159	RDRF Flag [Setting conditions] changed
21	1160	TDRE Flag [Setting conditions] changed
21	1161	SVA[9:1] description corrected
21	1162	correction: SVA[9:1] Bits are upper address bits
21	1163	typo ciorrection ([fm] -> [Fm])
21	1164	correction in formula: "RIICnBRH+1)/ IICφ" -> "(RIICnBRH+1)/ IICφ"
21	1164	correction in formula: "RIICnBRH+3)" -> "(RIICnBRH+3)"
21	1165	typo correction ("tf: SCL line rising time" -> "tf: SCL line falling time"
21	1165	"Duty cycle: 0% < Duty < 100%" added
21	1170	diagram corrected

Revision 1.00 History, Date Feb 27, 2015 (17/37)

Section	Page	Summary
21	1170	"A#:" explanation added
21	1171	additional remark concerning initial settings
21	1172	correction: "in response to setting of the TRS bit to 1" -> "in response to setting of the TRS and MST bits to 1"
21	1172	typo correction (W -> W#)
21	1176	explanation concerning master reception with 10-bit address changed
21	1177	diagram corrected
21	1178	correction in diagram: "Set RIICnMR3.ACKBT" -> "RIICnMR3.ACKBT = 1"
21	1180	Slave Transmit Operation (1) changed
21	1181	Slave Transmit Operation (6) changed
21	1181	correction in diagram: "[1] ... [6]" -> "[1] ... [7]"
21	1183	Slave Receive Operation (1) changed
21	1183	Slave Receive Operation (5) changed
21	1187	diagram corrected
21	1187	note corrected
21	1189	"Slave-Address Match Detection" explanation corrected
21	1192	correction in diagram: shift of DID rising edge
21	1193	typo correction (Automatically -> Automatic)
21	1193	explanation improved
21	1193	bulleted item added in <Master transmit mode>
21	1193	bulleted item added in <Slave transmit mode>
21	1198	explanation added concerning TRS bit
21	1199	"immediately cancels the slave match condition and" removed
21	1200	"is immediately released from the slave-matched state and" removed
21	1201	explanation changed
21	1201	diagram corrected (8 -> 9)
21	1204	corection in diagram: TMOS settings swappped
21	1205	[Output conditions for using the RIICnCR1.CLO bit] replace with [Additional output conditions for the SCL clock]
21	1206	typo correction (CKS2-0 -> CKS[2:0])
21	1206	typo correction (BC2-0 ->BC[2:0])
21	1206	typo correction (NF1-0 -> NF[1:0])
21	1207	note 1 changed
22	1286	channel index (n -> m) corrected in "CFTML[3:0] Bits"
22	1394	reference to FEINTFMSK register removed
22	1394	caution concerning RS-CAN RAM initialization changed
22	1398	typo correction ("sy nd rom" -> "syndrome")
23	1405	correction: RSCFDnXXX -> RSCANnXXX
23	1406	explanation of index "x" modified
23	1406	typo correction (p = 0 -> q = 0)
23	1407	typo correction (RSCAN0 -> RSCFD0)
23	1407	typo correction (RSCAN0 -> CAN0)
23	1407	typo correction (RSCAN0 -> CAN1)
23	1407	typo correction (RSCAN0 -> CAN0)
23	1407	note added concerning RS-CANFD interrupts and wake-up factors

Revision 1.00 History, Date Feb 27, 2015 (18/37)

Section	Page	Summary
23	1408	typo correction (m = 0 to 3 -> (m = 0 to 2)
23	1408	CANmDREN signals added
23	1409	"Protocol" specification modified
23	1409	"Communication speed" specification modified
23	1409	correction: RSCFDnCmCFG -> RSCANnCmCFG
23	1410	"Transmit history function" specification modified
23	1411	section "Interface Modes" added
23	1411	"Test function" specification modified
23	1412	block diagram modified
23	1412	corrections: RSCFDnCmCFG -> RSCANnCmCFG
23	1412	additional hint concerning clock settings in CAN FD mode
23	1413	register prefix "RSCFD" changed to "RSCANn" in entire "Registers (Classical CAN Mode)" section
23	1413	Global TX Interrupt Status Register 1 (RSCFDnGTINTSTS1) removed
23	1420	"CRCT Bit" explanation concerning communication with other CAN nodes modified
23	1438	typo correction (m = 0 to 5 -> m = 0 to 2)
23	1441	typo correction (RCAN0TMSTSp -> RSCFDnTMSTSp)
23	1441	typo correction (RCAN0TMSTSp -> RSCFDnTMSTSp)
23	1442	"RSCFDnGTINTSTS1 - Global TX Interrupt Status Register 1" removed
23	1450	register name RSCFDnGAFLP0j changed to RSCFDnGAFLP0_j
23	1450	typo correction (GAFLRVM -> GAFLRMV)
23	1452	register name RSCFDnGAFLP1j changed to RSCFDnGAFLP1_j
23	1457	register name RSCFDnRMDF0q changed to RSCFDnRMDF0_q
23	1458	register name RSCFDnRMDF1q changed to RSCFDnRMDF1_q
23	1466	register name RSCFDnRFDF0x changed to RSCFDnRFDF0_x
23	1467	register name RSCFDnRFDF1x changed to RSCFDnRFDF1_x
23	1469	typo correction (16 × n -> 16 × m)
23	1481	register name RSCFDnCFDF0k changed to RSCFDnCFDF0_k
23	1482	register name RSCFDnCFDF1k changed to RSCFDnCFDF1_k
23	1483	correction: 0001 FFFFH -> 03FF FFFFH
23	1490	typo correction (k = 0 to 17 -> k = 0 to 8)
23	1497	"THLEN Bit" explanation modified
23	1500	register name RSCFDnTMDF0p changed to RSCFDnTMDF0_p
23	1501	register name RSCFDnTMDF1p changed to RSCFDnTMDF1_p
23	1502	"TMIEp Bit Assignment" corrected
23	1504	typo correction (y = 0 to 2 -> y = 0, 1)
23	1505	"TMTRSTp Bit Assignment" corrected
23	1506	"TMTARSTp Bit Assignment" corrected
23	1508	"TMTCSTSp Bit Assignment" corrected
23	1510	typo correction (y = 0 to 2 -> y = 0, 1)
23	1510	"TMTASTSp Bit Assignment" corrected
23	1529	Global TX Interrupt Status Register 1 (RSCFDnCFDGTINTSTS1) removed
23	1530	FIFO DMA-related registers removed
23	1533	"RCMC Bit" explanation modified

Revision 1.00 History, Date Feb 27, 2015 (19/37)

Section	Page	Summary
23	1534	typo correction (m = 0 to 5 -> m = 0 to 2)
23	1535	cross reference corrected
23	1535	"NTSEG2[4:0] Bits" explanation modified
23	1537	SOCOIE bit description changed
23	1537	"ROM Bit" explanation modified
23	1538	"CRCT Bit" explanation concerning communication with other CAN nodes modified
23	1541	RSCFDnCFDCmSTS access description corrected
23	1549	explanation concerning RSCFDnCFDCmDCFG register access modified
23	1549	cross reference corrected
23	1549	"DTSEG2[2:0] Bits" explanation modified
23	1550	"TDCO[6:0]" explanation modified
23	1551	correction: Error Counting Method Select -> Error Occurrence Method Select
23	1551	removed "This gateway function cannot receive CAN FD frames with a payload length of more than 8 bytes."
23	1551	removed "Modify this bit only in channel reset mode."
23	1552	Table 1.101 explanation modified
23	1552	CAN mode format names modified
23	1552	"TDCO[6: 0] Bits" explanation modified
23	1552	"ESIC Bit" explanation modified
23	1552	"Error active" description modified
23	1552	"TDCE Bit" explanation modified
23	1553	"EOCCFG[2: 0] Bits" explanation modified
23	1554	"SOCCLR" function modified
23	1554	"EOCCLR" function modified
23	1554	"SOCCLR Bit" explanation modified
23	1554	"EOCCLR Bit" explanation modified
23	1555	RSCFDnCFDCmFDSTS register access description corrected
23	1555	typo correction (TDCV -> TDCVF)
23	1555	correction: communication success counter -> successful occurrence counter
23	1555	"EOC[7:0]" function modified
23	1555	"SOCO" function modified
23	1555	"EOCO" function modified
23	1555	"TDCVF" function modified
23	1555	"TDCR[6"0]" function modified
23	1555	"SOC[7: 0] Bits" explanation modified
23	1556	"EOC[7: 0] Bits" explanation modified
23	1556	"SOCO Flag" explanation modified
23	1556	"EOCO Flag" explanation modified
23	1556	"TDCVF Flag" explanation modified
23	1556	"TDCR[6:0] Flag" explanation modified
23	1558	TSBTCS[2:0] bits added
23	1558	TSSS bit added
23	1559	"TSBTCS[2:0] Bits" explanation added
23	1560	"TSSS Bit" explanation added

Revision 1.00 History, Date Feb 27, 2015 (20/37)

Section	Page	Summary
23	1560	correction: RSCFDnCFDCFCCd -> RSCFDnCFDCFCCk
23	1567	"EEFm Flag" explanation modified
23	1567	typo correction (m = 0 to 5 -> m = 0 to 2)
23	1568	"TS[15:0] Bits" explanation modified
23	1570	typo correction (RCANnCFDTMSTSp -> RSCFDnCFDTMSTSp)
23	1570	typo correction (RCANnCFDTMSTSp -> RSCFDnCFDTMSTSp)
23	1571	RSCFDnCFDGTINTSTS1 register removed
23	1574	RSCFDnCFDGAFLCFG1 register removed
23	1579	register name RSCFDnCFDGAFLP0j changed to RSCFDnCFDGAFLP0_j
23	1579	typo correction (GAFLRVM -> GAFLRMV)
23	1581	register name RSCFDnCFDGAFLP1j changed to RSCFDnCFDGAFLP1_j
23	1582	typo correction (0 to 96 -> 0 to 47)
23	1594	hint concening enabled DMA trasfer request removed
23	1594	correction: RSCFDnCFDRFDF0x -> RSCFDnCFDRFFDSTSx
23	1594	correction: RSCFDnCFDRFDF1x -> RSCFDnCFDRFDFd_x
23	1595	correction: RSCFDnRFIDxH -> RSCFDnCFDRFIDxH
23	1600	correction: RSCFDnCFDRFPTRq -> RSCFDnCFDRFPTRx
23	1601	CFITSS bit added
23	1603	"CFITSS Bit" explanation modified
23	1604	correction: RSCFDnCFCCk -> RSCFDnCFDCFCCk
23	1608	hint concening enabled DMA trasfer request removed
23	1610	CFRTR function modified
23	1611	"CFRTR Bit" explanation modified
23	1616	register name RSCFDnCFDCFDFdk changed to RSCFDnCFDCFDFd_k
23	1616	correction: RSCAN0CFCCk -> RSCFDnCFDCFCCk
23	1616	correction: RSCAN0CFPTRk -> RSCFDnCFDCFPTRk
23	1617	correction: 0001 FFFFH -> 03FF FFFFH
23	1624	typo correction (k = 0 to 17 -> k = 0 to 8)
23	1625	"Details of FIFO DMA-Related Registers" removed
23	1626	description concerning RSCFDnCFDTMCp registers correspondence corrected
23	1630	TMRTR function modified
23	1631	"THLEN Bit: explanation modified
23	1634	valid number range of p corrected
23	1636	register name RSCFDnCFDTMDFbp changed to RSCFDnCFDTMDFb_p
23	1637	typo correction (y = 0 to 2 -> y = 0, 1)
23	1638	TMIEp Bit Assignment corrected
23	1639	typo correction (y = 0 to 2 -> y = 0, 1)
23	1640	TMTRSTSp Bit Assignment corrected
23	1641	TMTARSTSp Bit Assignment corrected
23	1643	TMTCSTSp Bit Assignment corrected
23	1645	typo correction (y = 0 to 2 -> y = 0, 1)
23	1645	TMTASTSp Bit Assignment corrected
23	1649	function of bits 12 to 8 modified
23	1657	"TMTS[15:0] Bits" explanation modified

Revision 1.00 History, Date Feb 27, 2015 (21/37)

Section	Page	Summary
23	1661	numbering corrected
23	1664	correction: " Interrupt Sources and DMA Trigger" -> " Interrupt Sources"
23	1664	typo correction (CMPOF -> CMPOFIE)
23	1666	"CAN Channel Interrupt Block Diagram" figure added
23	1666	"DMA Trigger (Only in CAN FD Mode)" removed
23	1672	Note 3. modified
23	1672	correction: RSCFDnCmCFG -> RSCANnCmCFG
23	1674	RSCFDnCFDCDTCT and RSCFDnCFDCDTSTS registers removed from table
23	1677	typo correction (normal -> nominal)
23	1677	typo correction (normal -> nominal)
23	1678	"Timestamp Function Block Diagram" corrected
23	1679	correction: RSCFDnGCFG -> RSCANnGCFG
23	1681	correction concerning unused interval timer
23	1682	typo correction in figure (normal -> nominal)
23	1682	"Interval Timer Timing Chart" figure modified
23	1684	paragraph concerning transmit buffer modified
23	1684	typo correction ("stored for each message.]" -> "stored for each message.")
23	1684	correction: RSCFDnGCFG -> RSCANnGCFG
23	1684	correction: 402 cycles -> 386 cycles
23	1686	reference to "CRC Error Test" corrected
23	1689	typo correction (staandard -> standard)
23	1690	typo correction (CRCREG[15:0] -> CRCREG[14:0])
23	1690	correction: 3794 cycles -> 1897 cycles
23	1691	explanatory text in figure concerning "Receive rule setting" and "In CAN FD mode: RSCFDnCFDCnCFG register setting" modified
23	1691	correction in figure: RSCFDnCmCFG -> RSCANnCmCFG in classical CAN mode context
23	1692	correction: "by the RSCFDnCmCFG register" -> "by the corresponding register"
23	1692	correction: RSCFDnCmCFG -> RSCANnCmCFG
23	1692	correction: RSCFDnCmCFG -> RSCANnCmCFG
23	1692	correction: "segment value" -> "respective segment value"
23	1692	settings for classical CAN mode in figure modified
23	1693	references to Note 1. added in table
23	1694	correction in figure: normal bit time -> nominal bit time
23	1695	typo corrections in table (nominal -> Nominal)
23	1697	wording of "Buffer Setting" modified
23	1697	buffer settings for classical CAN mode changed
23	1697	change: "Setthe number of buffers ..." -> "Configure the number of buffers ..."
23	1698	number of fixed buffers in figure corrected
23	1699	settings for CAN FD mode added in figure
23	1700	correction: "maximum 3CANm bit time" -> "3 CANm bit time"
23	1701	"Read a message" registers corrected in figure
23	1702	typo correction (k = 0 to 17 -> k = 0 to 8)
23	1703	"Read a message" registers corrected in figure
23	1705	"FIFO Buffer Reading Procedure by DMA Transfer" removed

Revision 1.00 History, Date Feb 27, 2015 (22/37)

Section	Page	Summary
23	1706	"Store messages" registers corrected in figure
23	1708	correction: RSCFDnGCFG -> RSCANnGCFG
23	1709	correction: RSCFDnGCFG -> RSCANnGCFG
23	1710	"Store messages" registers corrected in figure
23	1720	"Detection and Correction of Errors in RS-CANFD RAM" section added
24	1739	CERCR and CEECR registers removed from table
24	1743	typo correction (TAL3[1:0] -> TA[31:0])
24	1747	typo correction ("b16 to b9" -> "b15 to b9")
24	1768	typo corrections in "Value after reset" rows
24	1775	typo correction (TSV[31:16] -> TSV[63:48])
24	1775	typo correction (TSV[15:0] -> TSV[47:32])
24	1777	typo correction ("b31?b0" -> "b31 to b0"
24	1777	correction: "H'0000 0000 to H'0000 FFFF" -> "H'0000 0001 to H'0000 FFFF"
24	1779	typo correction ("b31?b0" -> "b31 to b0"
24	1788	typo correction ("from 00 ... to 10" -> "from B'00 ... to B'10")
24	1788	typo correction ("from 00 ... to 10" -> "from B'00 ... to B'10")
24	1789	typo correction (CULF1 -> CLLF1)
24	1789	typo correction ("from 00 ... to 10" -> "from B'00 ... to B'01")
24	1804	"[Conditions for Changing]" corrected
24	1809	"[Conditions for Changing]" corrected
24	1809	correction: "The bit is set to 0" -> "The bit is set to 1"
24	1809	correction: "The bit is set to 0" -> "The bit is set to 1"
24	1809	correction: "The bit is set to 0" -> "The bit is set to 1"
24	1810	"[Conditions for Changing]" changed
24	1813	typo correction (AVB-DMAV -> AVB-DMAC)
24	1817	typo correction ("register Register" -> "Register")
24	1821	correction: Received PAUSE frames with the Timer value 0 are discarded." removed
24	1821	correction: "1 BP = 1 ns (1000 Mbps)" removed
24	1821	typo correction ("100 bps" -> "100 Mbps")
24	1824	ECSR register initial value corrected
24	1824	PHYI bit removed
24	1824	LCHNG bit removed
24	1825	PHYIM bit removed
24	1825	LINKIM bit removed
24	1826	PIR initial value corrected
24	1827	APFTP[15:0] description corrected
24	1828	MP[15:0] description corrected
24	1831	SPEED description corrected
24	1831	"The default value 0 of this bit must be changed to 1." removed
24	1834	correction: TPOCR -> TROCR
24	1834	correction: TPOC[15:0] -> TROC[15:0]
24	1834	correction: TPOCR -> TROCR
24	1834	correction: TPOC[15:0] -> TROC[15:0]
24	1834	correction: TPOC[15:0] -> TROC[15:0]

Revision 1.00 History, Date Feb 27, 2015 (23/37)

Section	Page	Summary
24	1841	"Lost Carrier Extension Counter Register (CERCR)" and "Carrier Extension Error Counter Register (CEECR)" removed
24	1846	correction concerning operation mode configuration
24	1850	reference to EMAC initialization flow corrected
24	1860	correction in diagram: DBAT.TA + 08h points to Tx queue 1
24	1921	correction in diagram: check operation mode via CSR.OPS[3:0]
26	1950	note added concerning WDTA0 operation during DEEPSTOP mode
26	1954	WDTAnWDTE reset sources corrected
26	1954	correction: "Software trigger start" -> "WDTA is disabled"
26	1956	WDTAnEVAC reset sources corrected
26	1956	correction: "Software trigger start" -> "WDTA is disabled"
26	1957	WDTAnREF reset sources corrected
26	1958	explanation concerning WDTAnMD updating corrected
26	1958	WDTAnMD reset sources corrected
26	1958	WDTAnWIE value after reset corrected
27	1978	typo correction (OSTMnTE -> OSTMnCTL)
29	2010	note 3 and 4 added for some TAUB channels
29	2013	"INTTAUBnIm from master channel" removed
29	2013	"Operating mode" item added
29	2016	Count clock selector inputs corrected in block diagram
29	2016	typo correction ("CK0 to CK2" -> "CK0 to CK3")
29	2017	"INTTAUBnIm from master channel" removed
29	2017	"Effective TAUBTTINm input edge" added as trigger
29	2018	note added
29	2023	reference to operation mode removed
29	2025	typo correction (TAUBnCSR.TAUBnOVF -> TAUBnCSRm.TAUBnOVF)
29	2025	TAUBnSTS[2:0] explanation corrected
29	2031	1-bit access removed
29	2034	TAUBnTOEm description corrected
29	2037	TAUBnTOLm description corrected
29	2043	TAUBnRDMm for C1 method corrected
29	2046	diagram corrected
29	2052	"TAUBnTOE.TAUBnTOEm" removed from bulleted list
29	2055	typo correction in diagram (TAUBnTOL -> TAUBnTOLm)
29	2070	description corrected
29	2074	"Description" changed
29	2079	description changed
29	2087	description changed
29	2090	status of TAUBn during operation corrected
29	2106	typo correction (CMORn:MD0 -> TAUBnCMORm.TAUBnMD0)
29	2106	typo correction (TAUBnCMORn: TAUBnMD0 -> TAUBnCMORm.TAUBnMD0)
29	2106	description corrected
29	2106	formulas removed
29	2107	typo correction (CMORn:MD0 -> TAUBnCMORm.TAUBMD0)

Revision 1.00 History, Date Feb 27, 2015 (24/37)

Section	Page	Summary
29	2107	typo correction (CMORn:MD0 -> TAUBnCMORm.TAUBMD0)
29	2111	typo correction (TAUBnCSC.CLOV -> TAUBnCSC.TAUBnCLOV)
29	2116	typo correction in entire paragraph (CSRn: OVF -> TAUBnCSRm.TAUBnOVF)
29	2129	typo correction in diagram ("NG" relocated)
29	2133	correction in diagram: "NG" included
29	2147	"Conditions" corrected
29	2148	"The following settings ..." paragraph removed
29	2149	diagram corected (relocation of TAUBnRSF.TAUBnRSFm)
29	2152	typo correction (TAUBnCDR.CDRm -> TAUBnCDR.TAUBnCDRm)
29	2157	typo corrections in note (TAUBnRDS -> TAUBnRDSm, TAUBnCDR -> TAUBnCDRm)
29	2161	diagram corrected (INTTAUBnIm shifted)
29	2165	reference to table corrected
29	2165	reference to table corrected
29	2169	addition concerning clock selection of amster and slave
29	2175	additions to (3)
29	2177	diagram corrected
29	2179	reference to table added
29	2179	reference to table corrected
29	2192	typo correction in diagram (INTTAUBnTm -> INTTAUBnIm) and Slave-2 TAUBnCNTm shifted
29	2192	typo correction ("the counter of the slave channels" -> "the counter of the slave channel 1"
29	2193	start value of Slave-3 TAUBnCNTm corrected in diagram
29	2195	typo corrections in diagram (TSm -> TAUBnTSm, TEm -> TAUBnTEm)
29	2196	reference to table corrected
29	2199	TAUBTTOUTm timing adjusted in diagram
29	2205	explanation corrected
29	2206	explanation corrected
29	2207	addition concerning usage of slave channel 1
29	2208	addition concerning usage of slave channel 1
29	2208	addition concerning INTTAUBnIM generation
29	2209	"Equations" corrected
29	2211	"Slave channel 3" settings corrected
29	2211	correction in diagram: Slave 3 TAUBnCDRm value f -> g
29	2215	typo correction (TDLm -> TAUBnTDL.TAUBnTDLm)
29	2218	table title corrected
29	2222	explanation corrected
29	2222	explanation corrected
29	2223	explanation corrected
29	2225	typo correction ("the set signal" -> "the reset signal")
29	2227	correction: "the set signal has priority" -> "the reset signal has priority"
29	2227	typo corrections in diagram (TAUBnCDRn -> TAUBnCDRm)
30	2235	"Valid edge of the TAUJTTINm input signal" removed
30	2235	"Operating mode" added
30	2242	operation reference for initial value removed

Revision 1.00 History, Date Feb 27, 2015 (25/37)

Section	Page	Summary
30	2245	TAUJnCOS[1:0] description changed
30	2253	restriction concerning Only restriction concerning TAUJnTOm removed
30	2255	TAUJnTOCm description corrected
30	2257	cross reference added
30	2263	"Description (6)" changed
30	2266	removed "TAUJnTOE.TAUJnTOEm" from bulleted list
30	2270	typo correction in diagram (TAUJn.CNTm -> TAUJnCNT.TAUJnCNTm)
30	2270	typo correction in diagram (TAUJn.CNTm -> TAUJnCNT.TAUJnCNTm)
30	2275	typo correction ("0, 1" -> "1, 0")
30	2276	cross reference to other section corrected
30	2278	typo correction (TAUJTTOUTm -> TAUJnTTOUTm)
30	2281	diagram correction: TAYJTTOUTm adjusted
30	2283	cross reference to other section corrected
30	2285	"The timing diagrams ..." paragraph changed
30	2287	"Conditions" changed
30	2298	typo correction (TAUJnCSC.CLOV -> TAUJnCSCm.TAUJnCLOV
30	2298	Status of TAUJn during operation changed
30	2303	note added
31	2374	typo correction (RTCA0CEST -> RTCAnCTL0.RTCAnCEST)
31	2375	typo correction (RTCA0WST -> RTCAnCTL2.RTCAnWST)
31	2377	typo correction (RTCA0CEST -> RTCAnCTL0.RTCAnCEST)
32	2403	typo correction (PWGAnSTR -> PWSAnSTR)
32	2403	typo correction (PWSAQUEj -> PWSAnQUEj)
32	2404	typo correction (PWGAnSTC -> PSGAnSTC)
32	2406	typo correction (PWSA0QUEn -> PWSA0QUEj)
33	2422	typo correction "he duty cycle" -> "The duty cycle")
33	2426	PWM base frequency calculation formula corrected
33	2432	M values in example corrected
33	2432	tone duration values corrected
33	2433	additional information concerning M
33	2434	typo correction (ALD -> ADI)
33	2435	reference to "M" calculation equations added
33	2436	note corrected
33	2450	references to other sections corrected
33	2452	typo correction (oncrement -> increment)
33	2453	correction: "ALD, ADI, and ADC modes" -> "ALD and ADI modes"
34	2459	note 2 added
34	2467	output control table corrected
34	2475	reference to other section corrected
34	2475	reference to other section corrected
35	2482	reset sources for SSIFn modules added
35	2482	caution added
35	2484	sampling frequency change interrupt removed
35	2486	correction: SSIFCCR_0, SSIFCMR_0, SSIFCSR_0 registers removed

Revision 1.00 History, Date Feb 27, 2015 (26/37)

Section	Page	Summary
35	2486	correction: SSIFCCR_1, SSIFCMR_1, SSIFCSR_1 registers removed
35	2486	correction: "and no need "_0" "_1" at the end. (i.e. SSIF0SSICR)" -> "and also needs "_0" "_1" at the end (i.e. SSIF0SSICR_0, SSIF0SSICR_1)."
35	2486	modification: SSIF0SSICR_1 -> SSIF1SSICR_1
35	2499	SSIFCCR, SSIFCMR, SSIFCSR register descriptions rmoved
35	2500	correction: 0: Continues AUDIO clock, 1: Stops AUDIO clock
35	2500	correction: 0: Continues SSISCK clock, 1: Stops SSISCK clock
35	2501	references to note 1 added in table
35	2501	correction: "using master transceiver mode" -> "using transceiver mode"
36	2519	<tbc> removed from Interrupt Controller INTLCBI0EMPTY
36	2519	<tbc> removed from Interrupt Controller INTLCBI0HALF
36	2519	<tbc> removed from Interrupt Controller INTLCBI0FULL
36	2519	<tbc> removed from Interrupt Controller INTLCBI0QTR
36	2519	<tbc> removed from Interrupt Controller INTLCBI03QTR
36	2557	initial value corrected
37	2566	caution added
37	2571	typo correction ("01 for channel 1" -> "10 for channel 1")
37	2573	typo correction ("01 for channel 1" -> "10 for channel 1")
37	2577	D1M1(H) pixel clock generator corrected and separated from others
37	2579	typo correction in diagram (CDECTL.VI1CTL -> VCDECTL.VI1CTL)
37	2581	"List of video channels clock generators registers" table added
37	2594	correction: CKSC_IVIDEO0S_CTL -> CKSC_IDOTCLK0S_CTL
37	2596	correction: CKSC_IVIDEO0S_CTL -> CKSC_IDOTCLK1S_CTL
37	2606	correction: CKSC_IVDCE0VIS_CTL also for D1M1(H)
37	2607	correction: CKSC_IVDCE0VIS_ACT also for D1M1(H)
37	2608	typo correction: PLL0PIX -> PLL0PIXCLK
37	2610	"All others: Setting prohibited" removed
37	2615	typo corrections (NP_EXT_SYNC_CN -> INP_EXT_SYNC_CNT)
37	2616	correction "VDCECTL.PXSL = 0" -> "VDCECTL.PXSL = 1"
37	2617	typo corrections (VDCE0_VI_ -> VDCE1_VI_)
37	2617	typo corrections (INP_EXT_SYNC_CN -> INP_EXT_SYNC_CNT)
37	2622	typo correction (MIPI_VI_D[23:0] -> MIPI0_VI_D[23:0])
37	2622	typo correction (MIPI_VI_D[23:0] -> MIPI0_VI_D[23:0])
37	2624	VTIM bit added
37	2624	typo correction ("TCLK-SETTLE are dined in MIPInSOT_COUNT.CLK_COUNT[7:0]." -> "TCLK-SETTLE is defined in MIPInSOT_COUNT.CLK_COUNT[7:0]."
37	2624	typo correction ("TCLK-SETTLE are dined in MIPInEOT_COUNT.EOT_COUNT[7:0]." -> "TEOT is defined in MIPInEOT_COUNT.EOT_COUNT[7:0].")
37	2625	RSTN_DATA_ON description corrected
37	2631	typo correction ("RX_STOP_STATE_C0 = RX_STOP_STATE_C1 = RX_STOP_STATE_C = 0" -> RX_STOP_STATE_0 = RX_STOP_STATE_1 = RX_STOP_STATE_C = 0"
37	2631	MIPInRX_STATE_PHY initial value corrected
37	2631	MIPInRX_STATE register description corrected
37	2635	values of bits 15 to 8 added and "R/W" corrected to "R"
37	2638	MIPInRDT_ERR_ON initial value corrected

Revision 1.00 History, Date Feb 27, 2015 (27/37)

Section	Page	Summary
37	2638	RDT_ERR_ ON3 description corrected
37	2638	RDT_ERR_ ON2 description corrected
37	2638	RDT_ERR_ ON1 description corrected
37	2638	RDT_ERR_ ON0 description corrected
37	2639	meaning of MIPInINTSTATUS bits corrected
37	2642	typo correction ("enable enable" -> "enable")
37	2646	typo correction (d-assert -> de-assert)
37	2651	typo correction ("4 bpp (RGB888)" -> "24 bpp (RGB888)")
37	2651	"Stop sequence" corrected
37	2652	DEMODE0.DE_4HS_EN and MIPInMODE.VTIM effect added in MIPI setup
37	2654	"VSYNC Timing from MIPI to VDCE" added
37	2658	new section "Video data and clock output settings" added, including further video output signal tables
37	2666	note added concerning accesses after register modifications
37	2670	typo correction ("the the" -> the)
37	2671	note added concerning accesses after register modifications
37	2672	"Initial value" and "R/W" rows corrected
38	2674	PCLK clock source for VDCEn corrected
38	2679	"Operating frequency" item removed from table, specification included in the data sheet
38	2680	typo correction (YCbCr422 -> YCC422)
38	2680	typo correction (YCbCr444 -> YCC444)
38	2680	note added concerning alpha blending and chroma-key feature for GR0
38	2680	addition: "ITU-R BT.656" -> "ITU-R BT.656 and BT.601"
38	2681	"distortion correction" removed from block diagram
38	2684	"(2) Notes" added
38	2685	Recorded external video input removed
38	2688	correction: "BT601" -> "BT601 (extended)"
38	2701	section "SVA/EAV Code in BT656 Progressive Format" renamed in "BT656 Progressive Format" and completely changed
38	2713	"Video input by data enable" added
38	2716	data enable input register tables added
38	2718	INP_HS_EDGE description corrected
38	2721	"Image Quality Adjustment Block Matrix YG Adjustment Register 1 (IMGCNT_MTX_YG_ADJ1)" displaced
38	2726	DEMODE0 register added
38	2727	DEMODE1 register added
38	2728	typo correction (IMP_UPDATE -> INP_UPDATE)
38	2729	typo correction (INT_UPDATE -> INP_UPDATE)
38	2731	cross reference corrected
38	2735	SC_RES_QVLOCK bit description improved
38	2738	correction: (VSYNC + V backporch lines) -> (VSYNC + (V backporch -1) lines)
38	2740	correction in diagram: "SC_RES_VS" -> "SC_RES_VS + 1"
38	2752	cross reference corrected
38	2754	"After making the setting to enable writing, writing starts from the second frame." added
38	2756	correction in diagram: "SC_RES_VS" -> "SC_RES_VS + 1"

Revision 1.00 History, Date Feb 27, 2015 (28/37)

Section	Page	Summary
38	2757	SC_RES_FLM_CNT[9:0] description corrected
38	2757	SC_RES_FLM_CNT_B[9:0] description corrected
38	2759	cross reference corrected
38	2759	GR_FLD_NXT description corrected
38	2760	"After making the setting to enable writing, writing starts from the second frame." added
38	2760	cross reference corrected
38	2762	SC0_SCL0_MON0 initial value corrected to 16 bit
38	2762	SC0_SCL0_INT initial value corrected to 16 bit and R/W access specified
38	2764	VDCE0 SC1_SCL0_DS1 and SC1_SCL0_DS6 registers added in table
38	2766	SC0_SCL0_MON0 address corrected and initial value changed to 16 bit
38	2766	SC0_SCL0_INT initial value changed to 16 bit and R/W specified
38	2768	VDCE1 SC1_SCL0_DS1 and SC1_SCL0_DS6 registers added in table
38	2777	SC0_RES_ QVLOCK bit description improved
38	2777	SC0_RES_ QVLACK bit description improved
38	2778	bit access modes corrected
38	2779	correction: (VSYNC + V backporch lines) -> (VSYNC + (V backporch - 1) lines)
38	2783	addition to note
38	2788	note added concerning initial value of SC0_SCL0_US5
38	2793	note added
38	2797	"After making the setting to enable writing, writing starts from the second frame." added
38	2798	SC0_RES_FLM_CNT[9:0] description corrected
38	2800	SC0_RES_FLM_CNT_B[9:0] description corrected
38	2802	correction: SC0_SCL0_FRC3 -> SC0_SCL1_WR1
38	2802	rows of reserved bits 7 to 1 merged
38	2802	GR0_IMR_FLM_INV reference removed
38	2803	GR0_FLD_NXT description corrected
38	2806	typo correction (YCC422 -> YCbCr422)
38	2806	typo correction (GR0_CNV444 -> GR0_CNV444_MD)
38	2807	note added concerning alpha blending and chroma-key feature for GR0
38	2808	note added concerning GR0_GRC_ VW[10:0] setting
38	2809	note added concerning GR0_GRC_HW[10:0] setting
38	2810	note added concerning alpha blending and chroma-key feature for GR0
38	2811	note added concerning alpha blending and chroma-key feature for GR0
38	2821	SC1_RES_ QVLOCK bit description improved
38	2821	SC1_RES_ QVLACK bit description improved
38	2822	"Scaling-Down Control Register (SC1_SCL0_DS1)" added
38	2823	"Vertical Scaling Register (SC1_SCL0_DS6)" added
38	2829	note corrected
38	2832	reference to GR1_IMR_FLM_INV removed
38	2833	GR1_FLD_NXT description corrected
38	2836	typo correction (YCC422 -> YCbCr422)
38	2838	note added concerning GR1_GRC_ VW[10:0] setting
38	2839	note added concerning GR1_GRC_ HW[10:0] setting
38	2842	GR1_CK_A[7:0] description corrected

Revision 1.00 History, Date Feb 27, 2015 (29/37)

Section	Page	Summary
38	2843	GR1_A0[7:0] description corrected
38	2844	GR1_A1[7:0] description corrected
38	2848	typo correction (GR_FLM_LNUM[9:0] -> GR_FLM_LNUM[10:0]
38	2851	block diagram corrected
38	2853	typo correction (ADJ_MTX_CBB_ADJ9 -> ADJ_MTX_CBB_ADJ0)
38	2867	note 3 added
38	2873	typo correction ("GR_BASE[31:3] = t (H_OFF ÷ 2)" -> "GR_BASE[31:3] = int (H_OFF ÷ 2)"
38	2874	typo correction (GR1_FLM6 -> GR_FLM6)
38	2874	typo correction (GR1_CNV444_MD -> GR_CNV444_MD)
38	2881	note added concerning alpha blending and chroma-key feature for GR0
38	2884	note added concerning alpha blending and chroma-key feature for GR0
38	2885	note added concerning alpha blending and chroma-key feature for GR0
38	2887	typo correction (GR_ACALC_MD -> GR_ARC_MUL)
38	2887	typo correction (GR_ACALC_MD -> GR_ARC_MUL)
38	2887	(premultiplication) removed
38	2896	"GR2_BST_MD is updated when GR2_IBUS_VEN and GR2_P_VEN in GR2_UPDATE are 1." removed
38	2904	note added concerning GR2_GRC_ VW[10:0] setting
38	2905	note added concerning GR2_GRC_ HW[10:0] setting
38	2916	"GR3_BST_MD is updated when GR3_IBUS_VEN and GR3_P_VEN in GR3_UPDATE are 1." removed
38	2924	note added concerning GR3_GRC_ VW[10:0] setting
38	2925	note added concerning GR3_GRC_ HW[10:0] setting
38	2937	note added concerning GR_VIN_ GRC_VW [10:0] setting
38	2938	note added concerning GR_VIN_ GRC_HW [10:0] setting
38	2945	Output Image Generator function description corrected
38	2948	OIR_RES_VS_SEL description correted
38	2949	correction: "an external input Vsync signal" -> "a Vsync signal input from the image synthesizer"
38	2950	descriptions in diagram improved
38	2950	figure "Vsync Signal Phases (One Frame-Buffer Plane Used)" removed
38	2951	correction: (VSYNC + V backporch lines) -> (VSYNC + (V backporch - 1) lines)
38	2952	"Screen Synthesis" removed
38	2953	section title changed
38	2953	"Accordingly, set the OIR_RES_IBUS_SYNC_SEL bit to 1 to select graphics display." removed
38	2953	table title and table content changed
38	2953	cross reference corrected
38	2954	OIR_SCL0_OVR1 register removed from table
38	2956	note added concerning OIR_SCL0_UPDATE availability in D1L2 devices
38	2958	OIR_RES_EN description corrected
38	2958	OIR_RES_VS_SEL description corrected
38	2962	correction: (VSYNC + V backporch lines) -> (VSYNC + (V backporch - 1) lines)
38	2964	correction: "written to the frame buffer" -> "written to VOWE"
38	2964	explanations concerning OIR_SCL1_WR1.GR_OIR_FLM_LOOP removed

Revision 1.00 History, Date Feb 27, 2015 (30/37)

Section	Page	Summary
38	2968	OIR_SCL0_OVR1 register removed
38	2969	note added concerning OIR_SCL1_UPDATE availability in D1L2 devices
38	2970	access type for bit 1 corrected
38	2970	typo correction (R/W -> R)
38	2972	note added concerning GR_OIR_UPDATE) availability in D1L2 devices
38	2974	note added concerning update of bit[9:8] and bit0
38	2977	correction: GR_OIR_HW[9:0] -> GR_OIR_HW[10:0]
38	2979	note added concerning GR_OIR_GRC_DISP_ON availability in D1L2 devices
38	2979	GR_OIR_DISP_SEL[1:0] description corrected
38	2980	note added concerning GR_OIR_ GRC_VW [10:0] setting
38	2981	note added concerning GR_OIR_ GRC_HW [10:0] setting
38	2984	access type of bit 16 corrected
38	2984	GR_OIR_LINE[10:0] bit positions corrected
38	2990	"GAM_G_TH_01 to GAM_G_TH_31 [7:0]" description corrected
38	2991	"GAM_B_TH_01 to GAM_B_TH_31 [7:0]" description corrected
38	2991	"GAM_R_TH_01 to GAM_R_TH_31 [7:0]" description corrected
38	3019	bit descriptions corrected
38	3019	typo correction ("next area d" -> "next area")
38	3020	bit descriptions corrected
38	3020	typo correction ("next area d" -> "next area")
38	3021	bit descriptions corrected
38	3022	bit descriptions corrected
38	3023	bit descriptions corrected
38	3024	bit descriptions corrected
38	3025	bit descriptions corrected
38	3026	bit descriptions corrected
38	3030	bit descriptions corrected
38	3031	bit descriptions corrected
38	3032	bit descriptions corrected
38	3033	bit descriptions corrected
38	3034	bit descriptions corrected
38	3035	bit descriptions corrected
38	3036	bit descriptions corrected
38	3037	bit descriptions corrected
38	3041	bit descriptions corrected
38	3042	bit descriptions corrected
38	3043	bit descriptions corrected
38	3044	bit descriptions corrected
38	3045	bit descriptions corrected
38	3046	bit descriptions corrected
38	3047	bit descriptions corrected
38	3048	bit descriptions corrected
38	3056	typo correction (xThough -> Though)
38	3071	reserved interrupts removed from table

Revision 1.00 History, Date Feb 27, 2015 (31/37)

Section	Page	Summary
38	3072	name change in entire section; INT0 -> S0_VI_VSYNC
38	3072	name change in entire section; INT1 -> S0_LO_VSYNC
38	3072	name change in entire section; INT2 -> S0_VSYNCERR
38	3072	name change in entire section; INT3 -> GR3_VLINE
38	3072	name change in entire section; INT4 -> S0_VFIELD
38	3072	name change in entire section; INT5 -> IV1_VBUFERR
38	3072	name change in entire section; INT6 -> IV3_VBUFERR
38	3072	name change in entire section; INT7 -> IV5_VBUFERR
38	3072	name change in entire section; INT8 -> IV6_VBUFERR
38	3072	name change in entire section; INT9 -> S0_WLINE
38	3072	name change in entire section; INT11 -> S1_LO_VSYNC
38	3072	name change in entire section; INT12 -> S1_VSYNCERR
38	3073	name change in entire section; INT15 -> IV4_VBUFERR
38	3073	name change in entire section; INT17 -> OIR_VI_VSYNC
38	3073	name change in entire section; INT18 -> OIR_LO_VSYNC
38	3073	name change in entire section; INT19 -> OIR_VLINE
38	3073	name change in entire section; INT22 -> IV8_VBUFERR
38	3074	INT_OUT23_ON bit removed from table
38	3077	correction: INT_STA7 is bit 28, INT-STA3 is bit 12
38	3079	access modes of bits 8 and 20 corrected
38	3080	access modes of bit 16 corrected
38	3081	INT_OUT5 bit added
38	3083	access mode of bits 8 and 20 corrected
38	3083	INT_OUT14_ON bit removed
38	3084	INT_OUT16_ON and INT_OUT23_ON bits removed
38	3084	access mode of bit 16 corrected
38	3085	initial values of bits 12, 8, 4 and 0 in table corrected
39	3086	PCLK clock source corrected
40	3099	flow of Video Output Monitor error flags clearing corrected
40	3102	note completed with "If the Activity Monitor for only channel 1 is to be enabled, channel 0 also is monitored at first because channel 0 is selected by default."
40	3107	VOCAnE18 description corrected
40	3107	note added concerning VOCAnE17 setting
40	3108	VOCAnCL18 description corrected
40	3108	typo correction (VOCSnCL18 -> VOCAnCL18)
40	3108	typo correction (VOCSnCL17 -> VOCAnCL17)
40	3109	typo correction (VOCSnCL16 -> VOCAnCL16)
40	3109	typo correction (VOCSnCLm -> VOCAnCLm)
40	3111	note added concerning VOCAnCH setting
40	3112	note added concerning availability of VOCAnTIME1
40	3112	typo correction ("Upper detection" -> "Lower detection")
40	3113	note added concerning availability of VOCAnOFFSj
40	3113	initial value of bit 16 corrected
40	3113	caution added

Revision 1.00 History, Date Feb 27, 2015 (32/37)

Section	Page	Summary
40	3114	note added concerning availability of VOCAnDISPj
40	3114	initial value of bit 16 corrected
40	3114	caution added
40	3115	note added concerning VOCAnACTj setting
40	3115	typo correction (VOCSnVOUTj -> VOCAnVOUTj)
40	3125	VOCAnEXPDk initial value corrected
41	3134	typo correction ("Pixel Format Data Width" removed)
41	3134	caution added concerning change of CMPSELP[3:0] default value
41	3139	typo correction (aRGB888 -> aRGB8888)
42	3146	GPU2D operation clock connections corrected
43	3164	typo correction in diagram (SPEanRLSL.SPEAnRBUSSEL -> SPEAnRLSL.SPEAnRBUSSEL)
43	3166	"that marks the virtual frame start address." added
43	3166	caution added
43	3170	correction in diagram: Red/Green12 -> Red/Green11 for 18 bpp colour format
43	3173	typo correction in diagram ("Image Synthesizer layer 10" -> "Image Synthesizer layer 11")
43	3174	typo correction (SPEnSkENm -> SPEAnSkENm)
43	3174	typo correction (SPEnSkDSm -> SPEAnSkDSm)
43	3175	typo correction (SPEAnSkLYmH -> SPEAnSkLYHm)
43	3176	typo correction (SPEnSkENm -> SPEAnSkENm)
43	3182	SPEAnRVDCi[3:0] description corrected
44	3193	Register access clock PCLK clock source corrected
44	3200	typo correction ("7" -> "7 to 3")
44	3207	INT[7:5] bit descriptions corrected
44	3209	JCRST initial value changed
44	3211	correction: "Byte/Word Swap" -> "Byte/Word/Longword Swap"
44	3215	correction: "in 8-byte units" -> "in 8-line units"
44	3225	DINLEN bit description corrected
44	3226	typo correction ("is only valid when" -> "is valid only when")
44	3232	typo correction ("H'0000 0000" -> "H'00")
44	3234	correction in "Input Pixel Format" section: JOUTSWAP -> DINSWAP
44	3235	case "DINSWAP bits = 011" added
44	3238	addition to "Error Handling"
44	3239	"Error Handling Flow" changed
44	3240	correction: "SOF0, DQT, and DHT" -> "SOF0, DQT, or DHT"
44	3240	correction: "after a bus reset." -> "after a reset."
44	3241	typo correction ("this error is not detected" -> "this error is not detected.")
44	3246	typo correction ("has been read" -> "has been read.")
44	3246	typo correction ("has been read" -> "has been read.")
45	3249	note added concerning external H/W trigger sources
45	3257	caution concerning ADCEnVCRj setting changed
45	3260	typos corrected (readwritten -> read/written)
45	3261	typo correction ("32-bit read-only register" -> "32-bit/16-bit read-only register"
45	3261	ADCEnDRj 16-bit access registers added

Revision 1.00 History, Date Feb 27, 2015 (33/37)

Section	Page	Summary
45	3261	ADCEnDRj 16-bit access addresses added
45	3263	correction: "32-bit read-only register" -> "32-bit/16-bit read-only register"
45	3263	ADCEnPWDTSNDRj 16-bit access registers added
45	3263	ADCEnPWDTSNDRj 16-bit access addresses added
45	3264	WFLG description changed
45	3267	ADCEnADCR Access corrected
45	3272	correction: ULMTB[15:0] -> ULMTB[11:0]
45	3272	correction: LLMTB[15:0] -> LLMTB[11:0]
45	3272	cautions added and changed
45	3273	ULEC description corrected
45	3273	OWEC description corrected
45	3274	ULSG[1:0] description improved
45	3277	typo correction (Levle -> Level)
45	3277	correction: ADCEnSGPRCR register can be read/written
45	3277	initial value in bit map corrected
45	3279	note added
45	3281	typo correction (PWM-Diag -> Temperature Sensor)
45	3290	caution added
45	3301	left/right cropped diagram corrected
45	3309	diagram changed
45	3310	diagram changed
45	3320	resistor name changed in figure
45	3321	"Diagnostic procedure 1." corrected
46	3324	motor control channel indices m for D1M2 devices corrected
46	3375	correction: ISMnGSTC.ISMnGIPCLAD -> ISMnGSTC.ISMnGIPCLADm
46	3375	"(all measured voltages were higher than the reference voltage)" changed to section reference
46	3376	correction: Channel management is not switched off by software reset
46	3377	correction: "writing 0 to ISMnGSTC.ISMnGIPCLREm." -> "writing 1 to ISMnGSTC.ISMnGIPCLREm."
46	3383	ISMnGIEDO description corrected
46	3383	ISMnGIEREm description corrected
46	3393	ISMnCIVEm description corrected
46	3401	ISMnGGRV2[3:0] description corrected
46	3402	ISMnGGRV1[3:0] description corrected
46	3405	correction: ISMnCZCFG.ISMnCZBTm[15:0] -> ISMnZPDCMP.ISMnCZBTm[15:0]
46	3411	correction: ISMSVDIS is bit 7 of ISMnEMU register
46	3412	reference to FEINTFMSK register removed
46	3416	typo correction ("sy nd rom" -> "syndrome")
47	3422	remark concerning ASIL B rephrased
47	3422	"Memory -DMA-CRC check" explanation improved
47	3422	"CANFD Interface (RS-CANFD)" added
47	3423	cross reference to VOCA corrected
48	3426	"Internal Peripheral Guards errors" removed from ECM errors list
48	3427	error numbers 42 to 47 added

Revision 1.00 History, Date Feb 27, 2015 (34/37)

Section	Page	Summary
48	3428	"Error Control Module Block Diagram" added
48	3428	delay timer clock supply corrected in figure and note added
48	3428	"Operations for Error Output" description improved
48	3430	section reference instead of description about interrupted unlock sequence
48	3430	description about delay timer clock suplly added
48	3432	correction: ECMMPCMD0 is "Not protected"
48	3432	correction: ECMCPCMD0 is "Not protected"
48	3434	correction: ECMmSSE037 -> ECMmSSE105
48	3434	correction: ECMEMK037 - ECMEMK105
48	3434	correction: "ECMIE037 bit of the ECMNMICFG1 register" -> "ECMNMIE105 bit of the ECMNMICFG1 and the ECMMIE105 bit of the ECMMICFG1 register"
48	3434	correction: ECMIRE037 -> ECMIRE105
48	3434	correction: ECMCLSSE037 -> ECMCLSSE105
48	3434	correction: ECMEMK037 - ECMEMK105
48	3434	correction: "ECMIE037 bit of the ECMMICFG1 register" -> "ECMMIE105 bit of the ECMMICFG1 and the ECMNMIE105 bit of the ECMNMICFG1 register"
48	3434	correction: ECMIRE037 -> ECMIRE105
48	3435	correction: ECMEMK037 - ECMEMK105
48	3435	correction: "ECMNMIE037 bit of the ECMMICFG1 register" -> "ECMMIE105 bit of the ECMMICFG1 and the ECMNMIE105 bit of the ECMNMICFG1 register"
48	3435	correction: ECMIRE037 -> ECMIRE105
48	3435	correction: ECMCLSSE037 -> ECMCLSSE105
48	3436	typo correction (ECMMSSExxx -> ECMmSSExxx)
48	3436	typo corrections (ECMMSSExxx -> ECMmSSExxx)
48	3437	typo correction (ECMMSSExxx -> ECMmSSExxx)
48	3437	typo correction (ECMMSSExxx -> ECMmSSExxx)
49	3462	typo correction in diagram (DCRAnCTL.DCRnPOL -> DCRAnCTL.DCRAnPOL)
49	3467	typo corrections ("DCRAnISZ = 10" -> "DCRAnISZ = 10B")
49	3467	typo correction (DCRAnCTL.DCRAnISZn -> DCRAnCTL.DCRAnISZ[1:0])
49	3468	typo correction (DCRA0COUT -> DCRAnCOUT)
51	3472	ADCE added to list of modules with optional peripheral break
51	3472	PWM-Diag added to list of modules with optional peripheral break
51	3472	"Modules that stop when EPC.SVSTOP = 1" table added
51	3473	"(14) Trace function" removed
51	3474	correction: "and inhibits interference from a debugger by register operation to special sequence access (SVACCESS)" removed
51	3474	SVACCESS bit removed
51	3476	note 3 added
52	3487	signal level notations in figure changed
52	3489	rephrased
52	3492	product name storage register's initial values for products with RS-CANFD added
52	3499	initial state of address parity error check removed
52	3514	prohibited item added during programming and erasure
53	3516	correction: "8-bit wide ECC RAM" -> "7-bit wide ECC RAM"
53	3516	correction: "The 8-bit ECC value is calculated for each 128-bit transfer." -> "The 7-bit ECC value is calculated for each 32-bit transfer."

Revision 1.00 History, Date Feb 27, 2015 (35/37)

Section	Page	Summary
53	3516	"Local RAM clock diagram" corrected
53	3518	LRAM test registers added to table
53	3520	LRERRINT register initial value corrected
53	3520	handling of LRERRINT register reserved bits changed
53	3521	correction in LRSTCLR description table: "Error Overflow Flag Clear" -> "Error Status / Error Overflow Flag Clear"
53	3523	"(during RMW processing for Local RAM)" removed
53	3524	correction: EADR[12:0] -> EADR[14:0]
53	3525	"LRTSTCTL — Local RAM Test Control Register" added
53	3526	"LRTSTCTL — Local RAM Test Control Register" added
53	3527	typo correction ("n = 0, 1" -> "n = 0")
53	3527	typo corrections in table (Local RAM -> Retention RAM)
53	3528	"Retention RAM initialization" added
53	3530	typo correction ("m = 0 to 6" -> "m = 0 to 7")
53	3530	typo correction (ECSEDEF[6:0] -> ECSEDF[7:0])
53	3530	correction: INTVRMECCDED -> INTECCDEDRRAM
53	3530	typo correction ("m = 0 to 6" -> "m = 0 to 7")
53	3530	typo correction (ECSEDEF[6:0] -> ECSEDEF[7:0])
53	3530	assertion of the overflow interrupt INTECCOVF removed
53	3531	"ECC modes overview" table corrected
53	3532	RRAM ECC test registers added to table
53	3532	note added concerning accesses after register modifications
53	3533	RRAMECCnCTL initial value corrected
53	3533	correction: ECC error judgement disabled by RRAMECCnCTL.ECERVF = 0
53	3533	correction: ECC error judgement disabled by RRAMECCnCTL.ECERVF = 0
53	3534	correction: ECER2C = 1 clears ECER2F/ECDEDFm/ECOVFF
53	3534	ECERVF description corrected
53	3534	EC2EDIC description corrected
53	3534	EC1EDIC description corrected
53	3534	correction: ECC error judgement disabled by RRAMECCnCTL.ECERVF = 0
53	3534	correction: ECC error judgement disabled by RRAMECCnCTL.ECERVF = 0
53	3535	correction: ECC error judgement disabled by RRAMECCnCTL.ECERVF = 0
53	3536	typo corrrections ("Initial value" and "R/W" rows corrected)
53	3536	correction: "read address = 8000 0000H + ECEAD[31:0] * 8" -> "read address = 3FCE 4000H + ECEAD[31:0] * 8
53	3537	"RRAMECCTMC – ECC test mode control register" added
53	3539	"RRAMECCTED1 – ECC encode/decode input/output substitution register 1" register added
53	3539	correction: "handles lower 32 bit data " -> "handles upper 32 bit data"
53	3539	correction: ECEDB[31:0] -> ECEDB[63:32]
53	3539	correction: RAM data [31:0] -> RAM data [63:32]
53	3540	"RRAMECCTED2 – ECC encode/decode input/output substitution register 2" register added
53	3540	correction: "handles upper 32 bit data" -> "handles lower32 bit"
53	3540	correction: ECEDB[63:32] -> ECEDB[31:0]

Revision 1.00 History, Date Feb 27, 2015 (36/37)

Section	Page	Summary
53	3541	"RRAMECCTRC – ECC test redundant bit data control register" added
53	3541	SYND[7:0] bits removed
54	3548	typo correction (WTOVFL[7:0] -> WTOVFL[6:0])
54	3548	typo correction (WTOVFL[7:0] -> WTOVFL[6:0])
54	3549	typo correction (RTOVFL[7:0] -> RTOVFL[6:0])
54	3549	typo correction (RTOVFL[7:0] -> RTOVFL[6:0])
54	3550	correction: "<VRAMn_base> + 40H + n x 10H" -> "<VRAMn_base> + 40H"
54	3550	correction: "<VRAMn_base> + 44H + n x 10H" -> "<VRAMn_base> + 44H"
54	3551	typo correction ("<VRAM_base> +40H + 10H"-> "<VRAMn_base> +40H")
54	3552	typo correction ("<VRAM_base> +44H + 10H"-> "<VRAMn_base> +44H")
54	3552	initial values of bit 1 and 0 corrected
54	3556	remove all SDRAM references from diagram
54	3558	exchanged VRAM0 and VRAM1
54	3558	note added concering DDR2-SDRAM > 128 Mbyte
54	3560	"ECC process during data write with less than 64 bit" removed and note added instead
54	3560	"VRAM initialization" added
54	3561	typo correction ("m = 0 to 6" -> "m = 0 to 7")
54	3561	typo correction (ECSEDEF[6:0] -> ECSEDF[7:0])
54	3561	correction: INTVRMECCDED -> INTECCDEDVRAM
54	3561	typo correction ("m = 0 to 6" -> "m = 0 to 7")
54	3561	typo correction (ECSEDEF[6:0] -> ECSEDEF[7:0])
54	3561	correction: overflow interrupt INTECCOVF assertion removed
54	3561	typo corrections (VRMECCnCTL -> VRAMnCTL)
54	3561	correction: "It is only cleared by ..." -> "It is cleared by ..."
54	3561	correction: "It is only cleared by ..." -> "It is cleared by ..."
54	3561	cross reference to VRAMnCTL register added
54	3562	"ECC modes overview" table corrected
54	3563	VRAM ECC test registers added to table
54	3563	note added concerning accesses after register modifications
54	3564	initial value of VRAMnCTL corrected
54	3564	correction: ECC error judgement disabled by VRAMnCTL.ECERVF = 0
54	3564	correction: ECC error judgement disabled by VRAMnCTL.ECERVF = 0
54	3565	ECERVF description corrected
54	3565	EC2EDIC description corrected
54	3565	EC1EDIC description corrected
54	3565	correction: ECC error judgement disabled by VRAMnCTL.ECERVF = 0
54	3565	correction: ECC error judgement disabled by VRAMnCTL.ECERVF = 0
54	3566	correction: ECC error judgement disabled by VRAMnCTL.ECERVF = 0
54	3567	typo correction (R/W indicators corrected)
54	3568	"VRAMECCTMC – ECC test mode control register" added
54	3570	"VRAMECCTED1 – ECC encode/decode input/output substitution register 1" register added
54	3570	correction: "handles lower 32 bit" -> "handles upper 32 bit data"
54	3570	correction: ECEDB[31:0] -> ECEDB[63:32]

Revision 1.00 History, Date Feb 27, 2015 (37/37)

Section	Page	Summary
54	3570	correction: "RAM data [31:0]" -> "RAM data [63:32]"
54	3571	"VRAMECCTED2 – ECC encode/decode input/output substitution register 2" register added
54	3571	correction: "handles upper 32 bit" -> "handles lower 32 bit data"
54	3571	correction: ECEDB[63:32] -> ECEDB[31:0]
54	3572	"VRAMECCTRC – ECC test redundant bit data control register" added
54	3572	SYND[7:0] bits removed

Revision 1.10 History

This revision history list shows all modifications of

Rev. 1.10 compared to Rev. 1.00.

NOTE

The chapter and page numbers in the table below refer to the older Rev. 1.10 and thus may not be valid for the current revision.

Revision 1.10 History, Date May 29, 2015 (1/5)

Section	Page	Summary
	4	Product Naming added
1	95	D1M1H external memory interfaces SDRAM I/F Max. clock increased to 100MHz
1	97	correction: voltage regulator of internal logic (ISO) (3.3V, 5V → 3.3V)
2	129	description removed concerning PDSC21
2	132	PDSCn_[15:0] description improved
2	141	correction: Pin Name CSIG0DCS, CSIG2DCS, CSIG3DCS, CSIG4DCS, CSIH0DCS, CSIH1DCS removed
2	204	correction: CSIH0DCS removed as 2nd alternative input of P0_1
2	205	correction: CSIG0DCS, CSIG1DCS, CSIG2DCS, CSIH0DCS, and CSIH1DCS removed
2	205	correction: CSIG3DCS, CSIG1DCS removed
2	211	description added concerning PDSC21
2	213	Table 2.109 Port42(P42) corrected
2	213	Table “Port 42(P42)” corrected
2	215	Table “Port 44(P44)” corrected
2	216	Table “Port 45(P45)” corrected
2	231	CAUTION added
2	232	CAUTION added
2	233	CAUTION added
2	234	Pin category corrected
2	234	SDRADM[3:0], SDRACS, SDRACAS, SDRARAS, SDRAWE pin status corrected while POC0RES = L SDRBDM[3:0], SDRBCS, SDRBCAS, SDRBRAS, SDRBWE pin status corrected while POC0RES = H
2	235	Pin category corrected
2	235	Pin category corrected
2	235	PWRCTL pin status corrected during DEEPSTOP SDRADM[3:0], SDRACS, SDRACAS, SDRARAS, SDRAWE pin status corrected while POC0RES = L
2	236	SDRBDM[3:0], SDRBCS, SDRBCAS, SDRBRAS, SDRBWE pin status corrected while POC0RES = H
2	236	SDRADM[3:0], SDRSCS, SDRSCAS, SDRSRAS, SDRAWE pin status corrected while POC0RES = L SDRBDM[3:0], SDRBCS, SDRBCAS, SDRBRAS, SDRBWE pin status corrected while POC0RES = H
2	239	SDRADQ[31:0] and CAUTION added
2	240	Pin TVCC removed
2	242	Pin TVSS removed
3	248	description concerning calculation of ECC value added

Revision 1.10 History, Date May 29, 2015 (2/5)

Section	Page	Summary
3	248	“ECC error detection” description changed
3	256	“and correction and 1-bit error detection” removed
3	256	correction; “ECC error detection and correction” → “ECC error detection”
3	262	description concerning calculation of ECC value added
3	264	correction: “cache miss occur” → “a cache miss occur”
7	392	BECMBUSERRINFO0 added
7	393	BECMBUSERRINFO1 added
9	447	RRAM module reset status added into AWO area, note5 added
10	469	Isolated-Area power supply description corrected
12	484	D1M1H Clock Controller block diagram corrected concerning SDR-SRAM Memory Controller
12	526	correction: PLL1C initial value of D1L1, D1L2(H), and D1M1(H) corrected(6000 003B_H → 0000 003B_H)
12	613	D1M1H SDRBCLK increased to 200MHz
12	617	note1 added
12	633	typo correction (CLMA Test Reset Signal → CLMA0 Error(_CLMA0RES) Signal)
12	633	“RESCLM” function corrected
12	635	ERRMSK removed
12	635	“RESCLM” function corrected
12	635	typo correction
12	635	typo correction
12	637	ERRMSK removed
12	637	“RESCLM” function corrected
14	676	“D1M1H: 64 bit x 100MHz = 800 MB/sec” added
14	688	description of PEID[2:0] bits added concerning Cross-connect XC1 master
15	720	table “D1M1H unused SDRAM interface handling” added
15	726	note added
15	728	table “ML-E unused SDRAM interface handling” added
15	732	“(e.g. <SDRAn_base> + 1 0000$_H$)” removed
15	735	description of CMD[2:0] bits corrected
15	746	correction: bits 15 to 13 are reserved
15	748	table “1Memory configuration settings(1/2)” corrected
15	749	table “1Memory configuration settings(2/2)” corrected
15	751	“Refresh counter reload” changed
15	754	15.5 SDR-SDRAM Transaction Restrictor added
15	757	table “x 16-bit SDRAM (DBCONF.BWIDTH[1:0] = 01$_B$)” corrected
15	757	table “1 x 32-bit or 2 x 16-bit SDRAM (DBCONF.BWIDTH[1:0] = 10$_B$)” corrected
15	758	table “32-bit interface” corrected
15	758	table “16-bit interface” corrected
16	761	note added
16	768	“(e.g. <SDRBn_base> + 1 0000$_H$)” removed
17	891	typo corrected
22	1224	description concerning calculation of ECC value added
22	1250	“ERRD Bit” description changed
22	1256	CRCREG[14:0] Flag description added

Revision 1.10 History, Date May 29, 2015 (3/5)

Section	Page	Summary
22	1289	RFE Bit description added
22	1302	CFMC[7:0] Bits description added
22	1348	description added concerning channel reset mode
22	1349	description added concerning channel reset mode
22	1353	description added concerning global reset mode
22	1354	description added concerning global reset mode
22	1355	description added concerning global reset mode
22	1358	RSCAN description corrected
22	1364	correction: “Three CANm bit times” → “Four CANm bit times”
22	1375	description added concerning channel halt mode
22	1377	description added concerning channel halt mode
22	1386	correction: “Communication speed” → “Communication Rate”
22	1406	correction: “Note: m = 0 to 2” → “Note: m = 0 to 1”
23	1422	description concerning calculation of ECC value added
23	1422	note added
23	1425	register name corrected (RSCFDnCFDGRMCFG → RSCFDn(CFD)GRMCFG)
23	1425	23.2.3 CAN FD protocol switchover added
23	1427	Global interface mode select register RSCANnGRMCFG added
23	1427	correction: “(r = 0 to 63)” → “classical CAN mode: r = 0 to 29, CAN FD mode: r = 0 to 41”
23	1431	Global interface mode select register RSCANnGRMCFG added
23	1436	ERRD bit description corrected
23	1442	CRCREG[14:0] Flag description corrected
23	1476	description of RFE Bit corrected
23	1489	description of CFMC[7:0] corrected
23	1513	description of THLEN Bit corrected
23	1534	description of THLE bit corrected
23	1535	description of THLMC[4:0] bits corrected
23	1541	description of C2ICBCE bit corrected
23	1541	description of C1ICBCE/C0ICBCE bits corrected
23	1542	description of RTME/ICBCTME bits corrected
23	1545	Globoal FD configuration register and global CRC configuration register added
23	1549	note1 added
23	1550	correction: “RAM Test Page Access Register (r = 0 to 63)” → “RAM Test Page Access Register (r = 0 to 29)”
23	1553	typo correction (error counter → error occurrence counter)
23	1554	description of ERRD bit corrected
23	1566	RFEE and FDOE bits added
23	1567	typo correction (Transceiver → Transmitter)
23	1567	description of REFE and FDOE bits added
23	1568	typo correction (transmission delay → transmitter delay)
23	1572	description of TDCR[6:0] Flag added
23	1573	SCNT[3:0] bit added
23	1573	description of SCNT[3:0] flag added
23	1577	typo correction (RSCFDnCFDRFCCA → RSCFDnCFDRFCCx)

Revision 1.10 History, Date May 29, 2015 (4/5)

Section	Page	Summary
23	1589	RSCFDnCFDGFDCFG — Global FD configuration register added
23	1590	RSCFDnCFDGCRCCFG —Global CRC configuration register added
23	1607	typo correction (RSCFDnCFDRMFDSTSp → RSCFDnCFDRMFDSTSq)
23	1610	description of RFE bit added
23	1611	description of RFMC[7:0] flag added
23	1618	typo correction (RSCFDnCFDRMFDSTSp → RSCFDnCFDRFFDSTSx)
23	1625	description of CFMC[7:0] bits added
23	1633	typo correction (RSCFDnCFDCFFDCSTSp → RSCFDnCFDCFFDCSTSk)
23	1633	elimination of unnecessary whitespace
23	1633	elimination of unnecessary whitespace
23	1645	typo correction (“m = 0 to 5” → “m = 0 to 2”)
23	1672	description of THLE bit added concerning channel reset mode
23	1673	description of THLMC[4:0] bits added concerning channel reset mode
23	1679	description of C2ICBCE bit added concerning global reset mode
23	1679	description of C1ICBCE/C0ICBCE bits added concerning global reset mode
23	1680	description of RTME/ICBCTME bits added concerning global reset mode
23	1683	RS-CANFD description corrected
23	1688	correction: “RAM Test Page Access Register (r = 0 to 63)” → “RAM Test Page Access Register (r = 0 to 41)”
23	1689	Table 23.174 Channel Mode Transition Time typo correction (Three → Four)
23	1692	Table 23.176 Registers Initialized in Global Reset Mode or Channel Reset Mode
23	1696	addition concerning CAN FD mode
23	1697	“CAN3 bit time clock” removed
23	1699	typo correction (“Setting the TMME bit” → “Setting the TMME bit to 1”)
23	1700	addition concerning the transition to channel halt mode
23	1700	description of Interval Transmission Function corrected
23	1701	typo correction (RSCANnGCFG → RSCFDn(CFD)GCFG, RSCANnCFCCk → RSCFDn(CFD)CFCCk)
23	1702	addition concerning the transition to channel halt mode
23	1702	elimination of unnecessary whitespace
23	1709	correction: CRCREG[15:0] → CRCREG[14:0]
23.0	1710.0	typo correction (RSCFDnCFDGRMCFG → RSCFDn(CFD)GRMCFG, RSCFDnCFDCnNCFG → RSCFDnCFDCnNCFG, RSCFDnCFDCnDCFG → RSCFDnCFDCmDCFG), note1 added
23	1711	typo correction (RSCFDnGCFG → RSCFDn(CFD)GCFG)
23	1713	typo correction (RSCFDn(CFD)CmCFG → RSCANnCmCFG, RSCFDn(CFD)CmNCFG→ RSCFDnCFDCmNCFG RSCFDn(CFD)CmDCFG → RSCFDnCFDCmDCFG)
23	1713	correction: “(r = 0 to 63)” → “(classical CAN mode: r = 0 to 29, CAN FD mode: r = 0 to 41)”
23	1714	correction: Communication speed → Communication Rate
23	1714	note added
23	1714	correction: “baud rate prescaler division values” → “nominal bit rate prescaler / data bit rate prescaler division values”
23	1717	correction: Transmit buffer mx16+0 → Transmit buffer 0, Transmit buffer mx16+15 → Transmit buffer 15
23	1719	correction: “a delay of” → “a delay up to”

Revision 1.10 History, Date May 29, 2015 (5/5)

Section	Page	Summary
23	1726	correction: “the first bit of the CRC field” → “the first bit of CRC field for the next transmission”
23	1727	correction: “only when the EEFE bit in the RSCANnGCFG register is 1 in classical CAN mode” → “In classical CAN mode, only when the EEFE bit in the RSCANnGCFG register is 1”
23	1727	correction: “only when the EEFE bit in the RSCANnGCFG register is 1 in classical CAN mode” → “In classical CAN mode, only when the EEFE bit in the RSCANnGCFG register is 1”
23	1729	elimination of unnecessary whitespace
23	1730	correction: “register (a = 0 to 8) is 1” → “register is 1”
23	1738	correction: “Note: m = 0 to 2” → “Note: m = 0 to 1”
23	1743	correction: “r = 0 to 63” → “r = 0 to 29(Classical CAN mode), 0 to 41(CAN FD mode)”
23	1750	correction: “the values” → “the value after reset”
23	1751	typo correction: “Receiveption” → “Receive”
30	2293	Figure 30.13 General Timing Diagram for Interval Timer Function corrected
35	2509	note added
35	2514	note2 added
35	2518	correction: bit 17 is reserved
35	2518	correction: bit 17 is reserved
37	2597	“CKSC_IVDCE0VOS_CTL” added
37	2617	correction: (D1M2(H) only → D1M1(H) and D1M2(H) only)
37	2617	table “CKSC_IVDCE0VOS_CTL register contents” changed
37	2618	correction: (D1M2(H) only → D1M1(H) and D1M2(H) only)
38	2698	Examples of input video image size corrected
38	2812	Note removed
40	3117	Video Output Monitor behaviour after error detection removed
40	3119	Monitor process status description changed
40	3126	VOCAnSELMON[3:0]] description changed
40	3127	“VOCAnMKVOC” function corrected
40	3129	caution added
42	3165	typo corrected
44	3213	“Reset Controller JCUA0RES” added
44	3267	“Software Reset Processing” description added
45	3343	“Relationship between analog input voltage and A/D conversion result” added
46	3416	typo correction (ISMnGGEN → ISMnGGCE)
46	3417	typo correction (ISMnGGEN → ISMnGGCE)
46	3418	typo correction (ISMnGGEN → ISMnGGCE)
46	3418	typo correction (ISMnGGEN → ISMnGGCE)
46	3419	typo correction (ISMnGGEN → ISMnGGCE)
46	3419	typo correction (ISMnGGEN → ISMnGGCE)
46	3433	description concerning calculation of ECC value added
52	3520	description concerning calculation of ECC value added
52	3523	Value after reset corrected ($0000\ 0000_H$ → $0000\ 0006_H$)
52	3528	description concerning calculation of ECC value added

Revision 1.20 History

This revision history list shows all modifications of

Rev. 1.20 compared to Rev. 1.10.

NOTE

The chapter and page numbers in the table below refer to the older Rev. 1.20 and thus may not be valid for the current revision.

Revision 1.20 History, Date Oct 26, 2015 (1/21)

Section	Page	Summary
1	96	corrected Table 1.2 RH850/D1M Products Overview (2/3): (“2D Graphics Processing Unit (GPU2D), 80 MHz operation clock” → “2D Graphics Processing Unit (GPU2D), 80 MHz operation clock” and “2D Graphics Processing Unit (GPU2D), 100 MHz operation clock”
2	102	corrected Figure 2.2 D1L2 pin connection diagram (144 pins LQFP package): (FLMD0 pin: “↔(input/output)” → “→ (Input)”)
2	132	(4) PDSCn/JPDSC0 — Port drive strength control register: “NOTE” added concerning table 2.25 PDSCn register contents
2	134	(3) Write Sequence of the Port Register to be Protected: “Step 4” corrected Figure 2.8 Write Sequence of the Port Register to be Protected: corrected
2	135	(1) Port Control Register (PCRn_m): bit 30, bit 28, bit 24 in the bit chart, corrected (“R” → “R/W”)
2	135	Table 2.28 PCRn_m register contents: functional description on PINV, PODC, PDSC, PISA, PIS, corrected
2	146	corrected Table 2.30 D1L1 ports input buffer characteristics (3/3): P21: “PISAn_m = 1” (“<t.b.c.>” → “—” (none)), “PISAn_m = 0” (“<t.b.c.>” → “CMOS1”), “Default” (“<t.b.c.>” → “CMOS1”)
2	146	Table 2.30 D1L1 ports input buffer characteristics (3/3): P21_6 to P21_9 deleted
2	214	Table 2.112 Port 43 (P43): P43_2 to P43_9: “Input” of “2nd Alternative”, corrected P43_7 to P43_9: “Input” of “3rd Alternative”, corrected
2	217	Table 2.121 Port 46 (P46): corrected
2	240	Table 2.142 Recommended connection of unused pins (2/3): “B5VCC” pin added
3	242	description corrected (“V8503v5-V” → “V850E3v5-V”, “R01US0058EJxxxx” → “R01US0123EJxxxx”)
3	243	3.1.1 Processor Element (PE) and CPU: description of “Floating Point Unit (FPU)”, “Code Flash cache”, corrected description of “CPU snooze instruction”, added
3	248 to 253	3.1.5 System Error Notification Control Function (SEG): added
3	254 to 255	3.1.6 Product denpendent registers: added
3	273	3.3.5 XC Cache control registers (D1L2(H), D1M1(H), D1M2(H)): NOTE” added concerning table 3.25 List of XC Cache control registers
3	297	3.3.6.5 AXCD1STERSTRn_PE1_OS . XC cache data RAM 1st error status register (n = 0 - 3): bit 25, bit 24, bit 17, bit 18, bit 9, bit 8, bit 1, bit 0 in the bit chart, corrected (bit name: “n+3” → “4n+3”)
3	297, 298	Table 3.51 AXCD1STERSTRn_PE1_OS register contents: corrected (bit name: “n+3” → “4n+3”, “n+2” → “4n+2”, “n+1” → “4n+1”, “n” → “4n”
3	305	3.3.6.12 AXCT1STEADRn_PE1_OS . XC cache tag RAM 1st error address register (n = 0, 1): address corrected (“FFC6 6450_H + 4 × n” → “FFC6 7450_H + 4 × n”)
3	306	3.4.1 Synchronization of Store Instruction Completion and Subsequent Instruction Generation: corrected (“(TS.W, etc.)” → “(ST.W, etc.)”)

Revision 1.20 History, Date Oct 26, 2015 (2/21)

Section	Page	Summary
4	310	4.1.1 Register protection unlock sequence: description added
4	313	4.2.1 Port protection clusters: description added
4	321	Table 4.9 CLMAnPCMD Register Contents: functional description on CLMAnREG[7:0], corrected
6	338	“CAUTION” added
7	346	7.4 Interrupt Exception Handler and Priority Operations: description corrected, and “NOTE” added
7	394	corrected (description of “which interrupt?”, deleted)
9	455	Table 9.2 Reset sources and targets overview: corrected Real-Time Clock RTCA0: “Reset level 2 (SYSRES)” (“reset” → “not reset”)
9	457	corrected (“sinals” → “signals”)
9	463	9.3.2.1 RESF, RESFR — Reset factor register: description correted
9	470	9.3.4.1 PWRGD_CNT — PWRGD counter register: “NOTE” added
9	472	Figure 9.5 POC reset timing: Note corrected
9	473	Figure 9.6 When $\overline{\text{RESET}}$ is Released before the Flash Sequence is Completed: Note 1 corrected
9	475	9.4.3 External Reset Signal ($\overline{\text{RESET}}$): description corrected
10	477	10.1 Function: description corrected
10	477	Table 10.1 Power Supply Pins Overview (1/2): description of “EVCC EVSS”, corrected
12	499	12.1.4 Clock selectors: description corrected
12	499	“CAUTION” added
12	503	12.2.2 Sub Oscillator (SubOsc): description corrected of “SubOsc enable trigger/disable trigger”
12	504	“NOTE” corrected
12	525	Table 12.13 ROSCSTPM register contents: fuctional description on ROSCSTPMSK, corrected
12	582	12.3.5.8 CKSC_ARTCAS_CTL . C_AWO_RTCA source clock selection register: “CAUTION” added
12	617	12.3.6.28 CKSC_IRLINS_ACT . C_ISO_RLIN source clock active register: bit 1 in the bit chart, corrected (initial value: “1” → “1*[1]”)
12	617	note 1. added concerning bit chart
12	622	Table 12.102 CPU and buses subsystems clocks maximum frequencies: C_ISO_PCLK: “D1M1H” corrected (100 MHz → 50 MHz) ETNBPCLK: “D1M1” (80 MHz → 80 MHz), “D1M1H” corrected (80 MHz → 50 MHz)
12	625	Table 12.106 Isolated-Area clock domain clocks maximum frequencies: C_ISO_RSCANXIN: “D1L1, D1L2, D1L2H, D1M1, D1M1H, D1M2, D1M2H” corrected (8 MHz → 16 MHz)
12	627	Table 12.109 FOUTDIV register contents: functional discription on FOUTDIV[9:0], corrected (Note 1 deleted)
12	629	Table 12.111 Units: corrected Clock Monitor A: “D1M2” added, “D1M2(H)” → “D1M2H” corrected
12	631	12.7.2 CLMA Enabling: description corrected
12	638	Table 12.117 CLMAnCTL0 Register Contents: fuctionctional description on CLMAnCLME, corrected, and Note 1 added
13	648	description added
13	648	13.1 Functions: description of “HALT mode”, corrected
13	648	13.1 Functions: description of “HALT” and “RUN”, deleted
13	648	13.1 Functions: “NOTE” added
13	648	Table 13.1 Clock Supply: corrected (“Clock” → “Unit Clock Name”)

Revision 1.20 History, Date Oct 26, 2015 (3/21)

Section	Page	Summary
13	650	13.1.3.1 I/O buffer hold state: description corrected
13	650	13.1.3.2 I/O buffers during DEEPSTOP mode: description corrected
13	651	13.1.4 Clock supply in DEEPSTOP mode: description of “ROSCSTPM: High Speed IntOsc”, corrected
13	651	13.1.4 Clock supply in DEEPSTOP mode: description of “NOTES 1”, corrected
13	653	Figure 13.1 State transition diagram of stand-by mode: corrected (“Wake-up event” → “Wake-up event*2”, “Stand-by request” → “Stand-by request*1” Note 1 corrected
13	655	13.2.2.1 STBC0PSC — Stand-by control register: description corrected
13	656	13.2.2.2 WUF0 — Wake-up factor register: description corrected
13	657	13.2.2.3 WUFMSK0 — Wake-up factor mask register: description corrected
13	658	13.2.2.4 WUFC0 — Wake-up factor clear registers: initial value corrected (“This register is always read as 0000 0000_H.” → “Undefined”) initial value of bit 31 to 0 in the bit chart, corrected (“0” → “—”) R/W of bit 31 to 19 in the bit chart (“W” → “R”)
13	658	Table 13.7 WUFC0 Register Contents: functional description on bit 31 to bit 19 (Reserved), corrected
13	659	13.2.2.5 IOHOLD — Port IOHOLD control register: description corrected
13	662	Figure 13.2 Example of DEEPSTOP mode transition: corrected
14	689	Table 14.15 PBUS structure (1/7): corrected PBG0A: Channel 1 added, (Channel: “1 to 3” → “2 to 3”), (Channel 6: “Data Flash ECC” → “Code Flash ECC / Data Flash ECC”) PBUS0: Module added (SYSCTRL, ECCIC1, ECCCPU1, ECCAXC))
14	690	Table 14.15 PBUS structure (2/7): corrected PBG2A: Channel 2 (“GPU2D” → “GPU2D0”) PBG3A: Channel 4 to Channel 7 (“RLIN30” → “RLN30”, “RLIN31” → “RLN31”, RLIN32” → “RLN32”, RLIN33” → “RLN33”) and Channel 8 (“RSCAN / RSCFD0*1” → “RSCAN0 / RSCFD0*1”)
14	691	Table 14.15 PBUS structure (3/7): corrected PBG30A: Channel 0 to Channel 11 (note 1 and note 2, added) PBG30B: Channel 0 (“ISM0RAMECC” → “ECCISM0”)
14	692	Table 14.15 PBUS structure (4/7): corrected PBG3B: Channel 3 (“SLPWGA” → “SLPWGA0”)
14	693	Table 14.15 PBUS structure (5/7): corrected PBUS3 (“PGB32A” → “PBG32A”): Channel 8 (“Analog port filter control FLCA0” → “Analog port filter control FCLA0”), Channel 15 (“Analog port filter control FLCA1” → “Analog port filter control FCLA1”), and PBUS3 (“PGB32B” → “PBG32B”)
14	694	Table 14.15 PBUS structure (6/7): corrected PBG5A: (Channel: “4 to 6” → “4 to 5”), Channel 6 added, Channel 7 (“SELB” added), Channel 11 (“Reserved” → “SD_ISO_VDD (QOS registers)”), Channel 12 (“Stand-by Controller” → “SYS (Stand-by Controller, RESF,RESFR, RESFC, RESFCR, SWRESA, PROTCMD0, PROTS0)”)
14	695	Table 14.15 PBUS structure (7/7): corrected PBUS5: Module added (CLMA0 to CLMA6, CLMAC, SYS (others), PWRG)
14	707	(4) Peripheral Device Protection Setting Register 0 (IPGPMTUM0, IPGnPMT0, and IPGnPMTUM0 (n=0)): bit chart corrected bits of bit 2 to bit 0 (“X0” → “—”, “W0” → “—”, “R0” → “—”) R/W of bit 2 to bit 0 (“R/W” → “R”)
14	707	Table 14.27 IPGPMTUM0, IPGnPMT0, and IPGnPMTUM0 (n=0) register contents: functional description of bit 3 to bit 0 (—), corrected
14	709	(7) Peripheral Device Protection Setting Register 4 (IPGPMTUM4, IPGnPMT4, and IPGnPMTUM4 (n=0)): bit chart corrected bits of bit 6 to bit 4 (“—” → “X1”, “—” → “W1”, “—” → “R1”) R/W of bit 6 to bit 4 (“R” → “R/W”)

Revision 1.20 History, Date Oct 26, 2015 (4/21)

Section	Page	Summary
14	709	Table 14.30 IPGPMTUM4, IPGnPMT4, and IPGnPMTUM4 (n=0) register contents: functional description on X1, W1, R1
14	710	14.5.4.1 List of Registers: desription corrected
14	713	Table 14.31 FSGDxxPROTn register contents (2/2): functional discription on PROTUM, corrected (note added)
15	736	Table 15.7 D1M1H unused SDRAM interface handling: corrected (“36 bit” → “32 bit”, “DM[3:0]: all used” and “DM[3:2]: leave open” added)
15	745	Table 15.14 DBDMOV register contents: functional description on TRCDC[1:0], corrected (“01_B” → “00_B”)
15	749	Table 15.17 DBTR2 register contents: functional description on WRRD[3:0], corrected (“1 cycles” → “2 cycles”, “2 cycles” → “3 cycles”, “12 cycles” → “13 cycles”)
17	887	Figure 17.16 Example of Initial Setting Flow in SPI Operating Mode: corrected
17	899	corrected ($f_{PHCLK}/f_{B\Phi}$: “100_B” → “011_B”)
17	900	corrected ($f_{PHCLK}/f_{B\Phi}$: “110_B” → “100_B”)
17	900	“NOTE” corrected (“$f_{B\Phi}$” → “f_{PHCLK}”)
17	901	Table 17.13 Clock adjustment settings examples (for f_{PHCLK} = 480 MHz): corrected (CKDLYOC[2:0]: “100_B” → “011_B”, “110_B” → “100_B”)
17	901	description of “Sampling point calibration procedure”, corrected
17	902	description of “Example: Calibration for f_{PHCLK} = 480 MHz, BΦ = 160 MHz”, corrected
18	904	Table 18.4 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
18	908	18.2.1 Functional Overview: description of “Maximum transfer clock frequency”, corrected
18	909	Figure 18.1 CSIG Block Diagram: corrected (“Prescaler Baudrate Generator” → “BRG”)
18	915	Table 18.16 CSIGnSTR0 Register Contents (1/2): functional description on CSIGnDCE, corrected
18	924	Table 18.24 List of Cautions when Setting Registers: description of contents for CSIGnBCTL0, corrected
18	925	18.4.1 Interrupt Delay: corrected (“CSIGnCFG0.CSIGnCKR” → “CSIGnCTL1.CSIGnCKR”)
18	925	18.4.2 INTCSIGTIC (Communication Status Interrupt): corrected (“CSIGnCFG0.CSIGnCKR” → “CSIGnCTL1.CSIGnCKR”)
18	926	Figure 18.4 Generation of INTCSIGTIC at the Beginning of Communication: corrected
18	926	18.4.3 INTCSIGTIR (Reception Status Interrupt): corrected (“CSIGnCFG0.CSIGnCKP” → “CSIGnCTL1.CSIGnCKR”)
18	928	18.5.1.1 Master Mode: corrected (“CSIGnCFG0.CSIGnCKP” → “CSIGnCTL1.CSIGnCKR”)
18	931	description of “Transfer clock frequency upper and lower limits”, corrected
18	937	18.5.8.1 Slave Mode: description corrected
18	942	Figure 18.24 Overrun Error Detection: corrected
18	943	18.6.1 Master Mode Transmission/Reception by DMA: corrected (“CSIGnCFG0.CSIGnCKP” → “CSIGnCTL1.CSIGnCKR”)
19	946	Table 19.6 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
19	947	Table 19.9 External Input/Output Signals: corrected (“CSIHTCSS[5:1]” → “CSIHTCSS[7:1]”)
19	961	Table 19.18 CSIHnSTR0 Register Contents (2/3): funtional description on CSIHnTMOE and CSIHnOFE, corrected
19	962	Table 19.18 CSIHnSTR0 Register Contents (3/3): funtional description on CSIHnPE, corrected
19	973	Table 19.25 CSIHnCFGx Register Contents (2/5): funtional description on CSIHnPE, corrected (“CSHBAnDLSx[3:0]” → “CSIHnDLSx[3:0]”)

Revision 1.20 History, Date Oct 26, 2015 (5/21)

Section	Page	Summary
19	977	19.3.12 CSIHnTX0W . CSIHn Transmit Data Register 0 for Word Access: value after reset corrected (“X0XX $XXXX_H$” → “Undefined”)
19	979	19.3.14 CSIHnRX0W . CSIHn Receive Data Register 0 for Word Access: value after reset corrected (“0XXX $XXXX_H$” → “Undefined”)
19	982	Table 19.32 Notes on Setting Registers (1/2): description of contents for CSIHnSTR0, corrected (“->” → “→”)
19	983	Table 19.32 Notes on Setting Registers (2/2): description of contents for CSIHnCFGx x = 0, corrected
19	997	19.5.3.1 Configuration Registers: description of “Hold time T_{hold}”, corrected
19	999	Figure 19.14 Chip Select and RCB Example: corrected (“(dominant: nigh priority)” → “(dominant: high priority)”)
19	1011	19.5.10 Slave Select (SS) Function: description corrected
19	1013	19.5.11.1 Slave Mode: description of “2.” corrected
19	1014	Figure 19.32 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 0): corrected (“CSIHnRX0-full” → “CSIHnRX0W-full”, “Read CSIHnRX0” → “Read CSIHnRX0W”)
19	1014	Figure 19.33 Busy Signal from the Slave (Direct Access Mode; CSIHnCFGx.CSIHnDAPx = 1): corrected (“CSIHnRX0-full” → “CSIHnRX0W-full”, “Read CSIHnRX0” → “Read CSIHnRX0W”)
19	1017	Figure 19.36 Data Consistency Check Functional Block Diagram: corrected (“On error” → “Error occurrence”)
19	1025	Figure 19.46 Example of CPU-Controlled High-Priority Communication: corrected (“INTCSIHTIC” → “INTCSIHTIJC”)
19	1026	Figure 19.48 Transition from High-Priority Mode to Low-Priority Mode: corrected
19	1039	19.6.3.3 Transmit/Receive in Slave Mode when Job Mode is Disabled: description corrected (additional description)
20	1047	Table 20.6 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
20	1052	Figure 20.1 LIN/UART Interface Block Diagram: corrected (“LINn interrupt controller circuit” → “LINn interrupt control circuit”)
20	1053	Table 20.11 Registers (1/2): corrected (“RLIN3n” → “RLN3n”)
20	1054	Table 20.11 Registers (2/2): corrected (“RLIN3n” → “RLN3n”)
20	1055	20.3.2.1 RLN3nLWBR — LIN Wake-Up Baud Rate Select Register: description of “LWBR0 Bit (Wake-up Baud Rate Select)”, corrected
20	1061	20.3.2.6 RLN3nLBFC — LIN Break Field Configuration Register: description of “BDT[1:0] Bits (Transmission Break Delimiter (High Level) Width Select)”, corrected
20	1064	20.3.2.9 RLN3nLIE — LIN Interrupt Enable Register: description corrected, and description of “ERRIE Bit (Error Detection Interrupt Request Enable), corrected”
20	1064	Table 20.20 RLN3nLIE Register Contents: functional description on bit 7 to bit 4 (Reserved), corrected
20	1066	note added concerning table 20.21 RLN3nLEDE Register Contents
20	1069	20.3.2.12 RLN3nLTRC — LIN Transmission Control Register: description of “RTS Bit (Response Transmission/Reception Start)” and “FTS Bit (Frame Transmission/Wake-up Transmission/Reception Start)”, corrected
20	1071	Table 20.25 RLN3nLST Register Contents: functional description on D1RC, corrected
20	1077	20.3.2.17 RLN3nLIDB — LIN ID Buffer Register: description of “IDP[1:0] Bits (Parity Setting)”, corrected
20	1079	Table 20.30 RLN3nLDBRb Register Contents: functional description on LDB[7:0], corrected
20	1084	20.3.3.4 RLN3nLBFC — UART Configuration Register: description of “UTPS Bit (UART Output Polarity Switch)”, corrected
20	1086	20.3.3.5 RLN3nLSC — UART Space Configuration Register: description of “IBS[1:0] Bits (Inter-Byte Space Select)”, corrected (additional description)

Revision 1.20 History, Date Oct 26, 2015 (6/21)

Section	Page	Summary
20	1088	20.3.3.7 RLN3nLCUC — UART Control Register: description of “OM0 Bit (LIN Reset)”, corrected
20	1092	20.3.3.10 RLN3nLST — UART Status Register: description of “ERR Flag (Error Detection Flag)”, corrected (“UPER, EXBT, OER, and ER flag” → “UPER, IDMT, EXBT, FER, OER, and BER flag”)
20	1100	20.3.3.17 RLN3nLUOR1 — UART Option Register 1: description of “UTIGTS Bit (Transmission Interrupt Generation Timing Select)”, corrected
20	1105	20.4 Interrupt Sources: description corrected
20	1110	20.7.1.1 Header Transmission: description corrected
20	1111	Table 20.60 Processing in Response Transmission: description of “(4)”, corrected (“RFC” → “RLN3nLDFC”)
20	1111	20.7.1.2 Response Transmission: “NOTE” corrected
20	1112	20.7.1.3 Response Reception: “NOTE” corrected
20	1115	20.7.3.1 Transmission of LIN Frames: description of “[Frame Separate Mode]”, corrected
20	1116	20.7.3.3 Multi-Byte Response Transmission/Reception Function: description corrected
20	1122	Table 20.64 LIN/UART Interface (UART Mode) Transmission Processing (1/2): description of “(3)”, corrected
20	1123	20.8.1 Transmission: “NOTE” added
20	1124	Table 20.65 UART Buffer Transmission Processing in LIN/UART Interface (in UART Mode) (1/2): description of “(2)”, corrected
20	1125	Table 20.65 UART Buffer Transmission Processing in LIN/UART Interface (in UART Mode) (2/2): description of “(4)”, corrected
20	1125	20.8.1.2 UART Buffer Transmission: “NOTE” added
20	1129	20.8.2 Reception: “NOTE” added
20	1133	20.8.3.4 Expansion Bit Reception (with Data Comparison): description corrected
20	1136	20.9 LIN Self-Test Mode: description corrected
20	1139	20.9.3 Reception in LIN Master Self-Test Mode: description corrected
20	1140	20.10 Baud Rate Generator: description corrected
20	1141	20.10.1 LIN Master Mode: description corrected
20	1143	Figure 20.31 Example of Noise Filter Circuit: corrected
21	1145	Table 21.4 Clock Supply: corrected (“Clock for the Unitn” → “Unit Clock Name”)
21	1153	Table 21.9 RIICnCR1 Register Contents (2/2): functional description on SDAI, corrected
21	1164	21.3.6 RIICnMR3 — I^2C Bus Mode Register 3: title of “NF[1:0] Bits (Digital noise Filter Stage Selection)”, corrected
21	1168	21.3.8 RIICnSER — I^2C Bus Status Enable Register: access corrected
21	1172	21.3.10 RIICnSR1 — I^2C Bus Status Register 1: bit name in the bit chart, corrected bits of bit 5, bit 3 to bit 0 (“RIICn” deleted)
21	1172	Table 21.18 RIICnSR1 Register Contents: bit name corrected bits of bit 5, bit 3 to bit 0 (“RIICn” deleted)
21	1175	21.3.11 RIICnSR2 — I^2C Bus Status Register 2: bit name in the bit chart, corrected bits of bit 7 to bit 0 (“RIICn” deleted)
21	1175	Table 21.19 RIICnSR2 Register Contents: bit name corrected bits of bit 7 to bit 0 (“RIICn” deleted)
21	1225	Table 21.26 RIIC Reset Functions (1/2): corrected RIICnSR1, RIICnSR2: bit name corrected (“RIICn” deleted)
22	1229	Table 22.5 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
22	1232	Table 22.10 RSCAN Module Specifications (1/2): corrected (Reception filter function: “(0 to 128)” → “(0 to 127)”)

Revision 1.20 History, Date Oct 26, 2015 (7/21)

Section	Page	Summary
22	1233	Table 22.10 RSCAN Module Specifications (2/2): description corrected of “Bus off recovery mode selection”
22	1243	Table 22.11 Registers (9/18): symbol name corrected (“RSCAN0MDF019” → “RSCAN0RMDF019”)
22	1255	22.3.2 RSCAN0CmCFG — Channel Configuration Register (m = 0 to 2): description corrected
22	1259	22.3.3 RSCAN0CmCTR — Channel Control Register (m = 0 to 2): description of “CSLPR Bit”, corrected
22	1260	22.3.4 RSCAN0CmSTS — Channel Status Register (m = 0 to 2): access and address corrected
22	1262	22.3.5 RSCAN0CmERFL — Channel Error Flag Register (m = 0 to 2): access corrected
22	1266	22.3.6 RSCAN0GCFG — Global Configuration Register: bit name in the bit chart, corrected bit 31 to bit 16 (“ITRCP” → “ITRCP[15:0]”)
22	1269	22.3.7 RSCAN0GCTR — Global Control Register: access and address corrected
22	1271	22.3.8 RSCAN0GSTS — Global Status Register: access and address corrected
22	1273	22.3.9 RSCAN0GERFL — Global Error Flag Register: access and address corrected
22	1273	Table 22.23 RSCAN0GERFL Register Contents: corrected bit position (“31 to 14, 7, 6, 3” → “31 to 14” “13 to 8” “7, 6”, “5” divided, and “4, 3” added) fuctional description on bit 13 to bit 8 (Reserved), corrected
22	1274	22.3.9 RSCAN0GERFL — Global Error Flag Register: “NOTE” added
22	1275	22.3.10 RSCAN0GTINTSTS0 — Global TX Interrupt Status Register 0: access and address corrected
22	1278	22.3.11 RSCAN0GTSC — Global Timestamp Counter Register: access and address corrected
22	1279	22.3.12 RSCAN0GAFLECTR — Receive Rule Entry Control Register: access and address corrected
22	1280	22.3.13 RSCAN0GAFLCFG0 — Receive Rule Configuration Register 0: access and address corrected
22	1286	22.3.16 RSCAN0GAFLP0j — Receive Rule Pointer 0 Register (j = 0 to 15): access and address corrected
22	1289	22.3.18 RSCAN0RMNB — Receive Buffer Number Register: access and address corrected
22	1295	22.3.24 RSCAN0RFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7): access and address corrected
22	1297	22.3.25 RSCAN0RFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7): access and address corrected
22	1299	22.3.26 RSCAN0RFPCTRx — Receive FIFO Buffer Pointer Control Register (x = 0 to 7): access and address corrected, and description of “RFPC[7:0] Bits”, corrected
22	1308	22.3.32 RSCAN0CFSTSk — Transmit/receive FIFO Buffer Status Register (k = 0 to 8): access and address corrected
22	1311	22.3.33 RSCAN0CFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register (k = 0 to 8): access and address corrected
22	1319	22.3.38 RSCAN0FESTS — FIFO Empty Status Register: access, address, value after reset corrected value after reset of bit 25 to bit 17 in the bit chart, corrected (“0” → “1”) description corrected
22	1321	22.3.39 RSCAN0FFSTS — FIFO Full Status Register: access and address corrected
22	1323	22.3.40 RSCAN0FMSTS — FIFO Message Lost Status Register: access and address corrected
22	1325	22.3.41 RSCAN0RFISTS — Receive FIFO Buffer Interrupt Flag Status Register: access and address corrected

Revision 1.20 History, Date Oct 26, 2015 (8/21)

Section	Page	Summary
22	1326	22.3.42 RSCAN0CFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register: access and address corrected
22	1327	22.3.43 RSCAN0CFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register: access and address corrected
22	1332	22.3.46 RSCAN0TMTRSTSy — Transmit Buffer Transmit Request Status Register (y = 0, 1): address corrected (“0253_H” → “0353_H”)
22	1334	22.3.47 RSCAN0TMTARSTSy — Transmit Buffer Transmit Abort Request Status Register (y = 0, 1): address corrected (“0263_H” → “0363_H”)
22	1344	22.3.52 RSCAN0TMPTRp — Transmit Buffer Pointer Register (p= 0 to 47): access and address corrected
22	1348	22.3.55 RSCAN0TXQCCm — Transmit Queue Configuration and Control Register (m = 0 to 2): access and address corrected
22	1350	22.3.56 RSCAN0TXQSTSm — Transmit Queue Status Register (m = 0 to 2): access and address corrected
22	1350	Table 22.75 RSCAN0TXQSTSm Register Contents: fuctional description on bit 12 to bit 8 (Reserved), corrected
22	1352	22.3.57 RSCAN0TXQPCTRm — Transmit Queue Pointer Control Register (m = 0 to 2): access and address corrected
22	1353	22.3.58 RSCAN0THLCCm — Transmit History Configuration and Control Register (m = 0 to 2): access and address corrected
22	1355	22.3.59 RSCAN0THLSTSm — Transmit History Status Register (m = 0 to 2): access and address corrected
22	1357	22.3.60 RSCAN0THLACCm — Transmit History Access Register (m = 0 to 2): access and address corrected
22	1358	22.3.61 RSCAN0THLPCTRm — Transmit History Pointer Control Register (m = 0 to 2): access and address corrected
22	1359	22.3.62 RSCAN0GTSTCFG — Global Test Configuration Register: access and address corrected, and description of “RTMPS[6:0] Bits”, corrected
22	1361	22.3.63 RSCAN0GTSTCTR — Global Test Control Register: access and address corrected
22	1362	22.3.64 RSCAN0GLOCKK — Global Lock Key Register: access and address corrected
22	1371	Table 22.88 Channel Mode Transition Time (2/2): corrected Channel halt: “Three CANm bit times” → “Four CANm bit times”
22	1372	description of Note 3. concerning table 22.89 Operation a Channel Transitions to Channel Reset Mode/Channel Halt Mode, corrected
22	1380	22.7.1 Transmit Priority Determination: description corrected
22	1385	22.8 Gateway Function: description corrected
22	1388	22.9.4 RAM Test: description corrected (additional description)
22	1395	Figure 22.21 Buffer Setting Procedure: corrected
22	1416	22.11.4 ECCRCAN0TMC — RSCAN0 ECC Test Mode Control Register: access corrected
22	1419	22.11.6 ECCRCAN0TRC — RSCAN0 ECC Redundant Bit Data Control Test Register: access corrected
22	1420	22.11.8 ECCRCAN0HORD — RSCAN0 ECC 7-Bit Redundant Bit Data Hold Test Register: access corrected
22	1420	22.11.9 ECCRCAN0ECRD — RSCAN0 ECC Encode Test Register: access corrected
22	1421	22.11.10 ECCRCAN0ERDB — RSCAN0 ECC Redundant Bit Input/Output Replacement Buffer Register: title corrected
23	1425	Table 23.5 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
23	1425	Table 23.6 Setting Range example of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/D1L/D1M (1/2): table title and description (“Data Bit Rate”, “pclk”, “clkc, clk_xincan*1”) corrected

Revision 1.20 History, Date Oct 26, 2015 (9/21)

Section	Page	Summary
23	1426	Table 23.6 Setting Range example of Operating Frequency Depending on the Transfer Rate and the Number of Channels in Use in the RH850/D1L/D1M (2/2): table title and description (“Data Bit Rate”, “pclk”, “clkc, clk_xincan*1”) corrected
23	1427	Table 23.9 External Input/Output Signals: corrected (“CANmDREN (m = to 2)” → “CANmDREN (m = 0 to 2)”)
23	1429	Table 23.10 RS-CANFD module Specifications (2/3): description of “Reception filter function”, corrected
23	1436	23.3.2.1 RSCANnGRMCFG — Global Interface Mode Select Register: access and address corrected
23	1444	23.3.2.4 RSCANnCmSTS — Channel Status Register (m = 0 to 2): access and address corrected
23	1446	23.3.2.5 RSCANnCmERFL — Channel Error Flag Register (m = 0 to 2): access corrected
23	1453	23.3.3.2 RSCANnGCTR — Global Control Register: access and address corrected
23	1455	23.3.3.3 RSCANnGSTS — Global Status Register: access and address corrected
23	1457	23.3.3.4 RSCANnGERFL — Global Error Flag Register: access and address corrected
23	1457	Table 23.24 RSCANnGERFL Register Contents: corrected bit position (“15 to 3” → “15, 14” “13 to 8” “7, 6”, “5” “4, 3” divided) fuctional description on bit 15, bit 14 (Reserved), corrected
23	1459	23.3.3.5 RSCANnGTSC — Global Timestamp Counter Register: access and address corrected
23	1460	23.3.3.6 RSCANnGTINTSTS0 — Global TX Interrupt Status Register 0: access and address corrected
23	1463	23.3.4.1 RSCANnGAFLECTR — Receive Rule Entry Control Register: access and address corrected
23	1464	23.3.4.2 RSCANnGAFLCFG0 — Receive Rule Configuration Register 0: access and address corrected
23	1470	23.3.4.5 RSCANnGAFLP0_j — Receive Rule Pointer 0 Register (j = 0 to 15): access and address corrected
23	1473	23.3.5.1 RSCANnRMNB — Receive Buffer Number Register: access and address corrected
23	1479	23.3.6.1 RSCANnRFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7): access and address corrected
23	1481	23.3.6.2 RSCANnRFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7): access and address corrected
23	1483	23.3.6.3 RSCANnRFPCTRx — Receive FIFO Buffer Pointer Control Register (x = 0 to 7): access and address corrected
23	1484	23.3.6.4 RSCANnRFIDx — Receive FIFO Buffer Access ID Register (x = 0 to 7): access and address corrected
23	1492	23.3.7.2 RSCANnCFSTSk — Transmit/receive FIFO Buffer Status Register (k = 0 to 8): access and address corrected
23	1495	23.3.7.3 RSCANnCFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register (k = 0 to 8): access and address corrected
23	1503	23.3.8.1 RSCANnFESTS — FIFO Empty Status Register: access and address corrected
23	1505	23.3.8.2 RSCANnFFSTS — FIFO Full Status Register: access and address corrected
23	1507	23.3.8.3 RSCANnFMSTS — FIFO Message Lost Status Register: access and address corrected
23	1509	23.3.8.4 RSCANnRFISTS — Receive FIFO Buffer Interrupt Flag Status Register: access and address corrected
23	1510	23.3.8.5 RSCANnCFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register: access and address corrected

Revision 1.20 History, Date Oct 26, 2015 (10/21)

Section	Page	Summary
23	1511	23.3.8.6 RSCANnCFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register: access and address corrected
23	1518	23.3.9.4 RSCANnTMPTRp — Transmit Buffer Pointer Register (p = 0 to 47): access and address corrected
23	1532	23.3.11.1 RSCANnTXQCCm — Transmit Queue Configuration and Control Register (m = 0 to 2): access and address corrected
23	1534	23.3.11.2 RSCANnTXQSTSm — Transmit Queue Status Register (m = 0 to 2): access and address corrected
23	1536	23.3.11.3 RSCANnTXQPCTRm — Transmit Queue Pointer Control Register (m = 0 to 2): access and address corrected
23	1537	23.3.12.1 RSCANnTHLCCm — Transmit History Configuration and Control Register (m = 0 to 2): access and address corrected
23	1539	23.3.12.2 RSCANnTHLSTSm — Transmit History Status Register (m = 0 to 2): access and address corrected
23	1541	23.3.12.3 RSCANnTHLPCTRm — Transmit History Pointer Control Register (m = 0 to 2): access and address corrected
23	1544	23.3.13.1 RSCANnGTSTCFG — Global Test Configuration Register: access and address corrected
23	1546	23.3.13.2 RSCANnGTSTCTR — Global Test Control Register: access and address corrected
23	1547	23.3.13.3 RSCANnGLOCKK — Global Lock Key Register: access and address corrected
23	1553	23.4.2.1 RSCFDnCFDGRMCFG — Global Interface Mode Select Register: access and address corrected
23	1555	23.4.3.1 RSCFDnCFDCmNCFG — Channel Nominal Bit Rate Configuration Register (m = 0 to 2): description of “NBRP[9:0] Bits”, corrected (additional description)
23	1561	23.4.3.3 RSCFDnCFDCmSTS — Channel Status Register (m = 0 to 2): access corrected
23	1564	23.4.3.4 RSCFDnCFDCmERFL — Channel Error Flag Register (m = 0 to 2): access corrected
23	1568	23.4.3.5 RSCFDnCFDCmDCFG — Channel Data Bit Rate Configuration register (m = 0 to 2): access and address corrected
23	1569	23.4.3.5 RSCFDnCFDCmDCFG — Channel Data Bit Rate Configuration register (m = 0 to 2): description of “DBRP[7:0] Bits”, corrected (additional description)
23	1570	23.4.3.5 RSCFDnCFDCmDCFG — Channel Data Bit Rate Configuration register (m = 0 to 2): description of “DBRP[7:0] Bits”, corrected (additional description)
23	1574	23.4.3.7 RSCFDnCFDCmFDCTR — Channel CAN FD Control Register (m = 0 to 2): access and address corrected
23	1575	23.4.3.8 RSCFDnCFDCmFDSTS — Channel CAN FD Status Register (m = 0 to 2): access corrected
23	1576	23.4.3.8 RSCFDnCFDCmFDSTS — Channel CAN FD Status Register (m = 0 to 2): description of “TDCVF Flag”, corrected
23	1583	23.4.4.2 RSCFDnCFDGCTR — Global Control Register: access and address corrected
23	1585	23.4.4.3 RSCFDnCFDGSTS — Global Status Register: access and address corrected
23	1587	23.4.4.4 RSCFDnCFDGERFL — Global Error Flag Register: access and address corrected
23	1589	23.4.4.5 RSCFDnCFDGTSC — Global Timestamp Counter Register: access and address corrected
23	1590	23.4.4.6 RSCFDnCFDGTINTSTS0 — Global TX Interrupt Status Register 0: access and address corrected
23	1593	23.4.4.7 RSCFDnCFDGFDCFG — Global FD configuration register: access and address corrected

Revision 1.20 History, Date Oct 26, 2015 (11/21)

Section	Page	Summary
23	1594	23.4.4.8 RSCFDnCFDGCRCCFG — Global CRC configuration register: access and address corrected
23	1595	23.4.5.1 RSCFDnCFDGAFLECTR — Receive Rule Entry Control Register: access and address corrected
23	1596	23.4.5.2 RSCFDnCFDGAFLCFG0 — Receive Rule Configuration Register 0: access and address corrected
23	1602	23.4.5.5 RSCFDnCFDGAFLP0_j — Receive Rule Pointer 0 Register (j = 0 to 15): access and address corrected
23	1605	23.4.6.1 RSCFDnCFDRMNB — Receive Buffer Number Register: access and address corrected
23	1611	23.4.6.5 RSCFDnCFDRMFDSTSq — Receive Buffer CAN FD Status Register (q = 0 to 47): access and address corrected
23	1613	23.4.7.1 RSCFDnCFDRFCCx — Receive FIFO Buffer Configuration and Control Register (x = 0 to 7): access and address corrected
23	1615	23.4.7.2 RSCFDnCFDRFSTSx — Receive FIFO Buffer Status Register (x = 0 to 7): access and address corrected
23	1617	23.4.7.3 RSCFDnCFDRFPCTRx — Receive FIFO Buffer Pointer Control Register (x = 0 to 7): access and address corrected
23	1622	23.4.7.6 RSCFDnCFDRFFDSTSx — Receive FIFO CAN FD Status Register (x = 0 to 7): access and address corrected
23	1628	23.4.8.2 RSCFDnCFDCFSTSk — Transmit/receive FIFO Buffer Status Register (k = 0 to 8): access and address corrected
23	1631	23.4.8.3 RSCFDnCFDCFPCTRk — Transmit/receive FIFO Buffer Pointer Control Register (k = 0 to 8): access and address corrected
23	1637	23.4.8.6 RSCFDnCFDCFFDCSTSk — Transmit/Receive FIFO CAN FD Configuration/ Status Register (k = 0 to 8): access and address corrected
23	1640	23.4.9.1 RSCFDnCFDFESTS — FIFO Empty Status Register: access and address corrected
23	1642	23.4.9.2 RSCFDnCFDFFSTS — FIFO Full Status Registerr: access and address corrected
23	1644	23.4.9.3 RSCFDnCFDFMSTS — FIFO Message Lost Status Register: access and address corrected
23	1646	23.4.9.4 RSCFDnCFDRFISTS — Receive FIFO Buffer Interrupt Flag Status Register: access and address corrected
23	1647	23.4.9.5 RSCFDnCFDCFRISTS — Transmit/receive FIFO Buffer Receive Interrupt Flag Status Register: access and address corrected
23	1648	23.4.9.6 RSCFDnCFDCFTISTS — Transmit/receive FIFO Buffer Transmit Interrupt Flag Status Register: access and address corrected
23	1655	23.4.10.4 RSCFDnCFDTMPTRp — Transmit Buffer Pointer Register (p = 0 to 47): access and address corrected
23	1657	23.4.10.5 RSCFDnCFDTMFDCTRp — Transmit Buffer CAN FD Configuration Register (p = 0 to 47): access and address corrected
23	1670	23.4.12.1 RSCFDnCFDTXQCCm — Transmit Queue Configuration and Control Register (m = 0 to 2): access and address corrected
23	1672	23.4.12.2 RSCFDnCFDTXQSTSm — Transmit Queue Status Register (m = 0 to 2): access and address corrected
23	1674	23.4.12.3 RSCFDnCFDTXQPCTRm — Transmit Queue Pointer Control Register (m = 0 to 2): access and address corrected
23	1675	23.4.13.1 RSCFDnCFDTHLCCm — Transmit History Configuration and Control Register (m = 0 to 2): access and address corrected
23	1677	23.4.13.2 RSCFDnCFDTHLSTSm — Transmit History Status Register (m = 0 to 2): access and address corrected

Revision 1.20 History, Date Oct 26, 2015 (12/21)

Section	Page	Summary
23	1679	23.4.13.3 RSCFDnCFDTHLPCTRm — Transmit History Pointer Control Register (m = 0 to 2): access and address corrected
23	1682	23.4.14.1 RSCFDnCFDGTSTCFG — Global Test Configuration Register: access and address corrected description of “RTMPS[6:0] Bits”, corrected (additional description)
23	1684	23.4.14.2 RSCFDnCFDGTSTCTR — Global Test Control Register: access and address corrected
23	1685	23.4.14.3 RSCFDnCFDGLOCKK — Global Lock Key Register: access and address corrected
23	1702	23.8.1 Transmit Priority Determination: description corrected
23	1708	23.9 Gateway Function: description corrected
23	1711	23.10.5 RAM Test: description corrected (additional description)
23	1715	23.11.1.2 Bit Timing Setting: description corrected
23	1716	Figure 23.17 Bit Timing Chart: corrected (“TSEG1 > TSEG2 ≥ SJW” → “TSEG1 ≥ TSEG2 ≥ SJW”)
23	1723	Figure 23.22 SSP timing: note added
23	1723	23.11.1.6 Transceiver delay compensation (Only in CAN FD Mode): description corrected
23	1746	23.12.5 ECCRCFDnTMC — RSCFDn ECC Test Mode Control Register: access corrected
23	1749	23.12.7 ECCRCFDnTRC — RSCFDn ECC Redundant Bit Data Control Test Register: R/W of bit 7 in the bit chart, corrected (“R/W” → “R”)
23	1750	23.12.8 ECCRCFDnSYND — RSCFDn ECC Decode Syndrome Data Register: access corrected
23	1750	23.12.9 ECCRCFDnHORD — RSCFDn ECC 7-Bit Redundant Bit Data Hold Test Register: access corrected
23	1751	23.12.10 ECCRCFDnECRD — RSCFDn ECC Encode Test Register: access corrected
23	1752	23.12.11 ECCRCFDnERDB — RSCFDn ECC Redundant Bit Input/Output Replacement Buffer Register: section title corrected
24	1758	Table 24.5 Reset sources: corrected (“ETNBn” → “ETNB0”)
24	1763	Table 24.9 Configuration of E-MAC-related Registers: corrected (register name: “MAC” → “E-MAC”), and “ETNBn Delayed collision detect counter register” deleted entire row
24	1793	Table 24.26 TGC register contents (2/2): corrected (“Non-ABV mode” → “Non-AVB mode”)
24	1801	Table 24.32 CIVRc register contents: corrected (“(0 to H’FFFF)” → “(1 to H’FFFF)”)
24	1855	24.3.58 EthernetAVB Mode Register (GECMR): “CAUTION” corrected
24	1856	24.3.59 E-MAC Address High Register (MAHR): corrected (“MAC” → “E-MAC”)
24	1856	Table 24.68 MAHR register contents: corrected (“MAC” → “E-MAC”)
24	1857	24.3.60 E-MAC Address Low Register (MALR): corrected (“MAC” → “E-MAC”)
24	1857	Table 24.69 MALR register contents: corrected (“MAC” → “E-MAC”)
24	—	24.3.62 Delayed Collision Detect Counter Register (CDCR): deleted
24	1867	24.4.1 AVB-DMAC Operating Modes: description corrected
24	1869	Figure 24.6 Flow for Transitions of Operating Mode (Other than from Operation Mode to Configuration Mode): corrected
24	1872	(1) Initializing the Receiver Section: description corrected
24	1881	Figure 24.14 Example of URAM Memory Map: corrected (“Storage element” added)
24	1884	24.4.3.4 Descriptor Chain Processing: description corrected
24	1887	Table 24.78 Summary of Descriptor Types: description on FEMPTY_IC, corrected
24	1893	(7) How to Use Chain Control Descriptors: description corrected

Revision 1.20 History, Date Oct 26, 2015 (13/21)

Section	Page	Summary
24	1896	24.4.4.1 Reception Queues: description corrected
24	1899	(2) Separating Streams: corrected (“ABV network” → “AVB network”, “22nd byte” → “23rd byte”)
24	1906	(a) Setting Up an Incremental Data Area: description corrected
24	1908	(4) Support for Reception Time Stamps: description corrected
24	1911	(a) Support for Traffic Classes and Associated Priority: description corrected
24	1916	(3) Setting the Size of the Transmission FIFO: description corrected
24	1919	(1) Transmitting Frames: description corrected
24	1920	Figure 24.35 Software Flow for Immediate Frame Transmission: corrected
24	1924	Figure 24.41 Mechanism to Support Transmission Time Stamps: corrected (“Frame 1 data in local memory with the TAG and TSR values.” → “Frame 2 data in local memory with the TAG and TSR values.”)
24	1927	24.4.6 CBS (Credit-Based Shaping): corrected (“being clear (0)” deleted)
24	1930	24.4.6 CBS (Credit-Based Shaping): description of “Example”, corrected
24	1934	24.4.6.3 Example: description corrected
24	1935	24.4.7.1 gPTP Timer: description corrected
24	1936	24.4.7.3 Synchronization with the Grandmaster Clock: description corrected
24	1938	24.4.8 Support for IEEE 1722: corrected (“pPTP time” → “gPTP timer”, “(TCCR.TCSS)” → “(GCCR.TCSS)”)
24	1939	24.4.9 Flow Control: description corrected (additional description)
24	1940	24.4.10 Magic Packet Detection: description corrected, and “NOTE” added
24	1941	Table 24.89 EthernetAVB Interrupts: corrected (EIINT Interrupt Channel Number) Other management (FIFO caution level, etc.) interrupt: “247” → “246” E-MAC interrupt: “248” → “247”
24	1943	Figure 24.49 Flow of E-MAC Initialization (for AVB Mode and Full-Duplex Operation): corrected (figure title, “ABV Mode” → “AVB Mode”(description))
24	1944	Figure 24.50 Flow of AVB-DMAC Initialization (for AVB Mode and Full-Duplex Operation): corrected (figure title, “ABV Mode” → “AVB Mode”(description))
26	1970	Table 26.2 Index: corrected (““n” (n = 0,1)” → ““n””)
26	1970	26.1.2 Register Base Addresses: section title corrected
26	1970	Table 26.3 Register Base Address: table title corrected
26	1970	Table 26.4 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
26	1974	Table 26.9 WDTA start-up options: fucntion and description on OPWDOVF[2:0], corrected
26	1977	26.3.2 WDTAnWDTE — WDTA Enable Register: value after reset of bit 7 in the bit chart, corrected (“0/1” added)
26	1977	Table 26.11 WDTAnWDTE Register Contents: functional description corrected
26	1979	26.3.3 WDTAnEVAC — WDTA Enable VAC Register: value after reset of bit 7 in the bit chart, corrected (“0/1” added)
26	1979	Table 26.14 WDTAnEVAC Register Contents: functional description corrected
26	1986	26.5.2.1 Calculating an Activation Code when the VAC Function is Used: “NOTE” corrected
27	1990	Table 27.4 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
27	1992	Figure 27.1 Block diagram of OSTM: figure title corrected (“OS Timer” → “OSTM”)
27	1993	27.2.4 Output Modes: description corrected (“OS Timer” → “OSTM”)
27	1996	27.3.2 OSTMnCMP - OSTMn Compare Register: section title corrected (“OSTM” → “OSTMn”)
27	1998	27.3.4 OSTMnTO - OSTMn Output Register: section title corrected (“OSTM” → “OSTMn”)

Revision 1.20 History, Date Oct 26, 2015 (14/21)

Section	Page	Summary
27	1998	27.3.5 OSTMnTOE - OSTMn Output Enable Register: section title corrected ("OSTM" → "OSTMn")
27	1999	Table 27.14 OSTMnTE Register Contents: functional description on OSTMnTE, corrected ("OSTMnTT.OSTMnTTF" → "OSTMnTT.OSTMnTT")
27	1999	27.3.6 OSTMnTE — OSTMn Count Enable Status Register: "NOTE" corrected
27	2000	Table 27.15 OSTMnTS Register Contents: functional description on OSTMnTS, corrected
27	2000	Table 27.16 OSTMnTT Register Contents: functional description on OSTMnTT, corrected
27	2002	27.3.10 OSTMnEMU — OSTMn Emulation Register: value after reset corrected
27	2002	Table 27.18 OSTMnEMU Register Contents: functional description on OSTMnSVSDIS, corrected
27	2004	27.4.2.1 Basic operation in interval timer mode: description of "Forced restart", corrected ("OS Timer" → "OSTM")
27	2005	27.4.2.1 Basic operation in interval timer mode: description of "(2)", corrected
27	2006	27.4.2.2 Operation when OSTMnCMP = 0000 0000_H: description of "(4)", corrected
27	2008	27.4.3.1 Basic operation in free-run compare mode: description of "OSTMnTINT period", corrected
27	2009	27.4.3.2 Operation when OSTMnCMP = 0000 0000_H: description of "(4)", corrected
29	2028	Table 29.4 TAUBn Clock Supply: corrected ("Clock for the Unit" → "Unit Clock Name")
29	2036	29.2.1 Features Summary: description of "Independent and synchronous operation", corrected
29	2037	29.2.2 Terms: description of "Trigger selector", corrected
29	2042	Table 29.10 List of Registers: address corrected ("<TAUBn_base> + m × 4_H" → "<TAUBn_base> + 0_H + m × 4_H")
29	2048	Table 29.15 TAUBnCMORm Register Contents (1/3): functional description on TAUBnCKS [1:0], corrected
29	2051	29.3.3.4 TAUBnCMURm — TAUBn Channel Mode User Register: value after reset corrected ("0000_H" → "00_H")
29	2052	29.3.3.6 TAUBnCSCm — TAUBn Channel Status Clear Register: access corrected ("0000_H" → "00_H")
29	2053	Table 29.20 TAUBnTE Register Contents: functional description corrected
29	2055	Table 29.23 TAUBnRDS Register Contents: corrected (table title) ("TAUBnRDM" → "TAUBnRDS")
29	2057	29.3.4.6 TAUBnRSF — TAUBn Channel Reload Status Register: description corrected
29	2063	29.3.7.1 TAUBnEMU — TAUBn Emulation Register: section title corrected ("TAUB" → "TAUBn")
29	2063	Table 29.35 TAUBnEMU Register Contents: corrected (table title) ("TTAUBn" → "TAUBn")
29	2065	29.5.1 Rules of Synchronous Channel Operation Function: description of "Operation clock", corrected
29	2066	29.5.2.1 Simultaneous Start and Stop within the Same Unit: section title corrected
29	2068	29.6.2.1 Initial Settings: reference description corrected
29	2069	29.6.2.2 Start Counter and Count Operation: corrected ("TAUBnRSF.RSFm" → "TAUBnRSF.TAUBnRSFm")
29	2069	29.6.3 Other General Rules of Simultaneous Rewrite: description corrected ("TAUBnRDE.RDEm" → "TAUBnRDE.TAUBnRDEm") ("TAUBnRDS.RDSm" → "TAUBnRDS.TAUBnRDSm") ("TAUBnRDM.RDMm" → "TAUBnRDM.TAUBnRDMm") ("TAUBnRDC.RDCm" → "TAUBnRDC.TAUBnRDCm") ("TAUBnTOL.TOLm" → "TAUBnTOL.TAUBnTOLm") ("TAUBnRDC.RDCm" → "TAUBnRDC.TAUBnRDCm")

Revision 1.20 History, Date Oct 26, 2015 (15/21)

Section	Page	Summary
29	2077	Figure 29.8 General Procedure for Specifying a TAUBTTOUTm Channel Output Mode: figure titile corrected
29	2079	Figure 29.9 Set/Reset Conditions for Synchronous Channel Output Mode 2 with Dead Time Output: corrected (“TAUBnTDL.TAUBnTDL” → “TAUBnTDL.TAUBnTDLm”)
29	2079	29.7.3.3 Synchronous Channel Output Mode 2 with Dead Time Output: corrected (“TAUBnTDM.TAUBnTDMm” → “TAUBnTDE.TAUBnTDEm”)
29	2081	29.8.2 Event Count Mode: description corrected
29	2084	29.10.1 Example of Combination of TAUBTTINm Input Pulse Interval Measurement Function and TAUBTTINm Input Interval Timer Function: description corrected
29	2085	29.10.2 Example of Combination of TAUBTTINm Input Signal Width Measurement Function and Overflow Interrupt Output Function (during TAUBTTINm Width Measurement): corrected (“TAUDTTINm” → “TAUBTTINm”)
29	2091	Table 29.39 Contents of the TAUBnCMORm Register for Interval Timer Function: functional description on TAUBnMD0, corrected
29	2104	29.12.3.1 Overview: “NOTE” corrected
29	2108	Table 29.53 Operating Procedure for Clock Divide Function: description of “Status of TAUBn”, corrected (“During operation” and “Stop operation”)
29	2111	29.12.4.1 Overview: description corrected
29	2112	29.12.4.3 Block Diagram and General Timing Diagram: section title corrected
29	2117	29.12.5.1 Overview: description of “Summary”, corrected
29	2133	29.12.7.3 Block Diagram and General Timing Diagram: corrected (“TAUBnCSRm.OVF” → “TAUBnCSRm.TAUBnOVF”)
29	2135	Table 29.71 Simultaneous Rewrite Settings for TAUBTTINm Input Signal Width Measurement Function: corrected (“(TAUBnRDE.RDEm = 0)” → “(TAUBnRDE.TAUBnRDEm = 0)”)
29	2141	29.12.8.1 Overview: “NOTE” corrected
29	2146	29.12.9.1 Overview: corrected (“(TAUBnCDRn + 1)” → “(TAUBnCDRm + 1)”)
29	2156	29.12.11.1 Overview: corrected (“(TAUBnTS.TSm)” → “(TAUBnTS.TAUBnTSm)”)
29	2165	29.12.13.1 Overview: corrected (“TAUBnCMURm.TIS[1:0]” → “TAUBnCMURm.TAUBnTIS[1:0]”)
29	2169	Table 29.96 Operating Procedure for Overflow Interrupt Output Function (during TAUBTTINm Input Period Count Detection): description of “Status of TAUBn”, corrected (“During operation” and “Stop operation”)
29	2176	Table 29.101 Operating Procedure for Simultaneous Rewrite Trigger Generation Function Type 1: description of “Status of TAUBn”, corrected (“Start operation” and “Stop operation”)
29	2179	29.14.1.3 Block Diagram and General Timing Diagram: “NOTE” corrected
29	2180	Table 29.102 Contents of the TAUBnCMORm Register for the Master Channel of the PWM Output Function: functional description on TAUBnCKS[1:0], corrected corrected (“CKS[1:0]” → “TAUBnCKS[1:0]”)
29	2182	Table 29.105 Contents of the TAUBnCMORm Register for the Slave Channel of the PWM Output Function: functional description on TAUBnCKS[1:0], corrected corrected (“CKS[1:0]” → “TAUBnCKS[1:0]”)
29	2184	Table 29.109 Operating Procedure for PWM Output Function: description of “Status of TAUBn”, corrected (“During operation”)
29	2187	(3) Operation stop and restart: section title corrected
29	2189	29.14.2.1 Overview: description of “Description”, corrected
29	2190	29.14.2.1 Overview: “NOTE” corrected
29	2193	Table 29.110 Contents of the TAUBnCMORm Register for the Master Channel of the One-Shot Pulse Output Function: functional description on TAUBnCKS[1:0], corrected corrected (“CKS[1:0]” → “TAUBnCKS[1:0]”)

Revision 1.20 History, Date Oct 26, 2015 (16/21)

Section	Page	Summary
29	2195	Table 29.115 Control Bit Settings for Independent Channel Output Mode 2: description of “Setting” (“TAUBnTDL.TAUBnTDLm”)
29	2196	Table 29.117 Operating Procedure for One-Shot Pulse Output Function: description of “Status of TAUBn”, corrected (“During operation”)
29	2200	(4) TAUBnCMORm.TAUBnMD0 = 1: description corrected
29	2203	29.14.3.1 Overview: description corrected
29	2209	Table 29.121 Contents of the TAUBnCMORm Register for the Slave Channel 1 of the Delay Pulse Output Function: functional description on bit 13 (Reserved) and bit 5 (Reserved), corrected
29	2210	Table 29.122 Contents of the TAUBnCMURm Register for the Slave Channel 1 of the Delay Pulse Output Function: functional description on bit 7 to bit 2 (Reserved), corrected
29	2215	Table 29.132 Operating Procedure for Delay Pulse Output Function (2/2): description of “Operation”, corrected (“Start operation”) corrected (“TAUBnTS.TSm” → “TAUBnTS.TAUBnTSm”)
29	2228	Table 29.141 Operating Procedure for Triangle PWM Output Function: description of “Status of TAUBn”, corrected (“During operation”) corrected (“TAUBnTTOUTm” → “TAUBTTOUTm”)
29	2240	Table 29.151 Contents of the TAUBnCMORm Register for the Slave Channel 3 of the Triangle PWM Output Function with Dead Time: functional description on TAUBnCCS0, corrected (“00_B” → “0_B”)
29	2242	Table 29.155 Operating Procedure for Triangle PWM Output Function with Dead Time: description of “Status of TAUBn”, corrected (“Start operation”)
29	2244	(3) TAUBTTOUTm (slave 2) = 0% and TAUBTTOUTm (slave 3) ≥ 0%: section title corrected corrected (“> 0%” → “ ≥ 0%”)
29	2251	(6) Inhibited INTTAUBnIm to set TAUBTTOUTm negative phase period: description corrected
29	2251	(7) Slave 2 TAUBnCDRm = 0000_H (Duty cycle = 100%): section title corrected
29	2251	Figure 29.110 Slave 2 TAUBnCDRm = 0000_H (Duty cycle = 100%): figure title corrected
30	2254	Table 30.4 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
30	2256	30.2.1 Functional Overview: description of “Independent and synchronous operation”, corrected
30	2258	30.2.6 Description of Blocks: description of “Prescaler block”, corrected
30	2259	30.2.6 Description of Blocks: description of “Clock and count selection”, corrected
30	2268	Table 30.15 TAUJnCMORm Register Contents (1/3): functional description on TAUJnCCS[1:0], corrected
30	2272	30.3.3.5 TAUJnCSRm — TAUJn Channel Status Register: value after reset of bit 1 in the bit chart, corrected (“0” → “×”)
30	2272	Table 30.17 TAUJnCSRm Register Contents: corrected bit position (“7 to 1” → “7 to 2”), and bit 1 added
30	2293	30.8.2 Other Operating Modes: description corrected
30	2294	Figure 30.9 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJnMD0 = 0): figure title corrected (“TAUJMD0” → “TAUJnMD0”)
30	2294	Figure 30.10 INTTAUJnIm Generation Timing (when TAUJnCMORm.TAUJnMD0 = 1): figure title corrected (“TAUJMD0” → “TAUJnMD0”)
30	2297	30.12.1.1 Overview: corrected (“TAUJnTTOUTm” → “TAUJTTOUTm”)
30	2302	(2) TAUJnCDRm = 0000 0000_H, count clock = PCLK: corrected (“TAUJnTTOUTm” → “TAUJTTOUTm”)
30	2305	30.12.2.3 Block Diagram and General Timing Diagram: corrected (“(TAUJnCMORm.TAUJnMD0 = 0)” → “(TAUJnCMORm.TAUJnMD0 = 1)”)
30	2319	30.12.4.1 Overview: corrected (“(TAUJnCSRm.OVF = 1)” → “TAUJnCSRm.TAUJnOVF”)

Revision 1.20 History, Date Oct 26, 2015 (17/21)

Section	Page	Summary
30	2327	30.12.5.1 Overview: description of “Description”, corrected
30	2335	Table 30.60 Operating Procedure for TAUJTTINm Input Period Count Detection Function: description of “Status of TAUJn”, corrected (“During operation”)
30	2347	30.13.1.3 Block Diagram and General Timing Diagram: “NOTE” corrected
31	2355	Table 31.4 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
31	2359	Table 31.8 Registers: register name, symbol, address of “sub-count register read buffer” and “clock error correction register”, corrected
31	2365	Table 31.13 RTCAnSRBU Register Contents: functional description on RTCAnSRBU[21:0], corrected
31	2377	31.3.4.9 RTCAnDAYC — RTCA Day of the Month Count Register, “NOTE” corrected
31	2390	Table 31.38 RTCAnEMU Register Contents: functional description on RTCAnSVSDIS, corrected
31	2392	31.4.2 Fixed Interval Interrupt Function: description corrected
31	2393	31.4.3 Alarm Interrupt Function: description corrected
31	2398	31.5.1.2 RTCA Initialization Procedure: “CAUTION” corrected
31	2399	31.5.2 Updating Clock Counters: description of “1.” in “CAUTION”, corrected
31	2400	31.5.3.1 Procedure for Reading Count Buffer Registers: description corrected
31	2401	31.5.3.1 Procedure for Reading Count Buffer Registers: description of “1.” and “3.” in “CAUTION”, corrected
31	2402	Figure 31.8 Reading Clock Counter Registers: corrected
31	2408	31.6.3 Timing of Sub-Counter Read Buffer Reading while Counter is Enabled: description of “(3)” corrected
32	2409	32.1.1 Number of Units and Channels: description corrected
32	2410	Table 32.4 Register Base Addresses: table title corrected
32	2410	Table 32.5 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
32	2411	Table 32.9 Internal Input/Output Signals: unit signal name corrected (“PWGA_TRGOUT” → “PWGA_TRGOUT[11:0]”)
32	2412	32.1.8 Outline of Functions: description of “PWBA” and “PWGA”, corrected
32	2413	32.1.8 Outline of Functions: description of “PWGA” and “PWSA”, corrected
32	2415	Table 32.10 List of PMW-Diag Registers: corrected (“Register name”) (“PWGAnCRDR” → “PWGAnCTDR buffer register”)
32	2418	32.2.1.3 PWBAnTS Register: description corrected (“(n = 0 to 3).” → “(m = 0 to 3)”) R/W of bit 7 to bit 4 in the bit chart, corrected (“W” → “R”)
32	2419	32.2.1.4 PWBAnTT Register: R/W of bit 7 to bit 4 in the bit chart, corrected (“W” → “R”)
32	2420	32.2.1.5 PWBAnEMU Register: description corrected section title: “PWBA0EMU” → “PWBAnEMU” bit name of bit 7 in the bit chart: “PWBA0SVSDIS” → “PWBAnSVSDIS”
32	2420	Table 32.15 PWBAnEMU Register Contents: corrected table title: “PWBA0EMU” → “PWBAnEMU” bit name: “PWBA0SVSDIS” → “PWBAnSVSDIS”
32	2422	Table 32.18 PWGAnCSDR Register Contents: functional description on PWGAnCSDR[11:0], corrected
32	2422	Table 32.19 PWGAnCRDR Register Contents: functional description on PWGAnCRDR[12:0], corrected
32	2423	Table 32.20 PWGAnCTDR Register Contents: functional description on PWGAnCTDR[11:0], corrected
32	2423	Table 32.21 PWGAnCSBR Register Contents: functional description on PWGAnCSBR[11:0], corrected
32	2424	Table 32.22 PWGAnCRBR Register Contents: functional description on PWGAnCRBR[12:0], corrected

Revision 1.20 History, Date Oct 26, 2015 (18/21)

Section	Page	Summary
32	2424	Table 32.23 PWGAnCTBR Register Contents: functional description on PWGAnCTBR[11:0], corrected
32	2425	Table 32.25 PWGAnRDT Register Contents: functional description on PWGAnRDT, corrected
32	2427	32.2.1.18 PWSAnSTR Register: description corrected
32	2427	Table 32.28 PWSAnSTR Register Contents: functional description on PWSAnQNE, corrected
32	2428	32.2.1.19 PWSAnSTC Register: R/W of bit 7 to bit 2, corrected (“W” → “R”)
32	2429	32.2.1.21 PWSAnPVCRx_y (x = 00, 02, 04 ... 10, y = 01, 03, 05 ... 11) Register: address corrected
32	2429	Table 32.31 PWSAnPVCRx_y Register Contents: bit position corrected (“7 to 6” → “7, 6”)
32	2430	Table 32.32 PWSAnEMU Register Contents: functional description on PWSAnSVSDIS, corrected
32	2431	Figure 32.2 PWM-Diag Operating Procedure: corrected (“PWGA0” → “PWGAn”)
32	2432	32.3 Operating Procedure: description corrected
32	2432	Figure 32.3 Simultaneous Rewrite Procedure: corrected
37	2607	37.5.5.2 CKSC_IPLL0PIXS_ACT . PLL0PIXCLK clock active register (D1L2(H), D1M1(H) and D1M2(H) only): section title corrected (“(D1M2(H) only)” → “(D1L2(H), D1M1(H) and D1M2(H) only)”)
37	2630	37.5.5.25 CKSC_IVDCE0VIS_CTL — C_ISO_VI0PIXCLK source clock selection register (D1M1(H) and D1M2H only): section title corrected (“(D1M1(H) and D1M2(H) only)” → “(D1M1(H) and D1M2H only)”)
37	2637	37.6 Sprite Engine registers update control: description corrected
37	2637	Figure 37.13 Sprite Engine registers update control for video channel n: corrected
37	2637	Table 37.34 VDCEnVEm signals (1/2): corrected (“VSYNC” → “VE”)
37	2638	Table 37.34 VDCEnVEm signals (2/2): corrected (“VSYNC” → “VE”)
37	2638	37.6 Sprite Engine registers update control: description corrected
37	2639	Figure 37.14 Video channel 0 video input selections: “*2” and “Note 2.” deleted
37	2641	Figure 37.15 Video channel 1 video input selections: “*2” and “Note 2.” deleted
37	2643	Table 37.42 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
37	2678	37.8.5.5 VSYNC Timing from MIPI to VDCE: description of “MIPI setting”, corrected (“16” → “17”)
37	2691	Table 37.73 VDCECTL register contents: functional description on UPDT1 and UPDT0, corrected (“VSYNC1” → “VUPDATE1”, “VSYNC0” → “VUPDATE0”)
37	2693	Table 37.75 SPEAUPDEN register contents: functional description on UPDEN[15:0], corrected (“VE” → “VSYNC”)
38	2704	Table 38.8 Features of Video Display Controller (2/2): functional description on “Graphics”, corrected (“Y: 8 bits, Cb/Cr: 8 bits” → “Y: 8 bits, Cb: 8 bits, Cr: 8 bits; 24 bits”)
38	2707	Table 38.10 Input/Output Pins (Channel 1): corrected (“Figure 38.4” → “Table 38.10”)
38	2718	Figure 38.9 BT601 Horizontal Timing (625 Lines/50.00 Hz): corrected
38	2721	Figure 38.13 BT656 Horizontal Timing (625 Lines/50.00 Hz): corrected
38	2723	38.3.1.8 Typical Signal Timing of BT656 Format: “Timing Example of 525-Line Interface Input in BT656 Format”, added (remove from “38.3.1.10 BT656 Progressive Format”)
38	2723	Figure 38.15 Timing of Vertical/Horizontal Synchronization Signal from 525-Line Interlaced Input in BT656 Format (Top to Bottom), added (remove from “38.3.1.10 BT656 Progressive Format”)
38	2723	Figure 38.16 Timing of Vertical/Horizontal Synchronization Signal from 525-Line Interlaced Input in BT656 Format (Bottom to Top), added (remove from “38.3.1.10 BT656 Progressive Format”)

Revision 1.20 History, Date Oct 26, 2015 (19/21)

Section	Page	Summary
38	2725	Table 38.24 SAV/EAV Code in BT656 Format (625 Lines/50.00 Hz): value for “1725” and “1727” of “valid area”, corrected
38	2726	Table 38.25 SAV/EAV Code in BT656 Format (525 Lines/59.94 Hz): value for “1713” and “1715” of “valid area”, corrected
38	2727	(c) Timing Example of 525-Line Interface Input in BT656 Format: deleted (removed to the end of “38.3.1.8 Typical Signal Timing of BT656 Format”)
38	2728	Table 38.27 SAV/EAV Code in BT656 Progressive Format (525 Lines/59.94 Hz): value for “1713” and “1715” of “valid area”, corrected
38	2728	Figure 38.17 Vertical/Horizontal Synchronization Signal in BT656 Format (525 Lines, Progressive): note corrected
38	2730	“NOTE” added
38	2731	Figure 38.19 YCbCr Data Expansion for YCbCr422 Input: corrected
38	2739	Table 38.39 Video input by data enable: corrected description on DE_VLAST_EN: note added register name corrected (“DEMODE0” → “DEMODE1”)
38	2752	38.3.2.15 Video input with data enable control register 0 (DEMODE0): initial value and description of bit 31 to 5 (Reserved), corrected
38	2753	38.3.2.16 Video input with data enable control register 1 (DEMODE1): initial value and description of bit 31 to 11 (Reserved), corrected
38	2786	Table 38.84 Frame Buffer Control: description on GR_FLM_SEL: corrected (“SC_SCL0_FRC3” → “SC0_SCL1_WR1”)
38	2798	38.4.3.3 Missing Vsync Compensation Control Register (SC0_SCL0_FRC2): note 2. added
38	3022	Table 38.149 Signals Generated by LCD TCON: description and signal name on STVA/VS and STVA/HS, corrected
38	3029	Table 38.155 Vertical Panel Driver Signal Generation: description on TCON_STVA_VS[10:0] and TCON_STVB_VS[10:0], corrected (“TCON_STVA_HS” → “TCON_STVA_VS”, “TCON_STVB_HS” → “TCON_STVB_VS”)
38	3031	Table 38.157 Panel Driver Signal Output Selection: description on TCON_STVA_SEL[2:0] and TCON_STH_SEL[2:0, corrected] (“VS” → “STVA/VS”, “HSS” → “STH/SP/HS”)
38	3076	38.8.2.35 TCON Vertical Timing Setting Register A2 (TCON_TIM_STVA2): description on TCON_STVA_SEL[2:0], corrected (“VS” → “STVA/VS”)
38	3080	38.8.2.39 TCON Horizontal Timing Setting Register STH2 (TCON_TIM_STH2): description on TCON_STH_SEL[2:0], corrected (“HS” → “STH/SP/HS”)
39	3112	Table 39.3 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
40	3134	Table 40.11 VOCAnCTL register contents (1/2): bit name (*3 added) and functional description on VOCAnMKVOC, corrected
40	3135	Table 40.11 VOCAnCTL register contents (2/2): functional description on VOCAnCL16 and VOCAnCLm, corrected, and note 3. added
40	3135	note 3. added concerning table 40.11 VOCAnCTL register contents.
41	3166	Table 41.19 Pixel Format: corrected (“CMDBT” → “CMPBT”)
41	3167	41.5.6 CRC Calculation Time Period and Comparison Timing: reference number corrected (“Figure 43.4” → “Figure 41.5”)
43	3188	Table 43.5 SPEA internal signal connections: corrected (“VSYNC” → “VUPDATE”)
43	3190	Figure 43.1 RLE Engines block diagram: corrected (“VSYNC” → “VUPDATE”)
43	3193	Table 43.6 RLE Units supported color formats: table title corrected
43	3195	Table 43.7 RLE Targa packets: table title corrected
43	3198	43.3.5 RLE definition registers modification: corrected (“VSYNC” → “VUPDATE”)
43	3199	Figure 43.9 Sprite Units block diagram: corrected (“VSYNC” → “VUPDATE”)

Revision 1.20 History, Date Oct 26, 2015 (20/21)

Section	Page	Summary
43	3201	(b) SPEAnSkLYm.SPEAnSkLYHm: height: description of “Image Synthesizer line offset”, corrected (“8192 KB” → “8192 Byte”)
43	3203	43.4.7 Sprite definition registers modification: corrected (“VSYNC” → “VUPDATE”)
43	3204	Table 43.8 Sprite and RLE Units registers overview: register name on SPEAnSkVDm (“VSYNC” → “VUPDATEn”)
43	3210	Table 43.14 SPEAnRUP register contents: functional description on SPEAnRUP1 and SPEAnRUP0, corrected (“VSYNC” → “VUPDATE”)
43	3214	Table 43.18 SPEAnSkUP register contents: functional description on SPEAnSkUP1 and SPEAnSkUP0, corrected (“VSYNC” → “VUPDATE”)
43	3214	“NOTE” corrected (“VSYNC” → “VUPDATE”)
43	3216	43.5.2.5 SPEAnSkVDm - Sprite Unit k sprite m VUPDATEn selection register: section title, description, and functional description, corrected (“VSYNC” → “VUPDATE”)
44	3239	44.3.20 JPEG Interface Compression Line Offset Register (JIFESOFST): references corrected
44	3262	44.4.2.1 Overview of Processing: description of “4.”, corrected (“JIFECNT” → “JIFDCNT”)
44	3265	(a) Error Marker: references corrected (“table 41.3” → “Table 44.43”)
44	3266	(b) Huffman Coded Segment Error: references corrected (“Table 41.4” → “Table 44.44”)
44	3270	44.4.3.7 Swap: description corrected (“JIFECNT” → “JIFDCNT”)
45	3281	Figure 45.3 Usage Example of Virtual Register: figure corrected, and description of “note 3.” corrected (“register m” → “register j”)
45	3282	Table 45.11 Register Address List (1/2): register name corrected (“Scan Group x status register” → “Scan Group status register”)
45	3284	Table 45.12 ADCEnVCRj register contents: functional description on GCTRL[5:0], corrected
45	3284	45.3.2.1 ADCEnVCRj — Virtual Channel Register j: “CAUITION” corrected
45	3288	45.3.2.4 ADCEnDRj — Data Register j: description corrected (“ADCEnVCRm” → “ADCEnVCRj”)
45	3289	Table 45.17 ADCEnDIRj register contents: table title (“ADCEnDIRm” → “ADCEnDIRj”) and functional description on ID[5:0], corrected
45	3291	Table 45.19 ADCEnPWDDIR Register Contents: functional description on WFLG and ID[5:0], corrected
45	3295	45.3.3.2 ADCEnADCR — A/D Control Register: “NOTE” corrected (“AD channel conversion” → “A/D channel conversion”)
45	3299	45.3.4.2 ADCEnULLMTBR0 to 2 — Upper Limit/Lower Limit Table Register 0 to 2: access corrected
45	3299	45.3.4.2 ADCEnULLMTBR0 to 2 — Upper Limit/Lower Limit Table Register 0 to 2: “CAUTION” corrected
45	3301	Table 45.28 ADCEnULER register contents (1/2): functional description on ULSG[1:0], corrected
45	3302	Table 45.28 ADCEnULER register contents (2/2): functional description on ULECAP[5:0], corrected
45	3315	45.3.5.11 ADCEnSGSTR — Scan Group Status Register: section title corrected
45	3330	Figure 45.6 Flow at the End of A/D Conversion: corrected, and note added
45	3342	45.4.11.1 Stopping Scan Group by ADHALT: description corrected (“(AD force halt trigger)” → “(A/D force halt trigger)”)
45	3342	45.4.12 Scan End Interrupt Request: description corrected (“ADCEnVCRm” → “ADCEnVCRj”)
45	3346	45.5.2 Diagnostic of Channel Multiplexer: description corrected
45	3350	45.7.1 Range of Channel Input Voltage: added
45	3351	45.7.2 Usage Notes for Analog Input Pins: description corrected

Revision 1.20 History, Date Oct 26, 2015 (21/21)

Section	Page	Summary
46	3446	46.6.6 ECCISMnTRC — ISMn RAM ECC Redundant Bit Data Control Test Register: R/W of bit 7 in the bit chart, corrected (“R/W” → “R”)
48	3456	Figure 48.1 Error Control Module block diagram: corrected (“OSTM1” → “OSTM0”, “OSTM1TTOUT” → “OSTM0TTOUT”)
48	3459	48.1.11 ECM Master - Checker operation: description corrected
48	3464	48.2.4 ECM Master/Checker Error Source Status Register 0 (ECMmESSTR0, m = M/C): corrected (“regsiter” → “register”)
48	3482	48.2.23 ECM Delay Timer Register (ECMDTMR): description corrected (additional description)
48	3483	48.2.24 ECM Delay Timer Compare Register (ECMDTMCMP): corrected (“ECMmSSE129n” → “ECMmSSE129”)
49	3489	Table 49.3 Clock Supply: corrected (“Clock for the Unit” → “Unit Clock Name”)
49	3495	Table 49.8 DCRAnCTL Register Contents: functional description on DCRAnPOL, corrected
49	3496	49.3.4 DCRAnCTL — CRC Control Register: “CAUTION” corrected
51	3499	(1) Debug interface: description corrected
51	3499	(2) Debug monitoring function: description corrected
51	3500	Table 51.4 Modules with optional peripheral break: “Control register bit” on PWM Generators and Diagnostic (PWM-Diag), corrected (“PWBA0EMU.PWBA0SVSDIS” → “PWBAnEMU.PWBAnSVSDIS”)
51	3501	(10) Timer function: description corrected
51	3501	(13) Security function: description corrected
51	3502	51.3.1.1 EPC — Emulation Peripheral Control Register: initial value of bit 7 in the bit chart, corrected (“0” → “0*1”), and note1. added
51	3502	51.3.1.1 EPC — Emulation Peripheral Control Register: description of “1.” in “NOTES”, corrected
51	3504	51.4.1 Treatment of devices using debugging: description corrected
52	3512	Table 52.5 Available Operations and Security Settings: “Serial programming” on Prohibition of programming commands, corrected (*1 added)
52	3512	note 1. added concerning table 52.5 Available Operations and Security Settings
52	3530	Table 52.23 CFERRINT_VCI2CFB/PE1_OS register contents: functional description on bit 31 to bit 2 (Reserved), corrected

Revision 2.01 History

This revision history list shows all modifications of

Rev. 2.01 compared to Rev. 1.20.

NOTE

The chapter and page numbers in the table below refer to the older Rev. 2.01 and thus may not be valid for the current revision.

Revision 2.01 History, Date Aug 26, 2016 (1/18)

Section	Page	Summary
1	4	D1M1A added
1	96	"Former products" section added
1	96	caution added: D1M1A features are preliminary
1	97	D1L devices' PLL1 frequency corrected: "fixed to 480 MHz" -> "max. 480 MHz"
1	98	"Error Control Module (ECM)" added
1	101	"RH850/D1M products overview" section split into D1M1 and D1M2 products overview subsections
1	101	D1M1 devices' PLL1 frequency corrected: "fixed to 480 MHz" -> "max. 480 MHz"
1	101	D1M1(H) devices' PLL0 frequency corrected: "480 MHz" -> "max. 480 MHz"
1	102	RS-CANFD modules removed from D1M1(H)
1	105	RS-CANFD modules removed from D1M1(H) block diagram
1	107	D1M1A added to overview table
1	107	D1M2 devices' PLL0 frequency corrected: "480 MHz" -> "max. 480 MHz"
1	107	D1M2 devices' PLL1 frequency corrected: "fixed to 480 MHz" -> "max. 960 MHz"
1	111	D1M1A block diagram added
1	111	note references for video channel 0 CLMA, MIPI, VI modules corrected (* -> *2)
1	112	part numbers for D1M1A added
1	112	D1M1(H) device variants with RS-CANFD removed from "Ordering information" table
2	113	"Pin Connection Diagrams" section rearranged
2	116	"D1M1A pin connection" section added
2	119	D1M1A port groups added to table
2	119	Port Group Index n = 22 added
2	119	number of D1M2 port groups corrected
2	121	expanded alternative mode selection by PFCAEn described
2	124	PFCAEn register added to table
2	132	correction: "PFCEn register" -> "PFECn and PFCAEn registers"
2	133	"Setting alternative functions" expanded with PFCAEn_m bits
2	133	correction: "PFCn register" -> "PFCn and PFCAEn registers"
2	134	"PFCAEn — Port Function Control Additional Expansion Register" added
2	135	addition: (the alternative input function that is set with PFCn.PFCn_m and PFCEn.PFCEn_m -> (the alternative input function that is set with PFCn.PFCn_m, PFCEn.PFCEn_m and PFCAEn_m)
2	147	correction: "...writing to the PDSCn and PINVn registers..." -> "...writing to the PDSCn, PODCn and PINVn registers..."
2	147	PFCAE added to "Pin-unit Register"
2	149	PFCAEn added in diagram

Revision 2.01 History, Date Aug 26, 2016 (2/18)

Section	Page	Summary
2	151	PFCAEn added in diagram
2	152	PFCAEn added in diagram
2	153	addtion: "PFCn_m and PFCEn_m" -> "PFCn_m, PFCEn_m and PFCAEn_m"
2	155	HyperBus Controller (HYPB0) pins added
2	155	NAND Flash Memory I/F A (NFMA0) pins added
2	168	section "D1M1A input buffers characteristics" added
2	172	"SDRA/SFMA I/O drive strength control Register (DSCTRL)" added
2	220	section "List of D1M1A Port Registers" added
2	229	PFCAE42 register added
2	242	HyperBus pins added as 3rd alternative of port 21
2	242	typo correction ("Medial Local Bus" -> "Media Local Bus")
2	242	SFMA0O[3:0]0 and SFMA0I[3:0]0 pins added to port group P21
2	243	section "Port 22 (P22)" added
2	267	D1M1A P22 pins states in normal operation mode added
2	268	D1M1A P22 pins states in debug mode added
2	270	D1M1A P22 pins states in serial programming mode added
2	272	PWRGD handling added for D1M1A
3	276	hypervisor removed from CPU protection functions
3	277	D1M1A added
3	278	typo corredction (busses -> buses)
3	278	typo corredction (busses -> buses)
3	279	HyperBus and OctaBus Controllers added in caution
3	279	hypervisor removed from CPU protection functions
3	280	typo correction (SNYCP -> SYNCP)
3	282	note 2 reference added to VCRE bit description
3	283	note 2 added
3	283	VCIE bit description completed
3	286	further explanation to SEGADDR regsiter description added
3	288	typo correction ("dependentregisters" -> "dependent registers")
3	288	typo correction (denpendent -> dependent)
3	291	correction: "In the initial state, notification of the ECC 2-bit error is enabled,..." -> "In the initial state, notification of the ECC 2-bit error is disabled,..."
3	304	D1M1A added
3	304	HyperBus Controller added to list
3	305	correction: all XC cache ECC error notifications are disabled initially
3	307	D1M1A added
3	307	caution corrected concerning hypervisor and supervisor mode
3	311	AXCERRADR.ADR[31:0] bit description corrected
3	313	"AXCCACHECMD — Cache command register" description corrected
3	324	D1M1A added
3	340	reference added regarding the procedure of system register update
3	341	"When switching Code Flash area:" added
3	341	description "When the updated results in the control registers and memories are reflected in the instruction fetch of a subsequent instruction:" modified
4	349	port P22 protection command and status registers added

Revision 2.01 History, Date Aug 26, 2016 (3/18)

Section	Page	Summary
5	368	D1M1A added
5	368	typo correction (dependend -> dependent)
5	370	D1M1A added
5	370	typo correction (dependend -> dependent)
5	371	"Cross-connect address map" section revised
5	371	correction: local RAM removed from all cross-connect address maps
5	372	typo correction (possiblity -> possibility)
7	378	typo correction ("e, setting of the IMRn register is reflected in the corresponding EIMK bit." removed)
7	381	typo correction (SNYCP -> SYNCP)
7	424	section "D1M1A Interrupt Exception Handler and Priority" added
7	428	NFMA0 interrupt added
7	445	correction: "Interrupt Response Times to be Considered" -> "Minimum interrupt Response Times to be Considered" and tables corrected
8	447	"Dual-address (2-cycle) transfer operation is adopted to DMA cycle." added
8	458	"In case that the continuous transfer enable bit (DTCTn.MLE) is set, ..." paragraph added
8	459	note 2 added
8	460	"DMA Transfer Error" section added
8	483	caution 1 modified
9	496	D1M1A added to "Note 3"
9	497	D1M1A added to "Note 4"
9	498	D1M1A added in figure title
9	498	D1M1A added in section title
9	502	addition: "In D1M1H, SDRB0RES bit corresponds ..." -> "In D1M1H and D1M1A, SDRB0RES bit corresponds ..."
9	510	addition: "In D1M1H, SDRB0RES bit ..." -> "In D1M1H and D1M1A, SDRB0RES bit ..."
9	510	addition: (D1M2(H), D1M1H only) -> "(D1M2(H), D1M1H, D1M1A only)
9	511	addition: (D1L2(H), D1M1(H), D1M2(H) only) -> (D1L2(H), D1M1(H), D1M2(H), D1M1A only)
9	511	addition: (D1M2(H) only) -> (D1M2(H), D1M1A only)
9	512	addition: "D1L1, D1L2(H), D1M1(H)" -> "D1L1, D1L2(H), D1M1(H), D1M1A"
9	513	addition: "D1Lx and D1M1(H)" -> "D1Lx, D1M1(H) and D1M1A"
9	515	resistor values removed from note 1 (are described in data sheet)
10	519	addition: "D1L1, D1L2(H) and D1M1(H)" -> "D1L1, D1L2(H), D1M1(H) and D1M1A"
10	519	D1M1A added to table
10	519	typo correction ("for for" -> "for")
10	520	D1M1A added in "Power-On-Clear" section
10	520	D1M1A added in "SDRAVCC, SDRAVSS" row
10	522	D1M1A added in diagram
12	530	D1M1A added in clock controller features summary
12	535	"D1M1A Clock Controller block diagram" section added
12	538	D1M1A added to "CPU and bus subsystems clock domains" section
12	540	note concerning header files names corrected
12	553	allowed PLL1 "Nr value" ranges corrected
12	553	typo corrections in "Nr value" (NI -> Nr)

Revision 2.01 History, Date Aug 26, 2016 (4/18)

Section	Page	Summary
12	553	PLLs Nr values for D1M1A added
12	553	typo correction ("Mr is determined by PLLkC.P[2:0]:" -> "Mr is determined by PLLkC.M[2:0]:")
12	554	PLLs Pr values for D1M1A added
12	558	write protection registers for "Isolated-Area clock control protection command register 2" corrected
12	562	typo corrections in MOSCCLKST[16:0] bit description (ROSCS.ROSCCLKST -> ROSCS.ROSCCLKACT)
12	566	typo corrections in SOSCCLKST[29:0] bit description (SOSCCLKST[16:0] -> SOSCCLKST[29:0])
12	572	PLL0.FVV[2:0] settings added for D1M1A device
12	575	PLL0.P[2:0] settings added for D1M1A device
12	576	PLL0.NI6:0] settings added for D1M1A device
12	576	typo correction (Inavalid -> Invalid)
12	579	D1M1A device added in PPL1C description
12	601	D1M1A added
12	602	D1M1A added
12	611	D1M1A added
12	612	D1M1A added
12	619	APB_CLK_RATIO regsiter access type corrected (R -> R/W)
12	637	typo correction ("31 to 1" -> "31 to 2")
12	638	typo correction ("31 to 1" -> "31 to 2")
12	639	D1M1A added
12	641	RSCANDCSID[2:0] description completed for D1M1A device
12	661	D1M1A added
12	661	D1M1A added
12	661	D1M1A added
12	662	D1M1A added
12	666	D1M1A added
12	666	D1M1A added
12	666	D1M1A added to "Base clocks maximum frequencies" table
12	667	D1M1A added to "CPU and buses subsystems clocks maximum frequencies" table
12	668	D1M1A added to "Always-On-Area clock domain clocks maximum frequencies" table
12	669	D1M1A added
12	670	C_ISO_LCBI removed from D1M1H and D1M2(H)
12	670	D1M1A added to "Isolated-Area clock domain clocks maximum frequencies" table
12	674	D1M1A added to CLMA units table
12	677	D1M1A added
12	677	note concerning header files names corrected
12	679	correction: "When CLMATMON stops or its frequency is lower than the limit*,..." -> "When CLMATMON's frequency is lower than the limit*,..."
12	679	note added concerning non-detection of completely stopped CLMAMON
12	690	D1M1A added
12	691	D1M1A added
13	695	Addtion in "I/O buffer hold state" section ("Further the P0 port buffers on the Always-On-Area, ...")

Revision 2.01 History, Date Aug 26, 2016 (5/18)

Section	Page	Summary
13	695	addtion in "I/O buffers during DEESPSTOP" ("Also the Always-On-Area port buffers of P0...")
13	695	typo correction (DeepSTOP -> DEEPSTOP)
13	706	DEEPSTOP mode preparation item added concerning stop of all active PLLs
14	709	D1M1A added
14	709	D1M1A added
14	710	NAND Flash Controller (NANDC) added
14	711	D1M1A added
14	712	D1M1A added
14	713	D1M1A added
14	713	typo correction ("...set the initial value..." -> "...set to the initial value...")
14	714	D1M1A added
14	714	correction: "...accesses is permitted..." -> "...accesses that are permitted..."
14	714	correction: "...accesses is permitted..." -> "...accesses that are permitted..."
14	715	D1M1A added
14	715	correction: "...accesses is permitted..." -> "...accesses that are permitted..."
14	715	correction: "...accesses is permitted..." -> "...accesses that are permitted..."
14	724	"D1M1A bus architecture" section added
14	733	D1M1A cross-connect bandwidth added
14	735	NANDC, HyperBus and OctaBus added in read access latency table
14	735	NANDC, HyperRAM and OctaRAM added in write access latency table
14	735	write access latency for Image Syntheziser 1 removed
14	735	correction: SDSB -> SDRA/SDRB
14	736	correction: SDSB -> SDRA/SDRB
14	737	"Bus Switch for HyperBus and OctaBus" section added
14	738	PBG2B guard added in diagram
14	740	"Data Flash Memory Control (DCIB)" added to PBUS0
14	741	PBG2A channel 8 PBUS guard added
14	741	PBG2B PBUS guard added
14	746	"XC Guard XCG15 (OctaBus)" added to PBG5B PBUS guard
14	746	correction in PBG5A-7 description: :SELB" -> "SELB (except SLPWGA0)"
14	747	"XC Guard XCG13 (HyperBus)" added to PBG5C PBUS guard
14	747	"XC Guard XCG13 (SFMA1)" added to PBG5C PBUS guard
14	748	document number reference for S/W User's Manual removed
14	761	" the own IPG and" removed in R0 bit description
14	761	" the own IPG and" removed in W0 bit description
14	762	addition: "ex) All corresponding registers for PBUS2 are not available in D1L1."
14	763	PBG2A PBUS guard channels corrected
14	763	PBG2B PBUS guard added
14	763	PBG5B PBUS guard channels corrected
14	763	PBG5C PBUS guard channels corrected
14	763	PBG0A PBUS guard channels corrected
14	764	typos in bitnames in FSGD0BPROTn register image corrected
14	770	modification: "Serial Flash Memory I/F (SFMA)" -> "Serial Flash Memory I/F 0 (SFMA0)"

Revision 2.01 History, Date Aug 26, 2016 (6/18)

Section	Page	Summary
14	770	correction: (n = 3 to 11) -> (n = 3 to 15)
14	770	XCG registers base addresses table completed
14	770	XCG Units table corrected
14	771	Cross-connect Guards error signals" table completed
15	803	D1M1A added
15	805	D1M1A added
15	806	"Data bus width and maximum bandwidth" added for D1M1A
15	824	correction: OFFSET[1:0] -> OFFSET[2:0]
15	824	ENOFFSET[3:0] values in bit description corrected
15	826	correction: "DBPDCNT0.ENOFFSET = 011B" -> "DBPDCNT0.ENOFFSET = 0011B"
15	831	typo correction ("The VRAM transaction restrictor..." -> "The SDR-SDRAM transaction restrictor...")
15	832	D1M1A added
15	833	D1M1A added
16	838	D1M1A added
16	864	"TRDPR[3:0] must be at least four times as large..." removed from TRDPR[3:0] bit description
16	867	correction: "TWRRD >= CWL + BL/2" -> "TWRRD >= CWL + BL/2 + tWTR"
17	912	D1M1A added in SFMA overview section
17	922	typo correction ("31 to 25" -> "28 to 25")
17	941	note added in CKDLYRX[2:0] description
17	946	correction: "Output delay nominal 2 ns" -> "Output delay minimum 2 ns"
17	946	correction: "Output delay nominal 2 ns" -> "Output delay minimum 2 ns"
17	959	"Output pin (CPHAT = 0)" signal in diagram corrected
17	960	"Output pin (CPHAT = 0)" signal in diagram corrected
17	961	typo correction (DRDMC -> DRDMCR)
17	961	typo correction (SMCMC -> SMDMCR)
17	969	correction: "fixed delay of nominal 2 ns" -> "fixed delay of minimum 2 ns"
17	969	note 2 added concerning SFMA1 clock frequencies
17	970	correction "fPHCLK/fBΦ = 4: set CKDLY.CKDLYOC[2:0] = 100B" -> "fPHCLK/fBΦ = 4: set CKDLY.CKDLYOC[2:0] = 101B"
17	971	"If the CKDLYRX[2:0] value reaches 000B ..." added in "Sampling point calibration prcoedure"
17	971	CKDLYOC[2:0] correction for fPHCLK/fBF = 4
17	971	note 1 added
17	972	"Sampling point calibration procedure flow" diagram added
18	978	correction: "...and data consistency checking" -> "...and data consistency checking is possible"
18	978	P42_12 added for CSIG3 data consistency check
18	1007	correction: "CSIGnCTL1.CSIGnCKR = 1" -> "CSIGnCTL1.CSIGnCKR = 0"
19	1033	CSIHnPE bit description corrected: "Writing to this bit is enabled when CSIHnCTL0.CSIHnPWR = 0." removed
19	1046	CSIHnIDx[2:0] bit description corrected
19	1047	CSIHnSPx[3:0] = 0000B added
19	1053	CSIHnSTR0.CSIHnHPST bit added to list of bits with write prohibition
19	1054	cautions for CSIHnRX0W and CSIHnRX0H modified

Revision 2.01 History, Date Aug 26, 2016 (7/18)

Section	Page	Summary
20	1118	typo correction (fulfil -> fulfill)
20	1140	typo correction ("transmission reception" -> "transmission/reception")
20	1149	paragraph "When response data of 9 bytes ..." repositioned
20	1150	correction: "...or reads the received data." -> "...or holds the received data."
20	1157	"Set IBS[1:0] bits to “00B” when UART buffer is not used." added
20	1160	typo correction ("RLN3nLRFC register" -> "RLN3nLDFC register")
20	1174	rephrasing: "Set the reception data" -> "Stores the received data"
20	1199	typo correction ("RLN3nBLFC register" -> "RLN3nLBFC register")
20	1208	typo corrections ("1010 0111" -> "1010 0111B", "1010 1000" -> "1010 1000B")
21	1231	typo correction (000 -> 000B)
21	1245	"When a restart condition is detected after a match ..." removed from GCA flag setting conditions
21	1280	typo correction in diagram ("7-bit slave address ICSAR1" -> "7-bit slave address RIICnSAR1")
21	1292	readability of text in the diagram improved
22	1298	D1M1(H) devices R7F701424 to R7F701427 removed from RSCAN Unit Configurations and Channels table
22	1298	D1M1(H) devices R7F701424 to R7F701427 removed from RSCAN units table
22	1298	D1M1A added in RSCAN units table
22	1299	typo correction ("... by the letter “y” (y = 0, 1)" -> "... by the index “y” (y = 0, 1)")
22	1299	r index number range corrected: (r = 0 to 29) -> (r = 0 to 63)
22	1323	RAM test page access registers 30 to 63 added in table
22	1331	rephrasing: "forcibly returns the state" -> "forcibly changes the state"
22	1335	BLF bit description corrected: "bus" -> "bus lock"
22	1335	correction: "If any of these flags is set to 0 at the same time..." -> "If any of these flags is set to 1 at the same time..."
22	1335	rephrase: "the flag bits are only set by the error event" -> "the flag bits are only set to 1 by the error event"
22	1335	typo correction ("were all 0 at the when time the error occurred." -> "at the time when the error occurred."
22	1335	typo correction (channe -> channel)
22	1337	correction: "(REC[7:0] or TEC[7:0] value > 127)" -> "((128 <= TEC[7:0] <= 255) or (128 <= REC[7:0]))"
22	1337	modifications in "BOEF Flag" description: "bus off state is reached" -> "bus off state is entered"
22	1340	repharsing: "the lowest transmit buffer number" -> "the transmit buffer with the smallest number"
22	1348	typo correction ("RCAN0TMSTSp register" -> "RSCAN0TMSTSp register")
22	1348	typo correction ("RCAN0TMSTSp register" -> "RSCAN0TMSTSp register")
22	1348	typo correction ("TMIE bit" -> "TMIEp bit")
22	1352	typo correction ("as the entire unit" -> "for the entire unit")
22	1352	typo correction ("in the entire unit" -> "for the entire unit")
22	1369	"This flag is 00H in global reset mode." added to RFMC[7:0] bits description
22	1369	RSCAN0RFSTSx register access type corrected
22	1371	typo correction ("is decremented" -> " is decremented by 1")
22	1376	CFITR bit description correction: pclk -> pclk/2
22	1378	typo correction (FIFIO -> FIFO)

Revision 2.01 History, Date Aug 26, 2016 (8/18)

Section	Page	Summary
22	1378	rephrasing: "on the identical channel" -> "on the same channel"
22	1382	correction: "The CFMLT flag..." -> "The CFFLL flag..."
22	1384	typo correction ("is decremented" -> " is decremented by 1")
22	1385	typo correction ("is incremented" -> " is incremented by 1")
22	1402	rephrasing: "arbitration loss" -> "arbitration-lost"
22	1421	correction: "(g + 1)-buffer transmit queue" -> "(g + 1) transmit queue"
22	1432	correction" "Do not access more than 96 bytes in the last page..." -> "Do not access more than 160 bytes in the last page..."
22	1435	typo corrections (<RCAN0_base> -> <RSCAN0_base>)
22	1436	correction: (r = 0 to 29) -> (r = 0 to 63)
22	1437	correction: "(in transmit mode, gateway mode)" -> "(in revceive mode, gateway mode)"
22	1444	correction: "CAN registers can be read," -> "Channel-related registers can be read,"
22	1444	corrections in "Channel Stop Mode" section: "GSLPR -> CSLPR
22	1452	obsolete sentence "The priority is determined by using one of the following methods." removed
22	1456	correction: "...timer is decremented by the next..." -> "...timer is decremented by 1 upon the next..."
22	1457	correction in table: "Buffer No. and "Buffer type" swapped
22	1461	correction: "Do not access more than 96 bytes..." -> "Do not access more than 160 bytes..."
22	1461	correction: (r = 0 to 29) -> (r = 0 to 63)
22	1471	correction: "...is incremented." -> "...is incremented by 1."
22	1472	typo correction in diagram ("RFMC[7:0], CFEMP, RFIF: Flags in the RSCAN0RFSTSx register" -> "RFMC[7:0], RFEMP, RFIF: Flags in the RSCAN0RFSTSx register")
22	1478	correction: "...is decremented." -> "...is decremented by 1."
22	1479	correction: "...is decremented." -> "...is decremented by 1."
22	1481	correction: "p = 47" -> "p = m x 16 + 15"
22	1483	typo corrections in diagram ("RSCAN0LOCKK register" -> "RSCAN0GLOCKK register")
22	1484	correction in diagram: "r = 0 to 29" -> "r = 0 to 63"
22	1487	ECERVF and EC1EDIC bits added to ECCRCAN0CTL register
22	1492	typo correction (ECCRSCAN0ERDB -> ECCRCAN0ERDB)
22	1492	typo correction (ECCRSCAN0ERDB -> ECCRCAN0ERDB)
22	1496	typo correction ("must not bet allocated" -> "must not be allocated")
23	1497	D1M1(H) devices R7F701424 to R7F701427 removed from RSCAN Unit Configurations and Channels table
23	1497	D1M1(H) devices R7F701424 to R7F701427 removed from RSCAN units table
23	1497	D1M1A added in RS-CANFD units table
23	1498	typo correction ("... by the letter “y” (y = 0, 1)" -> "... by the index “y” (y = 0, 1)")
23	1498	r index number range corrected
23	1515	rephrasing: "forcibly returns the state" -> "forcibly changes the state"
23	1520	BLF bit description corrected: "bus" -> "bus lock"
23	1520	correction: "If any of these flags is set to 0 at the same time..." -> "If any of these flags is set to 1 at the same time..."
23	1520	rephrase: "the flag bits are only set by the error event" -> "the flag bits are only set to 1 by the error event"
23	1520	typo correction ("were all 0 at the when time the error occurred." -> "at the time when the error occurred."

Revision 2.01 History, Date Aug 26, 2016 (9/18)

Section	Page	Summary
23	1522	correction: "(REC[7:0] or TEC[7:0] value > 127)" -> "((128 <= TEC[7:0] <= 255) or (128 <= REC[7:0]))"
23	1522	modifications in "BOEF Flag" description: "bus off state is reached" -> "bus off state is entered"
23	1525	repharsing: "the lowest transmit buffer number" -> "the transmit buffer with the smallest number"
23	1537	typo correction ("as the entire unit" -> "for the entire unit")
23	1537	typo correction ("in the entire unit" -> "for the entire unit")
23	1546	valid number range of NRXMB[7:0] corrected (47 -> 48)
23	1554	"This flag is 00H in global reset mode." added to RFMC[7:0] bits description
23	1556	typo correction ("is decremented" -> " is decremented by 1")
23	1561	CFITR bit description corrected (pclk -> pclk/2)
23	1563	rephrasing: "on the identical channel" -> "on the same channel"
23	1567	correction: "The CFMLT flag..." -> "The CFFLL flag..."
23	1568	typo correction ("is decremented" -> " is decremented by 1")
23	1569	typo correction ("is incremented" -> " is incremented by 1")
23	1584	table layout corrected
23	1586	rephrasing: "arbitration loss" -> "arbitration-lost"
23	1605	correction: "(g + 1)-buffer transmit queue" -> "(g + 1) transmit queue"
23	1621	correction: (r = 0 to 29) -> (r = 0 to 63)
23	1628	"NBRP[9:0] Bits" description modified
23	1633	rephrasing: "forcibly returns the state" -> "forcibly changes the state"
23	1638	BLF bit description corrected: "bus" -> "bus lock"
23	1638	correction: "If any of these flags is set to 0 at the same time..." -> "If any of these flags is set to 1 at the same time..."
23	1638	rephrase: "the flag bits are only set by the error event" -> "the flag bits are only set to 1 by the error event"
23	1638	typo correction ("were all 0 at the when time the error occurred." -> "at the time when the error occurred."
23	1638	typo correction (channe -> channel)
23	1640	correction: "(REC[7:0] or TEC[7:0] value > 127)" -> "((128 <= TEC[7:0] <= 255) or (128 <= REC[7:0]))"
23	1640	modifications in "BOEF Flag" description: "bus off state is reached" -> "bus off state is entered"
23	1642	"DBRP[7:0] Bits" description corrected
23	1645	"These bits are based on CAN clock frequency (fCAN)." added
23	1645	correction: "rounded to..." -> "rounded down to..."
23	1645	rephrasing: "...in the channel by setting of..." -> "...from the channel according to the setting of..."
23	1645	correction: "These bits are set to the SSP offset value." -> "These bits set the SSP offset value."
23	1646	typo correction ("CFESI bit in..." -> "(CFESI bit in...")
23	1649	"transmitter delay compensation" removed from "TDCR[7:0] Flag" description
23	1651	"These bits are cleared to 0 in channel reset mode." added
23	1652	TSBTCS[2:0] settings for channels 3 to 5 removed
23	1655	repharsing: "the lowest transmit buffer number" -> "the transmit buffer with the smallest number"
23	1666	access type of reserved RSCFDnCFDGFDCFG register bits corrected (R/W -> R)

Revision 2.01 History, Date Aug 26, 2016 (10/18)

Section	Page	Summary
23	1666	RSCFDnCFDGFDCFGH register removed from access options
23	1667	access type of reserved RSCFDnCFDGCRCCFG register bits corrected (R/W -> R)
23	1669	typo correction ("as the entire unit" -> "for the entire unit")
23	1669	typo correction ("in the entire unit" -> "for the entire unit")
23	1678	valid number range of NRXMB[7:0] corrected (47 -> 48)
23	1690	typo correction ("is decremented" -> " is decremented by 1")
23	1697	CFITR bit description corrected: pclk -> pclk/2
23	1699	rephrasing: "on the identical channel" -> "on the same channel"
23	1704	correction: "The CFMLT flag..." -> "The CFFLL flag..."
23	1705	typo correction ("is decremented" -> " is decremented by 1")
23	1706	typo correction ("is incremented" -> " is incremented by 1")
23	1708	typo correction ("the the" -> "the")
23	1711	typo correction (CFBRS -> CFESI)
23	1716	table layout corrected
23	1721	table layout corrected
23	1724	rephrasing: "arbitration loss" -> "arbitration-lost"
23	1730	correction: "When the MFDF bit is 1 (CAN FD frame)," -> "When the TMFDF bit is 1 (CAN FD frame),"
23	1734	table cross-reference corrected
23	1736	table cross-reference corrected
23	1738	table cross-reference corrected
23	1740	table cross-reference corrected
23	1744	correction: "(g + 1)-buffer transmit queue" -> "(g + 1) transmit queue"
23	1753	correction: "...register is decremented." -> "...register is decremented by 1."
23	1756	correction: "In CAN FD mode, do not access more than 96 bytes..." -> "In CAN FD mode, do not access more than 160 bytes..."
23	1759	correction: "(<RSCFDn_base> + 0000H to <RSCFDn_base> + 0FFFH)" -> "(<RSCFDn_base> + 0000H to <RSCFDn_base> + 05FFH)"
23	1760	correction: (r = 0 to 41) -> (r = 0 to 63)
23	1763	several register names corrected in diagram
23	1768	correction: "CAN registers can be read," -> "Channel-related registers can be read,"
23	1768	corrections in "Channel Stop Mode" section: "GSLPR -> CSLPR
23	1776	obsolete sentence "The priority is determined by using one of the following methods." removed
23	1780	correction: "...timer is decremented by the next..." -> "...timer is decremented by 1 upon the next..."
23	1781	correction in table: "Buffer No. and "Buffer type" swapped
23	1785	correction: "In CAN FD mode, do not access more than 96 bytes..." -> "In CAN FD mode, do not access more than 160 bytes..."
23	1785	RAM test index r corrected
23	1788	register name correction in diagram: RSCFDnCFDCnCFG ->RSCFDnCFDCmFDCFG
23	1789	description concerning NBRP[9:0] and DBRP[7:0] modified
23	1796	interrupt enabling more detailed in diagram
23	1797	"If the next message is received before reading the message out of buffer, the message will be overwritten." added
23	1799	correction: "...is incremented." -> "...is incremented by 1."

Revision 2.01 History, Date Aug 26, 2016 (11/18)

Section	Page	Summary
23	1801	typo correction in diagram ("RFMC[7:0], CFEMP, RFIF: Flags in the RSCAN0RFSTSx register" -> "RFMC[7:0], CFEMP, RFIF: Flags in the RSCAN0RFSTSx register")
23	1808	correction: "...is decremented." -> "...is decremented by 1."
23	1809	correction: "...is decremented." -> "...is decremented by 1."
23	1815	RAM test index r number range corrected
23	1818	ECERVF and EC1EDIC bits added to ECCRCFDnCTL register
23	1823	typo correction (ECCRSCFD0ERDB -> ECCRCFD0ERDB)
23	1828	typo correction ("must not bet allocated" -> "must not be allocated")
24	1830	D1M1A added in ETNB units table
24	1841	correction: "...to load the values from the count descriptor address register..." -> "...to load the values from the current descriptor address register..."
24	1848	typo correction (RAVBES1 -> ETNB)
24	1848	typo correction (RCR -> ESR)
24	1849	typo correction (RAVBES1 -> ETNB)
24	1849	typo correction (RAVBES1 -> ETNB)
24	1849	typo correction (RAVBES1 -> ETNB)
24	1921	initial value of ECSR register corrected
24	1923	initial value of PIR register corrected
24	1923	typo corrections in PIR register bits descriptions (ET_ -> AVB_)
24	1934	typo correction ("ET_RX-ER pin" -> "AVB_RX-ER pin")
24	1978	typo corrections in diagram (RAVBES1 -> ETNB)
24	2007	correction in diagram: txd[7:0] -> txd[3:0]
24	2028	typo corrections in next 3 diagrams (AVB_RX-ER -> AVB_RX_ER)
24	2029	typo corrections in following 6 diagrams (AVB_RXD7 to 0 -> (AVB_RXD3 to 0, AVB_RX-ER ->AVB_RX_ER, AVB_RX-CSR -> AVB_RX_CSR)
24	2032	correction in diagram: "ET_MDC = 0" -> "PIR_MDC = 0"
24	2032	typo corrections in next 4 diagrams (ET_ -> PIR_)
24	2035	typo correction (RAVBES1 -> ETNB)
25	2037	D1M1A added in MLBB units table
26	2045	"Conflict factors" table corrected
26	2056	"Start mode selection" table corrected
26	2059	"When writing the processing to clear WDTA successively, ..." paragraph added
27	2082	"OSTMnTINT period" description modified
29	2108	D1M1A added to note 4
29	2126	typo correction ("TAUBnCSRm Register Contents" -> "TAUBnCSCm Register Contents")
29	2134	typo corrections in TAUBnTOCm bit description (TAUBnTTOUTm -> TAUBTTOUTm)
29	2141	cross reference corrected ("Section 29.12" -> "Section 29.13")
29	2152	typo correction (TAUBnTTOUTm -> TAUBTTOUTm)
29	2191	correction: "subsequently, in a specific interval." -> "continues to generate interrupts at the specified interval."
29	2214	correction: "(100H + a+1)" -> "(10000H + a+1)"
29	2215	typo correction (TAUBnTTINm -> TAUBTTINm)
29	2224	typo correction (TAUBnTTINm -> TAUBTTINm)
29	2229	correction: "TAUBnCNTm ≤ TAUBnCDRm" -> "TAUBnCNTm > TAUBnCDRm"
29	2243	correction: "TAUBnCNTm reloads the TAUBnCDRm value" -> "TAUBnCNTm loads the TAUBnCDRm value"

Revision 2.01 History, Date Aug 26, 2016 (12/18)

Section	Page	Summary
29	2245	correction: "TAUBnCDRm_master + 1" -> "value of TAUBnCDRm of master channel subject to simultaneous rewrite + 1"
29	2257	correction: "Control Bit Settings for Independent Channel Output Mode 1" -> "Control Bit Settings for Synchronous Channel Output Mode 1"
29	2262	correction: " is issued when the count operation is started after restart." -> "is issued when the count operation is started."
29	2262	rephrasing: "Stopping and restarting the operation" -> "Operation stop and restart"
29	2270	correction: "TAUBnCNTm (slave) reloads the TAUBnCDRm ..." -> "TAUBnCNTm (slave) loads the TAUBnCDRm ..."
29	2270	typo correction (TAUBnTTINm -> TAUBTTINm)
29	2275	correction: "... concurrently sets TAUBnTE.TAUBnTEm to 0," -> "... clears TAUBnTE.TAUBnTEm to 0,"
29	2277	modification: "This in turn sets TAUBnTE.TAUBnTEm, ..." -> "This in turn sets TAUBnTE.TAUBnTEm to 1, ..."
29	2287	correction: "Control Bit Settings for Synchronous Channel Output Mode 2" -> "Control Bit Settings for Independent Channel Output Mode 2"
29	2289	correction: "TAUBnCNTm (master) reloads the TAUBnCDRm ..." -> "TAUBnCNTm (master) loads the TAUBnCDRm ..."
29	2290	TAUBnCNTm counter level corrected in diagram (0000H -> 0001H)
29	2292	typo corrections in diagram (TAUBnTTINm -> TAUBTTINm)
29	2293	typo correction in diagram (TAUBnTTOUTm -> TAUBTTOUTm)
29	2296	typo corrections in diagram (TAUBnTTINm -> TAUBTTINm)
29	2302	correcion: "TAUBnCNTm (slave) reloads the TAUBnCDRm ..." -> "TAUBnCNTm (slave) loads the TAUBnCDRm ..."
29	2302	correction: "TAUBnCNTm (master) reloads the TAUBnCDRm ..." -> "TAUBnCNTm (master) loads the TAUBnCDRm ..."
29	2305	correction: "... must be set to synchronous channel output mode" -> "... must be set to synchronous channel output mode 2"
29	2306	modification: "... starts to count down from this value" -> "... starts to count down from this TAUBnCDRm value."
29	2308	typo corrections in diagram (TAUBnTTINm -> TAUBTTINm)
29	2326	correction in diagram: "Eet forced ditomg TAUBnCDRm = 0000H" -> "Set is forced during TAUBnCDRm = 0000H"
29	2327	"AD Conversion Trigger Output Function Type 2" Summary modified
29	2327	typo corrections in diagram (TAUBnTTINm -> TAUBTTINm)
30	2347	"It is always read as 00H." added to TAUJnCSCm register access
30	2348	"It is always read as 00H." added to TAUJnTS register access
30	2349	"It is always read as 00H." added to TAUJnTT register access
30	2351	"It is always read as 00H." added to TAUJnRDT register access
30	2376	typo correction (AUJnCDRm -> TAUJnCDRm)
30	2382	"11: Setting prohibited" added to TAUJnTIS[1:0] bit description
30	2382	typo correction (TAUJnTTINm -> TAUJTTINm)
30	2388	"11: Setting prohibited" added to TAUJnTIS[1:0] bit description
30	2404	"11: Setting prohibited" added to TAUJnTIS[1:0] bit description
30	2410	typo correction (TAUJnTTINm -> TAUJTTINm)
30	2425	correction: "Control Bit Settings for Independent Channel Output Mode 1" -> "Control Bit Settings for Synchronous Channel Output Mode 1"
31	2430	typo correction ("base addresses" -> "base address")

Revision 2.01 History, Date Aug 26, 2016 (13/18)

Section	Page	Summary
31	2430	typo correction ("RTCAn base addresses are listed in the following table." -> "RTCAn base address is listed in the following table.")
31	2437	typo corrections in RTCAnRSUB register description (RTCA0SRBU -> RTCAnSRBU)
31	2464	typo correction ("0000 0001" -> "0000 0001B")
31	2464	typo correction ("0000 0100" -> "0000 0100B")
31	2464	typo correction ("0000 1010" -> "0000 1010B")
31	2476	correction: "... until the completion of the count buffer register reading." -> "... until the counter resuming."
31	2483	typo correction (RTCA0SRBU -> RTCAnSRBU)
32	2484	D1M1A added in PWM-Diag "Number of Units and Channels" section
32	2490	module name "SELB" added
32	2491	typo correction ("<PWBAn_base> + > + 0004H × m" -> <PWBAn_base> + 0004H × m")
32	2505	typo correction (PWSA0QUE0 -> PWSAnQUE0)
32	2507	typo correction in diagram (#PWGAnRSF -> PWGAnRSF)
36	2617	D1M1A added in LCBI units table
37	2667	D1M1A added
37	2667	D1M1A added
37	2667	D1M1A graphics subsystem features added
37	2668	D1M1A added
37	2668	RLE/Sprite Units configuration modified in diagram
37	2669	D1M1A added
37	2669	note 2 added
37	2669	RLE/Sprite Units configuration modified in diagram
37	2670	D1M1A added
37	2670	RLE/Sprite Units configuration modified in diagram
37	2671	D1M1A added
37	2671	RLE/Sprite Units configuration modified in diagram
37	2672	D1M1A added
37	2672	D1M1A added
37	2672	RLE/Sprite Units configuration modified in diagram
37	2673	D1M1A added
37	2673	RLE/Sprite Units configuration modified in diagram
37	2676	D1M1A added
37	2676	D1M1A added
37	2676	D1M1A added
37	2677	D1M1A added
37	2680	section "D1M1A video channels clock generator" added
37	2681	"D1M1A video channels clock generator" diagram added
37	2682	addition: "Clock selections for different layer configurations" -> "D1M2(H) Clock selections for different layer configurations"
37	2684	device list added to CKSC_IPLL0PIXS_CTL register description
37	2685	D1M1A added
37	2694	D1M1A added
37	2695	D1M1A added
37	2696	DOTCLK0DCSID[7:0] = 1 described for D1M1A

Revision 2.01 History, Date Aug 26, 2016 (14/18)

Section	Page	Summary
37	2698	D1M1A added
37	2698	DOTCLK1DCSID[7:0] = 1 described for D1M1A
37	2699	D1M1A added
37	2700	added "DOTCLK0 /7 (D1M1A only)" in VDCE0VOSCSID[2:0] bit description
37	2700	caution modified for D1M1A
37	2700	D1M1A added
37	2701	D1M1A added
37	2702	added "DOTCLK0 /7 (D1M1A only)" in VDCE1VOSCSID[2:0] bit description
37	2702	caution modified for D1M1A
37	2702	D1M1A added
37	2702	addition: "C_ISO_V1PIXCLK" -> "C_ISO_V1PIXCLK (D1M2(H) only)"
37	2703	D1M1A added
37	2704	CKSC_IVOEXS_CTL.VOEXSCSID0 bit description modified for D1M1A
37	2704	D1M1A added
37	2705	D1M1A added
37	2708	D1M1A added
37	2708	D1M1A added
37	2708	correction: "10B: C_ISO_MIPIPIXCLK (D1M2(H))" -> 10B: C_ISO_MIPIPIXCLK (D1M2H)"
37	2709	correction: "D1M1(H) and D1M2(H) only" -> "D1M1(H), D1M2H and D1M1A only"
37	2714	correction: CLMA6 for D1M2(H) only
37	2716	D1M1A added
37	2721	D1M1A added to MIPI units table
37	2724	"Reserved data type error check register" removed from table
37	2726	RSTN_CLK_ON bit description correction: RSTN_CLK_ON = 0 is prohibited
37	2726	DT_EN bit removed from MIPInMODE register
37	2727	RSTN_DATA_ON bit description correction: RSTN_DATA_ON = 0 is prohibited
37	2729	condtions concerning MIPI I/F reset corrected
37	2729	typo correction ("MIP interface" -> "MIPI interface")
37	2731	correction: 1/fMIPIADCLK -> 1/fMIPI_VI_CLK
37	2731	typo correction (MIPISDCLK -> MIPIADCLK)
37	2733	note added in RX_STOP_STATE_C bit description
37	2733	PHY STOP state indication corrected: "RX_STOP_STATE_C = 0" -> "RX_STOP_STATE_C = 1"
37	2739	"MIPInRDT_ERR_ON – Reserved data type error check register" section removed
37	2740	DT_ERR bit removed from MIPInINTSTATUS register
37	2741	"Packet Header word count error (bit[17])" description corrected
37	2743	DT_ERR_EN bit removed from MIPInINTENSET register
37	2745	DT_ERR_MASK bit removed from MIPInINTENCLR register
37	2747	DT_ERR_CLR bit removed from MIPInINTFFCLR register
37	2751	MIPInRDT_ERR_ON removed from "Start Sequence"
37	2752	"MIPInRDT_ERR_ON = 0000 0000H" removed from "Example start Sequence"
37	2752	MIPInRDT_ERR_ON removed from caution
37	2765	"D1M1A Video output selection" section added
37	2765	section "D1M1A video output mode configuration" added

Revision 2.01 History, Date Aug 26, 2016 (15/18)

Section	Page	Summary
37	2772	addition: "VDCE0" -> "VDCE0 (D1M2(H) only)"
37	2772	D1M1A added
37	2772	D1M1A added
37	2772	correction: VDCECTL.V1CTL for D1M2(H) and D1M1 only
37	2773	section name change: "RSDSCFG — RSDS configuration register" -> "RSDSCFG — RSDS and other video output control register"
37	2773	SRGB1_OEN, OLDI_OEN and VODDR_OEN bits added to RSDSCFG register
37	2773	(D1M2(H) only) added
37	2773	(D1M2(H) only) added
37	2774	(D1M2(H) only) added
37	2775	D1M1A added
37	2779	section "Open LDI Interface" added
38	2806	D1M1A added to VDCE units table
38	2813	Serial RGB progressive output added
38	2813	note 3 added in table
38	2815	note 1 added in diagram
38	2816	text addition concerning clocks in serial RGB mode
38	2820	note added concerning input controller and D1L2(H) devices
38	2847	D1M1A added
38	2866	note added concering scaling functions in D1L2(H) devices
38	2915	note added in SC0_RES_DS_H_ON bit description
38	2915	note added in SC0_RES_DS_V_ON bit description
38	2922	note added in SC0_RES_US_H_ON bit description
38	2922	note added in SC0_RES_US_V_ON bit description
38	2961	note added in SC1_RES_US_H_ON bit description
38	2961	note added in SC1_RES_US_V_ON bit description
38	3082	note added concerning Output Image Generator for D1L2(H) devices
38	3084	table note 2 added concerning D1L2(H) devices
38	3123	LVDS I/F remove from video output block diagram
38	3123	gamma correction excluded for D1L2(H) devices
38	3123	note added in diagram concerning gamma correction for D1L2(H) devices
38	3134	"Bit Allocation of LCD Serial for Serial RGB Output" section added
38	3135	"Parallel to Serial Conversion" section added
38	3137	note added concerning settings for Serial RGB output
38	3157	note added in GAM_ON bit description
38	3205	OUT_FORMAT[1:0] = 3 for serial RGB added
38	3205	OUT_FRQ_SEL[1:0] description expanded
38	3205	OUT_PHASE[1:0] bits added to OUT_SET register
38	3215	SYSCNT_INT6.INT_OUT16_ON and SYSCNT_INT6.INT_OUT17_ON removed from table
39	3227	D1M1A added to VOWE units table
40	3229	D1M1A added to video channels index table
40	3229	D1M1A added to VOCA units table
40	3231	D1M1A added
40	3232	D1M1A added

Revision 2.01 History, Date Aug 26, 2016 (16/18)

Section	Page	Summary
40	3253	D1M1A added
40	3253	VOCAnMAXj[11:0] = 4095 setting removed
40	3254	D1M1A added
40	3255	D1M1A added
41	3267	D1M1A added to DISCOM units table
42	3287	D1M1A added in GPU2D clock supply table
42	3287	D1M1A added to GPU2D units table
42	3301	"(VRAM or SDRAM, SFMA)" removed
43	3302	D1M1A added to Sprite Engine units table
43	3302	"Sprite and RLE Units indices" redefined
43	3304	"Bus master IDs" section expanded
43	3305	"Functional Overview" expanded for D1M1A
43	3305	"Indices" list removed here
43	3306	block names changed according to new indices definitions: "RLE Unit 0/1" -> "RLE Engine 0/1"; "RLE0/1 engine" -> "RLE0/1 decocder"
43	3309	addition: "not supported" -> "not supported by RLE units"
43	3314	D1M1A RLE definition registers update procedure added
43	3314	typo correction ("must be change,..." -> "must be changed,...")
43	3316	"Sprite X/Y position" for 16 bpp color formats added
43	3317	"Image Synthesizer line offset" for 16 bpp color formats added
43	3317	"Sprite width and height" for 16 bpp color formats added
43	3320	"RLE/Sprite Units (D1M1A only)" section added
43	3323	D1M1A RLE/Sprite Units registers overview table added
43	3323	typo correction (SPEAnRjRjRBYP -> SPEAnRjRBYP)
43	3324	D1M1A registers added in "RLE Units registers" section
43	3330	"SPEAnRjRBYP - RLE unit bypass mode register" added
43	3332	D1M1A registers added in "Sprite Units registers" section
43	3335	"SPEAnSkBYP - Sprite unit bypass mode register" added
44	3340	D1M1A added to JCUA units table
44	3345	module name for JCSWRST register corrected
44	3385	typo correction (syncm -> SYNCM)
44	3395	typo correction (syncm -> SYNCM)
45	3396	D1M1A added in A/D Converter channels table
45	3396	correction of virtual channels index: (j = 0 to 19) -> (j = 0 to 20)
45	3400	correction: "ADCE0 has 20 virtual channels. " -> "ADCE0 has 21 virtual channels."
45	3459	typo corrections in "Overwrite Error Detecting Function" section (DR -> ADCEnDRj, DIR -> ADCEnDIRj)
45	3471	correction: "(CDGn = 0, CDGn-15 = 1)" -> (CDGn-1 = 0, CDGn to CDG15 = 1)
45	3471	typo correction ("When ad over-voltage ..." -> "When an over-voltage ...")
45	3472	first "Usage Notes for Analog Input Pins" corrected
45	3472	A/D conversion result value formula corrected
46	3473	D1M1A added in Motor control channels table
46	3475	ports for Motor Drivers 2 to 5 corrected
46	3476	note corrected concerning module names in header files
46	3489	typo correction (SMVDD -> SMVCC)

Revision 2.01 History, Date Aug 26, 2016 (17/18)

Section	Page	Summary
46	3513	cross-reference improved
46	3515	cross-reference improved
46	3543	typo correction ("Redirection to the power supply SMVDD" -> "Redirection to the power supply SMVCC")
46	3548	ISMnVAR6CFGm register bits description corrected
46	3563	ECERVF and EC1EDIC bits added to ECCISMnCTL register
48	3574	"ECM Master and Checkers" section added
48	3579	typo correction (ERROUT -> ERROROUT)
48	3579	typo correction in diagram (ERROUT -> ERROROUT)
48	3582	typo corrections in diagram (ERROUT -> ERROROUT)
51	3625	note 2 corrected concerning module names in header files
51	3626	note concerning header file names modified
52	3628	flash memory overview table updated
52	3642	note corrected concerning module names in header files
52	3643	"Registers Related to Product Information" section updated
52	3645	note corrected concerning module names in header files
52	3646	sentence "Changes in settings such as initial setting ..." removed
52	3647	note corrected concerning module names in header files
52	3648	correction: OPBT9 can not be changed by the user
52	3648	note corrected concerning module names in header files
52	3649	correction: OPBT10 can not be changed by the user
52	3649	table note "Temperature sensor reference values are evaluated and written during device production test." removed
52	3649	note corrected concerning module names in header files
52	3650	correction: "...and error notification is enabled upon detection of an ECC 1-bit error." -> "...and error notification is disabled upon detection of an ECC 1-bit error."
52	3660	description of "Reserved" bits in DFECCCTL register description modified
52	3661	description of "Reserved" bits in DFERSTR register description modified
52	3662	description of "Reserved" bits in DFERSTC register description modified
52	3663	description of "Reserved" bits in DFERRINT register description modified
52	3663	typo correction ("32 to 2" -> "31 to 2")
52	3664	description of "Reserved" bits in DFTSTCTL register description modified
52	3664	typo correction ("31 to 21" -> "31 to 16")
53	3666	RAM overview table updated
53	3676	address parity check removed from "LRTSTCTL — Local RAM Test Control Register" description
53	3678	D1M1A added to RRAMECC units table
54	3695	D1M1A added to Video RAM and Wrapper units table
54	3700	typo correction ("..., either of the following events occurs:" -> "..., until either of the following events occurs:")
54	3700	typo correction ("RTOVFL[6:0], x 2" -> "RTOVFL[6:0] x 2")
54	3704	"2D Graphics Processing Unit (GPU2D) write transaction restrictor (D1M1A only)" section added
54	3704	note modified
54	3704	additional description to GPU2D read transactions restrictor
54	3704	additional description to GPU2D write transactions restrictor

Revision 2.01 History, Date Aug 26, 2016 (18/18)

Section	Page	Summary
54	3704	section "2D Graphics Processing Unit (GPU2D) read transaction restrictor" relocated
54	3704	typo correction ("... restricts read transaction" -> ""... restricts read transactions")
54	3704	typo correction ("... restricts write transaction" -> ""... restricts write transactions")
54	3705	correction: "...while a VRAM access is ongoing." -> "...while a GPU2D access is ongoing."
54	3705	typo correction ("FFCO 605CH" -> "FFC0 605CH")
54	3705	typo correction ("These registers..." "This register...")
54	3711	"D1M1A Mirror/VRAM Wrapper window memory map" section added
55	3730	D1M1A added in Boundary scan ID register description
55	3730	correction: "MID[10:0] + 1" -> "{MID[10:0], 1}"
56	3731	section "HyperBus Controller (HYPB)" added
57	3769	section "OctaBus Controller (OCTA)" added
58	3771	section "NAND Flash Memory Interface A (NFMA)" added

Revision 2.10 History

This revision history list shows all modifications of

Rev. 2.10 compared to Rev. 2.01.

NOTES

1. The following revision history table does not mention corrections of very simple and obvious spelling and formatting errors.
2. Throughout this document the features and properties of the new D1M1-V2 device are added. These additions are not explicitly mentioned in the table below.
3. The chapter and page numbers in the table below refer to the older Rev. 2.10 and thus may not be valid for the current revision.

Revision 2.10 History, Date Mar 31, 2017 (1/12)

Section	Page	Summary
1	98	D1M1-V2 device added in caution
1	98	D1M1_3.75M and D1M1_5M added to former products list
1	103	section "RH850/D1M1 products overview" split into "former products" and "development product" overviews
1	107	Serial Flash Memory I/F 2 added for D1M1A
1	107	"Max. clock" specification removed from NAND Flash I/F (NFMA)
1	107	D1M1_3.75M and D1M1_5M devices moved to former products
1	109	correction for D1M1A VO1: "30 MHz VODDR" -> "10 MHz VODDR"
2	153	corrections in "Step 1" to "Step 4": "in 32-bit units" -> "in accordance with each register's size"
2	154	cautions added concerning port control registers (PCRn_m) setting
2	179	DSCTRL register address corrected
2	179	drive strength control Register (DSCTRL) description corrected
2	187	correction: PIPC17 bits 4 to 11 are available
2	234	PFCE21 register availability for D1M1A devices corrected
2	235	PFCE22.bit10 availability for D1M1A devices corrected
2	237	PFC43.bit1 availability for D1M1A devices corrected
2	237	PFCE43.bit[1:0] availability for D1M1A devices corrected
2	239	PFCAE45 register for D1M1A devices added
2	276	note 3 added concerning JP0_4 (DCUTRST) during RESET = L in normal operation mode
2	276	P10, P11, ADCE0Im normal operation pin states during DEEPSTOP corrected
2	277	P10, P11, ADCE0Im debug pin states during DEEPSTOP corrected
2	277	addition to note 3 concerning SYSRES, external RESET and JP0_4
2	279	note 2 added concerning JP0_4 (DCUTRST) during RESET = L in serial programming mode
2	279	P10, P11, ADCE0Im serial programming pin states during DEEPSTOP corrected
2	280	addition to note 2 concerning SYSRES, external RESET and JP0_4
2	281	addition: "Must always be connected" -> "Must always be connected to oscillator"
2	281	recommended connection of unused XT2 corrected
2	281	addition concerning JP0_4 handling if unused
3	284	name of the CPU user's manual corrected

Revision 2.10 History, Date Mar 31, 2017 (2/12)

Section	Page	Summary
3	289	correction: "HV privilege" -> "SV privilege (PSW.UM = 0)"
3	290	correction: "(VM = 0)" => "(UM = 0)"
3	297	additions: "SV" -> "SV (PSW.UM = 0)"
3	305	"If more than one error..." removed from ID1STERSTR_PE1_OS register description
3	311	"If more than one error..." removed from IT1STERSTR_PE1_OS register description
3	324	correction: TAG[19:0] -> TAG[18:0]
3	340	"If more than one error..." removed from AXCD1STERSTRn_PE1_OS register description removed
3	347	"If more than one error..." removed from AXCT1STERSTRn_PE1_OS register description removed
7	386	correction: "PMF bit in ICSR" -> "PMFI bit in ICSR"
7	390	CPU Subsystem interrupts removed from all "Interrupt Exception Handler and Priority" tables
7	390	reserved level interrupts indicated in all "Interrupt Exception Handler and Priority" tables
7	395	Flash access error (INTFLERR) added
7	402	Flash access error (INTFLERR) added
7	409	Flash access error (INTFLERR) added
7	410	correction: D1M1 interrupt channel 205 is reserved
7	416	Flash access error (INTFLERR) added
7	417	correction: D1M1H interrupt channel 205 is reserved
7	423	Flash access error (INTFLERR) added
7	424	correction: D1M2 interrupt channel 205 is reserved
7	430	Flash access error (INTFLERR) added
7	431	correction: D1M2H interrupt channel 205 is reserved
7	437	Flash access error (INTFLERR) added
7	438	correction: D1M1A interrupt channel 205 is reserved
7	447	typo correction (EEINT ->EIINT)
7	448	typo correction (EEINT ->EIINT)
7	455	typo correction (EEINT ->EIINT)
8	475	typo correction (DMACCTL.DMASPD -> DMACTL.DMASPD)
8	478	correction: "the master information of the CPU or PCU assigned to the DMA channel" -> "the master information of the CPU assigned to the DMA channel"
8	509	typo correction (DRQCn.DRQC -> DTFRRQCn.DRQC)
8	509	correction: DRQCn.DRQC -> DTFRRQCn.DRQC
8	510	typo corrections in DRQC bit description (DRQS.DRQ -> DTFRRQn.DRQ)
9	513	reset sources for Real-Time Clock RTCA0 corrected
9	516	"Isolated-Area reset control" seperated for different devices
11	543	typo corrections (TSNSMP[7:0] -> TSNSMPT[7:0])
11	543	initialization step "Temperature measurement start control register" removed
12	565	correction in figure: CLMA0RES is active low level
12	571	typo correction ("Mr is determined by PLLkC.P[2:0]:" -> "Mr is determined by PLLkC.M[1:0]:")
12	629	CKSC_IXCETNBS_CTL register also available for D1M1H
12	630	CKSC_IXCETNBS_ACT register also available for D1M1H
12	659	RSCANDCSID[2:0] description completed
12	684	correction: D1M1H -> D1M1(H)

Revision 2.10 History, Date Mar 31, 2017 (3/12)

Section	Page	Summary
12	685	typo correction: C_ISO_CPU_CLK -> C_ISO_CPUCLK
12	686	SFMA1 and SFMA2 added to C_ISO_SFMA clock domain
12	690	FOUTDIV[9:0] description completed
12	711	"ECM detects false error signals from CLMA at resuming DEEPSTOP" usage note added
13	713	caution 1 corrected
13	715	D1M1A added
14	729	XC0 master ID table completed
14	741	correction in diagram: Video Output Warping Engine has bi-directional data
14	741	correction in diagram: write path of Output Image Generator to XC0 removed
14	747	D1M1A cross-connect XC0 diagram corrected
14	747	SFMA2 added in "D1M1A bus architecture" section (SFMAn, n= 0 to 2)
14	749	correction in diagram: Video Output Warping Engine has bi-directional data
14	760	typo correction ("MRTSC.XC0RES = 1" -> "MRSTC.XC0RES = 1")
14	760	"Bus Switch for HyperBus and OctaBus" section changed to "Bus Switch for external memory interface" and completed description
14	763	PBUS guard 0 channels 2 and 3 added for FCURAM and FACI command-issuing area
14	779	correction in "Right" table column (HV -> SV) and related table note
14	784	corrections in IPGPMTUM4, IPGnPMT4, and IPGnPMTUM4 register description: "PBUS and Data Flash" -> "cross-connect XC1"
14	785	correction: D1L1BL-2M-100/144 -> D1L1
14	787	typo correction (LOCK -> PROTLOCK)
14	788	PROTRDPDEF, PROTWRPDEF, PROTRD. PROTWR bit descriptions modified
14	797	register and bit names corrected in "Guard function matrix" table header row
14	798	typo correction (’Address mask registers" -> "Address valid registers")
14	800	XCGnVLDi register’s reserved bits descriptions added
14	801	XCGnCTL register’s reserved bits descriptions added
14	802	XCGnCTLWi register’s reserved bits descriptions added
14	803	XCGnINTL1 register’s reserved bits descriptions added
14	804	XCGnOVF register’s reserved bits descriptions added
14	807	XCGnERRRTYPE register’s reserved bits descriptions added
14	809	XCGnERRWTYPE register’s reserved bits descriptions added
14	810	section "XC Guard Operation" added
14	811	information about Performance Monitor availability corrected
14	815	PMRIDSET.RIDSET[3:0] bits description corrected
14	816	PMWIDSET.WIDSET[3:0] bits description corrected
14	826	"PRLB units assignments" tables corrected
14	826	note added concerning PRLID settings
15	833	access type of several DBSVCR register bits corrected (R/W -> R)
15	835	access type of several DBKIND register bits corrected (R/W -> R)
15	836	access type of several DBEN register bits corrected (R/W -> R)
15	837	access type of several DBCMDCNT register bits corrected (R/W -> R)
15	838	access type of several DBCONF register bits corrected (R/W -> R)
15	839	access type of several DBDMOV register bits corrected (R/W -> R)
15	840	access type of several DBTR0 register bits corrected (R/W -> R)

Revision 2.10 History, Date Mar 31, 2017 (4/12)

Section	Page	Summary
15	842	access type of several DBTR1 register bits corrected (R/W -> R)
15	843	access type of several DBTR2 register bits corrected (R/W -> R)
15	844	access type of several DBRFPDN0 register bits corrected (R/W -> R)
15	845	access type of several DBRFPDN1 register bits corrected (R/W -> R)
15	846	access type of several DBRFPDN2 register bits corrected (R/W -> R)
15	847	access type of several DBRFSTS register bits corrected (R/W -> R)
15	848	access type of several DBMRCNT register bits corrected (R/W -> R)
15	849	access type of several DBPDCNT0 register bits corrected (R/W -> R)
15	851	note 3 concering setting of DBDCNT0 register removed
15	852	note 2 concering setting of DBDCNT0 register removed
15	855	note 2 concering setting of DBDCNT0 register removed
15	857	reserved bits in SDRATRCTL register description added
15	858	reserved bits in SDRATRINTVL register description added
16	869	access type of several DBSVCR register bits corrected (R/W -> R)
16	871	access type of several DBSTATE0 register bits corrected (R/W -> R)
16	872	access type of several DBACEN register bits corrected (R/W -> R)
16	873	access type of several DBRFEN register bits corrected (R/W -> R)
16	874	access type of several DBCMD register bits corrected (R/W -> R)
16	876	access type of several DBWAIT register bits corrected (R/W -> R)
16	877	access type of several DBKIND register bits corrected (R/W -> R)
16	878	access type of several DBCONF0 register bits corrected (R/W -> R)
16	878	DBCONF0.AWRW0[4:0] bit description completed
16	880	access type of several DBPHYTYPE register bits corrected (R/W -> R)
16	881	access type of several DBTR0 register bits corrected (R/W -> R)
16	882	access type of several DBTR1 register bits corrected (R/W -> R)
16	883	access type of several DBTR2 register bits corrected (R/W -> R)
16	884	access type of several DBTR3 register bits corrected (R/W -> R)
16	885	access type of several DBTR4 register bits corrected (R/W -> R)
16	886	access type of several DBTR5 register bits corrected (R/W -> R)
16	887	access type of several DBTR6 register bits corrected (R/W -> R)
16	888	access type of several DBTR7 register bits corrected (R/W -> R)
16	889	access type of several DBTR8 register bits corrected (R/W -> R)
16	890	access type of several DBTR9 register bits corrected (R/W -> R)
16	891	access type of several DBTR10 register bits corrected (R/W -> R)
16	892	access type of several DBTR11 register bits corrected (R/W -> R)
16	893	access type of several DBTR12 register bits corrected (R/W -> R)
16	894	access type of several DBTR13 register bits corrected (R/W -> R)
16	895	access type of several DBTR14 register bits corrected (R/W -> R)
16	896	access type of several DBTR15 register bits corrected (R/W -> R)
16	897	access type of several DBTR16 register bits corrected (R/W -> R)
16	899	access type of several DBTR17 register bits corrected (R/W -> R)
16	900	access type of several DBTR18 register bits corrected (R/W -> R)
16	902	access type of several DBBL register bits corrected (R/W -> R)
16	902	DBBL.BL[1:0] bit description completed

Revision 2.10 History, Date Mar 31, 2017 (5/12)

Section	Page	Summary
16	903	access type of several DBADJ0 register bits corrected (R/W -> R)
16	904	access type of several DBADJ2 register bits corrected (R/W -> R)
16	904	DBADJ2 register bit descriptions completed
16	905	access type of several DBRFCNF0 register bits corrected (R/W -> R)
16	906	typo correction (03FFH -> 3FFFH)
16	908	access type of several DBRFCNF2 register bits corrected (R/W -> R)
16	909	access type of several DBRNK0 register bits corrected (R/W -> R)
16	910	access type of several DBPDNCNF register bits corrected (R/W -> R)
16	910	DBPDNCNF.PDMODE[1:0] bit description completed
16	910	DBPDNCNF.PDWAIT[7:0] bit description completed
16	911	access type of several DBPDCNT0 register bits corrected (R/W -> R)
16	912	access type of several DBPDCNT1 register bits corrected (R/W -> R)
16	913	access type of several DBPDCNT3 register bits corrected (R/W -> R)
16	914	access type of several DBBS0CNT1 register bits corrected (R/W -> R)
16	915	access type of several DBLGCNTi register bits corrected (R/W -> R)
16	915	DBLGCNTi register’s reserved bits description added
16	915	correction: DBLGSTSn.UDF -> DBLGSTSi.UDF
16	916	access type of several DBTMVAL0i register bits corrected (R/W -> R)
16	916	DBTMVAL0i register’s reserved bits description added
16	916	correction: "2001H to 3FFFFH" -> "2001H to 3FFFH"
16	917	access type of several DBRQCTRi register bits corrected (R/W -> R)
16	917	DBRQCTRi register’s reserved bits description added
16	918	access type of several DBTHRES0i register bits corrected (R/W -> R)
16	918	DBTHRES[2:0]i register’s reserved bits description added
16	919	DBLGSTSi register’s reserved bits description added
16	920	access type of several DBLGQONi register bits corrected (R/W -> R)
16	920	DBLGQONi register’s reserved bits description added
16	923	typo correction ("OPC = PDEA" -> "OPC[5:0] = PDEn")
16	926	threshold register names in the diagram corrected
16	926	correction: DBTHRESi[2:0] -> DBTHRES[2:0]i
16	926	correction: DBTHRESi[2:0] -> DBTHRES[2:0]i
16	927	correction: DBTHRESi[2:0] -> DBTHRES[2:0]i
16	929	QoS counter name corrected in diagram
16	930	QoS counter name corrected in diagram
17	938	3rd Serial Flash Memory Interface A (SFMA2) added for D1M1A devices
17	950	DRCR.BFM{2:0] bit description completed
17	979	correction: "64 bits x RBURST[3:0]" -> "64 bits x (RBURST[3:0] + 1)"
17	980	correction in diagram: "64 x RBURST (read burst length) bits" -> "64 x (RBURST (read burst length) bits + 1)"
17	981	correction in diagram: "64 x RBURST bits" -> "64 x (RBURST (read burst length) bits + 1)"
17	981	correction in diagram: "64 x RBURST bits" -> "64 x (RBURST (read burst length) bits + 1)"
17	983	correction: "64 × RBURST (read burst length)" -> "64 × (RBURST[3:0] (read burst length) + 1)"

Revision 2.10 History, Date Mar 31, 2017 (6/12)

Section	Page	Summary
17	989	correction: "64 x RBURST[3:0] bits" -> "64 x (RBURST[3:0] bits + 1)"
17	997	note 2 concerning SFMA1 clock frequencies removed
18	1030	typo correction (q -> alpha)
19	1135	figure corrected
19	1136	procedure step 6 split into 6 and 7
19	1136	typo corrections in register and bit names: CSSIH0 -> CSIHn
19	1139	figure corrected
19	1139	procedure step 6 split into 6 and 7
21	1262	access type of RIICnMR3.ACKWP bit corrected (R -> R/W)
22	1489	addition concerning channels not included in the test
23	1813	addition concerning channels not included in the test
23	1813	note *1 added concerning RAM test size
24	1858	Ethernet AVB MAC "External clock" defined
24	1860	table note 1 added concerning signals with MII-Lite interface
24	1864	gPTP captured Presentation Time register (GCPT) register removed from table
24	1865	access type of several CCC register bits corrected (R/W -> R)
24	1869	access type of several DLR register bits corrected (R/W -> R)
24	1872	access type of several CSR register bits corrected (R/W -> R)
24	1876	access type of several ESR register bits corrected (R/W -> R)
24	1878	access type of several RCR register bits corrected (R/W -> R)
24	1883	access type of several RPC register bits corrected (R/W -> R)
24	1885	access type of several UFCW register bits corrected (R/W -> R)
24	1886	access type of several UFCS register bits corrected (R/W -> R)
24	1887	access type of several UFCVi register bits corrected (R/W -> R)
24	1889	access type of several UFCDi register bits corrected (R/W -> R)
24	1890	access type of several SFO register bits corrected (R/W -> R)
24	1893	access type of several TGC register bits corrected (R/W -> R)
24	1895	access type of several TCCR register bits corrected (R/W -> R)
24	1897	access type of several TSR register bits corrected (R/W -> R)
24	1901	access type of several TFA2 register bits corrected (R/W -> R)
24	1908	access type of several DIS register bits corrected (R/W -> R)
24	1910	access type of several EIC register bits corrected (R/W -> R)
24	1912	access type of several EIS register bits corrected (R/W -> R)
24	1918	access type of several RIS0 register bits corrected (R/W -> R)
24	1920	access type of several RIC1 register bits corrected (R/W -> R)
24	1921	access type of several RIS1 register bits corrected (R/W -> R)
24	1922	access type of several RIC2 register bits corrected (R/W -> R)
24	1924	access type of several RIS2 register bits corrected (R/W -> R)
24	1928	access type of several TIS register bits corrected (R/W -> R)
24	1940	access type of several GIC register bits corrected (R/W -> R)
24	1941	access type of several GIS register bits corrected (R/W -> R)
24	1941	gPTP captured Presentation Time register (GCPT) register removed
24	2061	typo corrections (E-DMAC -> AVB-DMAC)
28	2121	AWOTnCTL register's reserved bits description added

Revision 2.10 History, Date Mar 31, 2017 (7/12)

Section	Page	Summary
28	2122	AWOTnTOE register's reserved bits description added
28	2123	correction: AWOTnSTC can be accessed in 8-bit units.
28	2123	AWOTnFLG register's reserved bits description added
28	2123	AWOTnSTC register's reserved bits description added
28	2127	AWOTnEMU register's reserved bits description added
30	2373	TAUJnCSRm register value after reset corrected
30	2377	correction: (TAUJnRDFm = 1) -> (TAUJnRDTm = 1)
32	2514	typo corrections in "PWGA" description paragraph: PWGAnCRBR -> PWGAnCRDR
34	2586	correction: "PCMPnSTC.PCMPnOV = 1" -> "PCMPnSTR.PCMPnOV = 1"
34	2586	correction: "PCMPnSTC.PCMPnUR = 1" -> "PCMPnSTR.PCMPnUR = 1"
34	2603	correction in PCMPnOV and PCMPnUR bit descriptions: interrupt -> flag
34	2604	correction in PCMPnOVC and PCMPnURC bit descriptions: interrupt -> flag
34	2604	correction: "This register is used to reset the FIFO buffer interrupt flags." -> "This register is used to reset the FIFO buffer status flags."
35	2609	SSIRxD and SSITxD signals combined to SSIDATA
35	2611	SSIRxD and SSITxD signals combined to SSIDATA in block diagram
35	2614	corrections in SCKP bit description: SCK -> SSISCK
35	2617	corrections in TOIRQ, RUIRQ, ROIRQ and IIRQ bit descriptions: "SSI interrupt" -> "SSIF interrupt"
35	2633	type correction in diagram: "SSI module" -> "SSIF module"
35	2640	typo correction in diagram: "SSI status register bits" -> "SSIF status register bits"
35	2642	typo correction in diagram: "SSI status register bits" -> "SSIF status register bits"
36	2659	LCBInTGTFTSEQ signal removed from diagram
36	2685	clock divisors in LCBInTFTPCPRS[15:0] bit description corrected
37	2698	D1M1H removed from note 1
37	2707	several video input clock settings added
37	2730	correction: CKSC_IDOTCLK0S_CTL -> CKSC_IDOTCLK1S_CTL
37	2734	typo correction (DOTCLK0 -> DOTCLK1)
37	2734	VDCE1VOSCSID[2:0] bit description corrected
37	2736	CKSC_IVOEXS_CTL.VOEXSCSID0 bit description corrected for D1M1A
37	2740	correction: D1M2H -> D1M2(H)
37	2740	correction: CKSC_IVDCE0VIS_CTL register also in D1M2 available
37	2741	correction: CKSC_IVDCE0VIS_ACT register also in D1M2 available
37	2747	typo correction ("VDCEnVENm signals" -> "VDCEnVEm signals")
37	2751	correction: "INP_FORMAT[2:0] = 000B" for RGB888, YCrCb444
37	2756	typo correction (MIPI0VI_VSYNC -> MIPI0_VI_VSYNC)
37	2758	condition for RSTN_CLK_ON = 0 corrected
37	2759	condition for RSTN_DATA_ON = 0 corrected
37	2762	correction: "MIPInBUF_CTL – Buffer control" -> "MIPInBUF_CTL – SubLVDS buffer control"
37	2763	correction: CLK_COUNT -> DATA_COUNT[7:0]
37	2764	correction: CLK_COUNT -> DATA_COUNT[7:0]
37	2764	correction: CLK_COUNT -> DATA_COUNT[7:0]
37	2764	correction: CLK_COUNT -> DATA_COUNT[7:0]
37	2764	correction: CLK_COUNT -> DATA_COUNT[7:0]

Revision 2.10 History, Date Mar 31, 2017 (8/12)

Section	Page	Summary
37	2773	correction: "Escape Mode Entry Command Error (bit[15])" -> "Escape Mode Entry Command Error (bit[15], [11])"
37	2773	correction: "False Control Error (bit[16])" -> "False Control Error (bit[16], [12])"
37	2776	corrections in CTL_ERR_0_EN, ESC_ERR_0_3_EN, SOT_SYNC_ERR_0_EN, SOT_ERR_0_EN bit descriptions: "lane 1" -> "lane 0"
37	2776	corrections in CTL_ERR_1_EN, ESC_ERR_1_3_EN, SOT_SYNC_ERR_1_EN, SOT_ERR_1_EN bit descriptions: "lane 2" -> "lane 1"
37	2777	corrections in CTL_ERR_1_MASK, ESC_ERR_1_3_MASK, SOT_SYNC_ERR_1_MASK, SOT_ERR_1_MASK bit descriptions: "lane 2 enable" -> "lane 1 mask"
37	2778	correction in BUF_OR_ERR_MASK bit description: "BUFFER overrun error enable" -> "BUFFER overrun error mask"
37	2778	corrections in CTL_ERR_0_MASK, ESC_ERR_0_3_MASK, SOT_SYNC_ERR_0_MASK, SOT_ERR_0_MASK bit descriptions: "lane 1 enable" -> "lane 0 mask"
37	2779	corrections in CTL_ERR_0_CLR, ESC_ERR_0_3_CLR, SOT_SYNC_ERR_0_CLR, SOT_ERR_0_CLR bit descriptions: "lane 1 enable" -> "lane 0 enable clear"
37	2779	corrections in CTL_ERR_1_CLR, ESC_ERR_1_3_CLR, SOT_SYNC_ERR_1_CLR, SOT_ERR_1_CLR bit descriptions: "lane 2 enable" -> "lane 1 enable clear"
37	2780	correction in BUF_OR_ERR_CLR bit description: "BUFFER overrun error enable" -> "BUFFER overrun error enable clear"
37	2783	typo correction (MIPIn_RST_CTL -> MIPInRST_CTL)
37	2789	typo correction ("OUT_SET.OUT_FORMAT[2:0] = 000B" -> "OUT_SET.OUT_FORMAT[1:0] = 00B"
37	2791	typo correction ("OUT_SET.OUT_FORMAT[2:0] = 000B" -> "OUT_SET.OUT_FORMAT[1:0] = 00B"
37	2797	note *1 removed from Serial RGB
37	2798	footnote 5 added
37	2798	D1M1A video output selection figure corrected
37	2807	D1M1A removed from C_ISO_VI1PIXCLK selection
37	2811	correction: VOWEMSC.RBMASK[31:0] -> VOWEMMC.RBMASK[31:0]
37	2814	correction: CKSC_OLDI0SYS_CTL -> CKSC_IVOEXS_CTL
37	2818	addition to LVEN and LVRES bit descriptions: bits must not be changed to 0 during Open LDI operation
37	2819	OLDInCR1 description corrected
37	2825	"Sample Configuration flow of OpenLDI" revised
37	2829	"Limitation" section added
38	2847	S0_WLINE interrupts removed from table
38	2861	correction: INP_ES_EDGE -> INP_HS_EDGE
38	2876	correction: SC_SCL0_DS4.SC_RES_DS_H_ON -> SC_SCL0_DS1.SC_RES_DS_H_ON
38	2878	correction: SC_SCL0_DS4.SC_RES_DS_H_ON -> SC_SCL0_DS1.SC_RES_DS_H_ON
38	2886	correction: SCL0_DS3.RES_HW -> SC_SCL0_DS3.SC_RES_HW
38	2887	correction: DE_HS_BYPASS -> DE_HSBYPS_EN
38	3044	correction: "The GR_BASE[31:0] bits should be set in units of 64 bits" -> "The GR_BASE[31:0] bits should be set in units of 128 bytes"
38	3044	"Restriction", "Recommendation" and "Example" added concering frame buffer addressing
38	3049	"Calculation Formula" for GR_BASE[31:7] corrected

Revision 2.10 History, Date Mar 31, 2017 (9/12)

Section	Page	Summary
38	3050	correction in GR_STA_POS[5:0] bit description: "amount of data" -> "amount of pixels"
38	3073	note added concerning frame buffer addressing
38	3078	correction in GR_STA_POS[5:0] bit description: "amount of data" -> "amount of pixels"
38	3090	GR3_UPDATE.GR3_P_VEN bit description corrected
38	3114	correction: "set GR_VIN_HW to 2 and GR_VIN_GRC_HW..." -> "set GR_VIN_GRC_HW[10:0]..."
38	3123	corrections in table: GR_OIR_VEN -> GR_OIR_UPDATE
38	3150	note 2 added concerning frame buffer addressing
38	3153	name of bits 12 to 10 corrected
38	3183	correction: TCON_STVA_HW -> TCON_STVA_VW[10:0]
38	3183	correction: TCON_STVB_HW -> TCON_STVB_VW[10:0]
38	3250	OUT_SET.OUT_PIXEL_INV_ON and OUT_SET.OUT_SUM_MOVE[4:0], OUT_SET.OUT_PHASE[1:0] bit descriptions removed from table
38	3251	note added concerning interrupt settings and clock supply
38	3251	S0_WLINE interrupt removed
38	3252	SYSCNT_INT2.INT_STA9 bit removed
38	3253	SYSCNT_INT5.INT_OUT14_ON bit description removed from table
38	3253	SYSCNT_INT5.INT_OUT9_ON bit removed
38	3259	INT_STA9 bit removed from SYSCNT_INT2 register
38	3263	INT_OUT9_ON bit removed from SYSCNT_INT5 register
40	3275	correction: VOCAnCFG0 -> VOCAnMmCFG0
40	3275	correction: VOCAnCFG1 -> VOCAnMmCFG1
40	3275	correction: VOCAnOFFSj.VOCAnHOFFS/.VOCAnVOFFS -> VOCAnOFFSj.VOCAnHOFFSj/.VOCAnVOFFSj
40	3275	typo correction (VOCAnHOFFS -> VOCAnHOFFSj)
40	3275	typo correction (VOCAnVOFFS -> VOCAnVOFFSj)
40	3282	activity monitor detection time settings corrected
40	3282	Activity Monitor functional description corrected
40	3284	correction: VOCAn.VOCAnE16/E17 -> VOCAnSTR.VOCAnE16/E17
40	3287	VOCAn register names corrected in overview table
40	3293	time values in VOCAnTIMEj register description defined accurately
40	3294	VOCAnOFFSj register's reserved bits description added
40	3295	VOCAnDISPj register's reserved bits description added
40	3296	VOCAnACTj register's reserved bits description added
40	3297	VOCAnDIFF register's reserved bits description added
40	3298	VOCAnMmCFG0 register's reserved bits description added
40	3299	VOCAnMmCFG1 register's reserved bits description added
40	3300	VOCAnMmCFG2 register's reserved bits description added
40	3301	VOCAnMmCFG3 register's reserved bits description added
40	3302	VOCAnMmCFG4 register's reserved bits description added
40	3303	VOCAnMmCFG5 register's reserved bits description added
40	3304	VOCAnMmCFG6 register's reserved bits description added
40	3305	VOCAnMmCFG7 register's reserved bits description added
40	3306	VOCAnEXPDk register's reserved bits description added
40	3307	"Procedure for entering and resuming DEEPSTOP" usage note added

Revision 2.10 History, Date Mar 31, 2017 (10/12)

Section	Page	Summary
43	3366	SPEAnRjRUP, SPEAnRjRBYP and SPEAnRjRCFG register addresses corrected
43	3367	access type of several SPEAnRLSL/SPEAnRjRLSL register bits corrected (R/W -> R)
43	3368	access type of several SPEAnSTAi/SPEAnRjSTAi register bits corrected (R/W -> R)
43	3369	access type of several SPEAnPHAi/SPEAnRjPHAi register bits corrected (R/W -> R)
43	3370	access type of several SPEAnVDCi/SPEAnRjVDCi register bits corrected (R/W -> R)
43	3370	"SPEAnVDCi/SPEAnRjVDCi - RLE Engine i read master ID register" description modified
43	3370	typo corrections in master ID assignment tables (SPEAnVDCi[3:0] -> SPEAnRVDCi[3:0])
43	3372	access type of several SPEAnRCMi/SPEAnRjRCMi register bits corrected (R/W -> R)
43	3373	access type of several SPEAnRUP/SPEAnRjRUP register bits corrected (R/W -> R)
43	3374	access type of several SPEAnRjRBYP register bits corrected (R/W -> R)
43	3375	access type of several SPEAnRCFG/SPEAnRjRCFG register bits corrected (R/W -> R)
43	3376	access type of several SPEAnSkEN register bits corrected (R/W -> R)
43	3377	access type of several SPEAnSkDS register bits corrected (R/W -> R)
43	3378	access type of several SPEAnSkUP register bits corrected (R/W -> R)
43	3379	access type of several SPEAnSkBYP register bits corrected (R/W -> R)
43	3380	access type of several SPEAnSkDAm register bits corrected (R/W -> R)
43	3381	access type of several SPEAnSkVDm register bits corrected (R/W -> R)
43	3382	access type of several SPEAnSkLYm register bits corrected (R/W -> R)
43	3383	access type of several SPEAnSkPSm register bits corrected (R/W -> R)
44	3388	several JCUA register names in overview table corrected
44	3414	correction: JIFECNT -> JIFDCNT
44	3426	typo correction (JIFDSLC -> JIFDSDC)
44	3427	correction in diagram: "JINEN bit in JINTE0" -> "JINEN bit in JINTE1"
44	3428	correction: JINT1 -> JINTS1
44	3428	typo correction in figure (JUIFDDOST -> JIFDDFST)
44	3430	correction in diagram: "Detect INTJCU0EDI" -> "Detect INTJCUA0EDI"
44	3430	note added concerning unknow markers
44	3438	correction in diagram: "Detect INTJCU0DTI interrupt" -> "Detect INTJCUA0DTI interrupt"
45	3448	Pull Down control register 1 (ADCEnPDCTL1) removed
45	3459	correction: ADCEnADCR.TSNSELFDIAG bit is R/W
45	3459	typo correction: ADCELnDRj -> ADCEnDRj
45	3468	correction: ADCEnPWDTSNDR.TSN[15:0] -> ADCEnPWDTSNDR.TSNDR[15:0]
45	3480	correction: "The SHACT and SGACT bits are cleared when HALT is executed." -> "The SGACT bits are cleared when HALT is executed."
45	3481	ADCEnSGSTR.SEF[0] bit description corrected
45	3482	initial value of ADCEnSGTSELx register corrected
45	3486	initial value of ADCEnTSNSGTSEL register corrected
45	3488	output signal names corrected
45	3489	Pull Down control register 1 (ADCEnPDCTL1) description removed
45	3507	sentence "ADPEm is enabled if PEIE of ADCEnSFTCR is set to 1." removed
45	3511	ADCEnPDCTL1.PDNAm bits removed from "Diagnostic of Open Pins" section
45	3511	ADCEnPDCTL1.PDNAm bits removed from figure
45	3512	correction: "...declines to 0 V" -> "...declines to almost 0 V"
46	3574	access type of several ISMnGCTL register bits corrected (R/W -> R)

Revision 2.10 History, Date Mar 31, 2017 (11/12)

Section	Page	Summary
46	3576	access type of several ISMnGSTR register bits corrected (R/W -> R)
46	3577	access type of several ISMnGCFG register bits corrected (R/W -> R)
46	3580	correction: ISMnCTL -> ISMnGCTL
46	3581	access type of several ISMnGSTC register bits corrected (W -> R)
46	3582	access type of several ISMnCCMRm register bits corrected (R/W -> R)
46	3584	access type of several ISMnCCMPm register bits corrected (R/W -> R)
46	3585	access type of several ISMnCIOCm register bits corrected (R/W -> R)
46	3586	access type of several ISMnCOPT register bits corrected (R/W -> R)
46	3587	access type of several ISMnPAR0CFGm register bits corrected (R/W -> R)
46	3588	access type of several ISMnPAR1CFGm register bits corrected (R/W -> R)
46	3589	access type of several ISMnPARiCFGm register bits corrected (R/W -> R)
46	3590	access type of several ISMnVARjCFGm register bits corrected (R/W -> R)
46	3591	access type of several ISMnVAR6CFGm register bits corrected (R/W -> R)
46	3592	access type of several ISMnGZPDCTL register bits corrected (R/W -> R)
46	3598	access type of several ISMnZPDCTL register bits corrected (R/W -> R)
46	3598	signal names corrected
46	3599	access type of several ISMnZPDOPT register bits corrected (R/W -> R)
46	3603	access type of several ISMnZPDCMPm register bits corrected (R/W -> R)
46	3605	access type of several ISMnEMU register bits corrected (R/W -> R)
46	3608	correction in ECCISMnCTL.EC2EDIC bit description: INTECCDCNRAM -> INTECCDPERIRAM
46	3608	correction in ECCISMnCTL.ECER2FC bit description: INTECCDCNRAM -> INTECCDPERIRAM
48	3620	FACI error sources No. 17, 18 added in ECM error inputs list
48	3620	note 3 added to Video Output Monitor error
48	3621	note 4 added to Clock Monitor interrupts CLMAn
48	3623	correction: Error Output Level = L in the last table row
48	3636	added note concerning ECMMICFG1 register write protection
51	3670	access type of several IDMODI register bits corrected (W -> R)
52	3691	"Chip ID Register" added
52	3697	Code Flash sub-test control register (PE1) added in the table
52	3697	Code Flash sub-test control register (VCI) added in the table
52	3702	"If more than one error occurs..." removed from CF1STERSTR_VCI2CFB/PE1_OS register description
52	3704	"CFSTSTCTL_VCI2CFB/PE1 — Code flash sub-test control register" description added
52	3709	typo corrections in ERRCLR bit description (CF1STERSTR -> DFERSTR)
52	3712	correction: items prohibited also during blank check
53	3713	D1M1_5MB and D1M1H_5MB product names corrected in table
53	3721	"If more than one error occurs..." removed from LR1STERSTR register description
53	3725	Local RAM "Usage Notes" added
53	3732	ECDEDFm, ECSEDFm clearing conditions corrected
53	3733	ECOVFF clearing conditions corrected
54	3767	ECDEDFm, ECSEDFm clearing conditions corrected
54	3768	ECOVFF clearing conditions corrected
56	3781	typo correction ("MRTSC.XC0RES = 1" -> "MRSTC.XC0RES = 1")

Revision 2.10 History, Date Mar 31, 2017 (12/12)

Section	Page	Summary
56	3781	OctaBus Controller connected signals corrected
56	3782	note added concerning HyperBus memory area access in Manual Mode
56	3782	maximum flash/RAM size given in "HyperBus Interface" features
56	3795	"Manual Mode Enable Setting Register (SMENR)" section corrected
56	3802	DRDMCR.DMCYC[3:0] bit description completed
56	3803	SMDMCR.DMCYC[4:0] bit description completed
56	3807	correction: "By external address space read mode,..." -> "By external address space read/write mode,..."
56	3808	correction: "SMCMR = 0x0060_0000" -> "SMCMR = 0x0020_0000"
56	3809	correction: "DRCMR = 0x00e0_0000" -> "DRCMR = 0x00a0_0000"
56	3815	"Write Buffer Area" information table corrected
57	3818	typo correction ("MRTSC.XC0RES = 1" -> "MRSTC.XC0RES = 1")
57	3818	OctaBus Controller connected signals corrected
57	3819	OctaBus module description added
58	3898	DMA_CTRL.DMA_BURST[2:0] bit description corrected
58	3954	correction in figure: "write DMA_TRIG_LVL" -> "write DMA_TRIG_TLVL"

Revision 2.20 History

This revision history list shows all modifications of

Rev. 2.20 compared to Rev. 2.10.

NOTE

The following revision history table does not mention corrections of very simple and obvious spelling and formatting errors.

Revision 2.20 History, Date Jan 26, 2018 (1/8)

Section	Page	Summary
1	98	caution concerning D1M1A and D1M1-V2 devices preliminary specification removed
1	103	D1M1H_ CPU frequencies corrected
1	108	PLL0 frequency correction: "max. 480 MHz" -> "max. 960 MHz"
1	109	correction: TCON available in D1M1A devices
1	110	footnote numbers for D1M1 development products "Voltage supply" items corrected
1	110	D1M1-V2 package type corrected
1	115	correction: TCON available in D1M2 devices
1	116	footnote numbers for D1M2 products "Voltage supply" items corrected
2	121	corrections in diagram: A0VREFN -> A0VSS, ZPDVREFVSS -> ZPDVSS
2	123	corrections in diagram: SDRADQM[3:0] -> SDRADM[3:0]
2	129	appropriate "PPRn_m read value" column cells merged
2	179	device dependency of DSCTRL.SDRDSA[1:0] added
2	179	device dependency of DSCTRL.SDRDSD2[1:0] added
2	179	device dependency of DSCTRL.SDRDSD3[1:0] added
2	180	note added concerning header files module name
2	180	device dependency of DSCTRL.SDRDSC[1:0] added
2	180	device dependency of DSCTRL.SDRDSD0[1:0] added
2	180	device dependency of DSCTRL.SDRDSD1[1:0] added
2	180	correction in all port register tables: PPCMDn and PPROTSn access size 8 bit -> 32 bit
2	196	correction: PIPC3.bit 0 not available in D1L2H
2	204	correction: PFCE44.bit5 not available in D1L2H
2	235	correction: PFC22.bit10 not available in D1M1-V2
2	237	correction: PIPC43.bit[1:0] not available in D1M1A
2	237	correction: PFC43.bit1 not available in D1M1A
2	237	correction: PFCE43.bit[1:0] not available in D1M1A
2	238	correction: PIPC44.bit[11:10,8,6,3,1] not available in D1M1A
2	239	correction: PIPC45.bit[11:6,4,2:0] not available in D1M1A
2	239	device dependency of PFCAE45.bit13 added
2	256	note 2 added concerning usage of P45_1
2	257	description of 2nd chip select for Octa memory interface added
2	276	ADCE0Im DEEPSTOP pin state in normal operation mode corrected
2	276	P10, P11 pin state in normal operation mode corrected
2	277	ADCE0Im DEEPSTOP pin state in debug mode corrected
2	277	P10, P11 pin state in debug mode corrected
2	279	ADCE0Im DEEPSTOP pin state in serial programming mode corrected

Revision 2.20 History, Date Jan 26, 2018 (2/8)

Section	Page	Summary
2	279	P10, P11 pin state in serial programming mode corrected
2	281	recommended connection of unused XT2 corrected
3	322	access type of reserved bits corrected (R/W -> R)
3	333	"XC cache tag RAM ECC Control Register" removed
3	333	correction: "XC cache data RAM ECC Control Register" -> "XC cache data/tag RAM ECC Control Register"
3	334	correction: "cache data RAM" -> "cache data RAM and tag RAM"
3	334	correction: "XC cache data RAM data ECC control register" -> "XC cache data/tag RAM ECC control register"
3	342	"AXCTECCCTL_PE1_OS – XC cache tag RAM ECC control register" description removed
4	353	step 3 of the "Register protection unlock sequence" rephrased
5	377	extended area added to address map
5	379	extended area added to address map
7	383	"Index" table added
7	386	correction: "PMFI bit in ICSR" -> "PMEI bit in ICSR"
7	389	index correction throughout this section: INTPn ->INPx
7	452	DMA bus error status registers addresses corrected throughout this section
7	454	initial value of BECMBUSERRMASK register corrected
8	478	"VM" item removed from table
8	491	VM bit removed from DMnnCM registers
9	516	note added
9	516	PWRGD counter frequency corrected: 240 kHz -> 100 kHz
9	517	step (6) description improved
9	517	note added concerning ISOPWRES occurrence
9	517	correction: "0.5 ms, when High Speed IntOsc clock fRH is running" -> "0.8 ms, when High Speed IntOsc clock fRH is running"
9	533	diagram corrected
9	534	diagram corrected
12	548	typo correction ((HS IntOsc) -> (LS IntOsc))
12	551	typo correction (ROSC -> ROSCS)
12	552	typo correction (ROSC -> ROSCS)
12	553	typo correction (ROSC -> ROSCS)
12	554	typo correction (ROSC -> ROSCS)
12	555	typo correction (ROSC -> ROSCS)
12	557	note 2 added related to APC_CLK_RATIO
12	588	note added concerning wake-up by INTPx
12	619	CKSC_IPLLFIXS_ACT register availability corrected
12	638	device dependency of APB_CLK_RATIO register corrected
12	646	correction: "...at a frequency of 4 MHz or higher..." -> "...at a frequency of higher than 4 MHz..."
12	687	maximum C_AWO_RTCA clock frequency corrected: 8 MHz-> 4 MHz
12	689	note 3 rephrased
13	714	cautions 3 and 4 added
13	714	correction: INTPn->INTPx
13	716	corrected description concerning IOHOLD in DEEPSTOP

Revision 2.20 History, Date Jan 26, 2018 (3/8)

Section	Page	Summary
13	727	"Preparation for DEEPSTOP" description corrected
13	730	figure added
13	730	figure corrected
14	764	correction: "SFMA2 SPBSSL" -> "SFMA2SSL"
14	764	XCRAMCFG0.CSSEL2 (bit 21) added
14	767	correction: PBGA10A -> PBG10A
14	774	reference added concerning Flash Memory User's Manual
14	782	"IPG Protection Setting Registers for Illegal Virtual Machine" section removed
14	783	VD bit desription corrected
14	783	"IPGnECR, and IPGnECRUM (n=0)" removed from register description
14	784	"IPGnADR, and IPGnADRUM (n=0)" removed from register description
14	784	"IPGnEN, and IPGnENUM (n=0)" removed from register description
14	785	"IPGnPMT0, and IPGnPMTUM0 (n=0)" removed from register description
14	786	"IPGnPMT2, and IPGnPMTUM2 (n=0)" removed from register description
14	786	"IPGnPMT3, and IPGnPMUTM3 (n=0)" removed from register description
14	787	"IPGnPMT4, and IPGnPMTUM4 (n=0)" removed from register description
14	788	note added
14	789	PBG0A PBUS guard channels corrected
14	789	PBG2B PBUS guard added
14	789	PBG5A PBUS guard added
14	790	PROTLOCK (bit 31) added to FSGD0BPROTn register
14	790	PROTVM (bit 26) removed from FSGDxxPROTn register
14	794	VM (bit 7) removed from ERRSLVxxTYPE registers
14	814	information about Performance Monitor availability corrected
14	818	PMRIDSET.RIDSET[3:0] bits description corrected
14	819	PMWIDSET.WIDSET[3:0] bits description corrected
14	830	section "Video frame buffer and PRLB usage" added
15	862	addition concerning signal representations
17	948	CMNCR.MOIIO1[1:0] bit description modified
17	948	CMNCR.MOIIO2[1:0] bit description modified
17	948	CMNCR.MOIIO3[1:0] bit description modified
17	949	CMNCR.IO0FV[1:0] bit description modified
17	949	CMNCR.IO2FV[1:0] bit description modified
17	949	CMNCR.IO3FV[1:0] bit description modified
17	949	CMNCR.MOIIO0[1:0] bit description modified
17	953	note added to DRCR.BFM[2:0] bit description
17	966	SMRDR0.RDATA0[31:0] bit description modified
17	967	SMRDR1.RDATA1[31:0] bit description modified
17	968	SMWDR0.WDATA0[31:0] bit description modified
17	969	SMWDR1.WDATA1[31:0] bit description modified
17	971	CKDLY_TS[1:0] bits added to CKDLY register of D1M1A and D1M1-V2 devices
17	986	note added concerning SFMA access via XC cache
17	999	correction in "Pin Status (2)" table: MOIIOx -> IOxFV
17	999	correction in "Pin Status (3)" table: MOIIOx -> IOxFV

Revision 2.20 History, Date Jan 26, 2018 (4/8)

Section	Page	Summary
17	1002	description concerning sampling point value modified
17	1004	note concerning data learning pattern corrected
17	1006	"Notes on using the SFMA Cache" added
19	1061	CSIH1BRS[11:0] bits access modes corrected (R -> R/W)
19	1105	correction: ", and thus all CSIHTSCK are stopped" -> ", and thus CSIHTSCK of the corresponding channel is stopped"
19	1107	correction: ", and thus all CSIHTSCK are stopped" -> ", and thus CSIHTSCK of the corresponding channel is stopped"
19	1139	CSIHnMCTL2.CSIHnRRA[6:0] in figure corrected
20	1182	"For response reception:" description corrected
20	1222	typo correction (PRER -> RPER)
20	1222	typo correction (PRER -> RPER)
20	1223	"When the FSM bit in the RLN3nLDFC ..." added
22	1350	correction: "<RSCAN0_base> + <RSCAN0_base> + 104CH" -> "<RSCAN0_base> + 104CH"
22	1411	added "After all other bits in the RSCANnCFCCk register ..." to CFE bit description
22	1472	references to *1 note added in the table
22	1517	note added concerning ECC decoder assignments
22	1519	relation to ECC 1-bit detection added in EC1EDIC register description
22	1519	relation to ECC 1-bit detection added in ECER1F register description
22	1519	relation to ECC 2-bit detection added in EC2EDIC register description
22	1519	relation to ECC 2-bit detection added in ECER2F register description
23	1535	ISO standard compatibilities in "CAN FD protocol switchover" corrected
23	1537	references added concerning registers initialization
23	1541	"To switch the RS-CAN FD module ..." in "RCMC Bit" description added
23	1589	RSCANnRFIDx access options completed
23	1594	references to register tables in "CFTML[3:0] Bits" description corrected
23	1638	reference to register table in "TXQDC[3:0] Bits" description added
23	1651	reference to figure in "RTME Bit" description added
23	1654	references added concerning registers initialization
23	1658	"To switch the RS-CAN FD module ..." in "RCMC Bit" description added
23	1677	"GWBRS Bit" description corrected
23	1678	"Modify this bit only in channel reset mode." added to "GWEN Bit" description
23	1682	note added in "TDCR[7:0] Flag" description
23	1687	", and depending on the DRE bit ..." added in "CMPOC Bit" description
23	1700	references to ISO standards in RSCFDnCFDGCRCCFG register description corrected
23	1708	RSCFDnCFDGAFLP0_j.GAFLDLC[3:0] description corrected
23	1732	references to register tables in "CFTML[3:0] Bits" descritpion corrected
23	1743	"Reserved" bit description corrected
23	1777	reference to register table added in "TXQDC[3:0] Bits" description
23	1790	reference to figure in "RTME Bit" description added
23	1799	BOM[1:0] settings corrected in figure
23	1799	note 3 added
23	1806	", and depending on the DRE bit ..." added
23	1810	"(Use this count source ..." added

Revision 2.20 History, Date Jan 26, 2018 (5/8)

Section	Page	Summary
23	1811	note 1 added in figure
23	1818	addition concerning channels not included in the test
23	1822	correction in figure for "ClassicalCAN mode:": "TSEG1 > TSEG2 > SJW" ->"TSEG1 > TSEG2 >= SJW"
23	1847	note 1 added in figure
24	2066	Usage note "Receive frame interrupt and descriptor interrupt may be issued before completion of writing data" added
26	2084	correction concerning WDTAnMD value becoming effective
26	2084	rephrased: "generated generates an error," -> "generated causes an error,"
26	2089	paragraph "After writing the processing ..." rephrased
29	2154	TAUBnCMORm.TAUBnMD[4:0] bit description completed
29	2165	TAUBnTOLm description corrected
29	2199	typo correction (TAUBnMDO -> TAUBnMD0)
29	2222	TAUBnCNTm 0000H and 0001H lines corrected in figure
29	2296	TAUBnCNTm 0000H and 0001H lines corrected in figure
29	2301	TAUBnCNTm 0000H and 0001H lines corrected in figure
29	2302	TAUBnCNTm 0000H and 0001H lines corrected in figure
29	2303	TAUBnCNTm 0000H and 0001H lines corrected in figure
29	2304	TAUBnCNTm 0000H and 0001H lines corrected in figure
29	2306	TAUBnCNTm 0000H and 0001H lines corrected in figure
29	2311	TAUBnCNTm 0000H and 0001H lines corrected in figure
29	2337	brackets settings corrected in "Equations"
30	2375	TAUJBnCMORm.TAUJnMD[4:0] bit description completed
30	2385	TAUJnTOLm description corrected
30	2415	figure corrected
31	2462	note added concerning RTCA interrupts
31	2463	correction: "32 kHz to 4.194304 MHz" -> "32 kHz to 4 MHz"
32	2519	typo correction ("or ADCE1" removed)
37	2714	VDCE1_VO_CLK in Open LDI output mode corrected
37	2719	note added concerning video clock generator's registers module name in header files
37	2805	description of D1M1A video output selection for D1M1-V2 deleted
37	2806	section "Phase shift" revised
37	2806	cleaned up figure
37	2834	typo correction in figure (VDCECTL.VOS -> VDCECTL.VOSL)
38	2849	typo correction (INTVDCE0IRVLINE -> INTVDCE0OIRVLINE)
38	2888	"Input timing by data enable" figure and explanation corrected
38	2908	scaler 0 sync signals flow corrected
38	2909	scaler 1 sync signals flow corrected
38	2918	notes added in figure
38	2930	notes added in figure
38	2936	note added in figure
38	2937	note added in figure
38	2954	note added to SC0_SCL0_FRC7.SC0_RES_F_HW[10:0] bit description
38	2993	note added concerning CLUT table switching
38	2999	note added to SC1_SCL0_FRC7.SC1_RES_F_HW[10:0] bit description

Revision 2.20 History, Date Jan 26, 2018 (6/8)

Section	Page	Summary
38	2999	typo correction (SC0_RES_F_HW -> SC1_RES_F_HW)
38	3025	note added concerning CLUT table switching
38	3066	formulars corrected in "Alpha Blending Calculation" because of up-rounding
38	3067	note added concerning CLUT table switching
38	3094	note added concerning CLUT table switching
38	3114	note added concerning CLUT table switching
38	3133	remark added concerning vertical front porch setting
38	3134	note added to figure
38	3141	note added to OIR_SCL0_FRC7.OIR_RES_F_HW[10:0] bit description
38	3141	typo correction (SC0_RES_F_HW -> OIR_RES_F_HW)
38	3179	added "Timing of Parallel to Serial Conversion in Triple Speed Mode Mode" figure for OUT_SEQ_SEL = 1
38	3179	figure "Timing of Parallel to Serial Conversion in Triple Speed Mode" corrected
38	3179	OUT_SEQ_SEL included in "Specifications of Serial RGB Output" table
38	3185	correction: "...every horizontal period." -> "...every vertical period."
38	3186	correction: "TCON_POLA_HS[10:0] + 1" -> "TCON_POLA_HS[10:0]"
38	3186	correction: "TCON_POLB_HS[10:0] + 1" -> "TCON_POLB_HS[10:0]"
38	3238	correction: "TCON_STH_HS[10:0] + 1" -> "TCON_STH_HS[10:0]"
38	3240	correction: "TCON_STB_HS[10:0] + 1" -> "TCON_STB_HS[10:0]"
38	3242	correction: "TCON_CPV_HS[10:0] + 1" -> "TCON_CPV_HS[10:0]"
38	3249	OUT_SEQ_SEL bit added to OUT_SET register
40	3276	"Restriction" added
40	3280	(max. horizontal/vertical size is 128) added
40	3290	"Completion signal failure" added in figure
40	3292	notes added concerning indices
40	3304	notes added to VOCAnMmCFG1 register’s bits descriptions
40	3306	note added concerning VOCA threshold
43	3384	initial values of SPEAnSkBYP corrected
45	3453	notice added concening index j format
45	3454	notice added concening index j format
45	3456	caution added concerning ADCEnVCRj setting
45	3457	caution added concerning ADCEnVCRj setting
45	3457	initial value of ADCEnTSNVCR register corrected
45	3458	notice added concening index j format
45	3459	notice added concening index j format
45	3474	ADCEnOWER.SGPR[18:0] bit description completed
45	3484	typo corrections in register description (ADCEnTSNSGSEFCRx -> ADCEnTSNSGSEFCR)
45	3504	correction: "As the trigger signal PVCR_TRG..." -> "If the trigger signal PVCR_TRG..."
45	3509	figure corrected
45	3520	1st usage note item ("To use the standby function, ...") removed
46	3610	ISMnEMU.ISMnSVDIS bit description improved
48	3625	initial settings for FACI and flash sequencer error sources No. 17, 18 corrected
52	3683	OTP description with regards to configuration clearing command corrected
52	3683	security setting options for D1M1A, D1M1-V2 corrected

Revision 2.20 History, Date Jan 26, 2018 (7/8)

Section	Page	Summary
52	3685	Hardware protection summary description modified
52	3696	note added concerning chip ID register module name in header files
52	3698	"Debug Interface" table added
52	3702	explanation added concerning masters for code flash access
52	3715	DFERRINT register's "Initial value" corrected
53	3727	LR1STEADRn register description corrected
53	3737	ECDEDFm, ECSEDFm clearing conditions rephrased
53	3738	ECOVFF clearing conditions rephrased
53	3738	ECER2C bit description corrected
54	3749	note 3 for INTECCSED of VRAM1 added
54	3758	note added concerning GPU2DTRCTL register module name in header files
54	3772	ECDEDFm, ECSEDFm clearing conditions rephrased
54	3773	ECOVFF clearing conditions rephrased
54	3775	VRAMnEADm register description corrected
55	3783	boundary scan instructions table corrected
56	3786	HYPB clock supplies corrected
56	3787	note added concerning support of dual-die-HyperRAM devices
56	3787	correction: HyperRAM -> HyperFlash/RAM
56	3787	correction: manual mode does not support write access to HyperRAM
56	3787	supported burst type added
56	3788	INT output removed from HyperBus Controller block diagram
56	3791	SSLDR register's initial value corrected
56	3791	SSLDR.SCKDL[2:0] bit description corrected
56	3791	SSLDR.SLNDL[2:0] bit description corrected
56	3792	DRCR.SSLN access type corrected (R -> W)
56	3792	DRCR.SSLN bit description corrected
56	3792	note in DRCR.SSLE bit description corrected
56	3795	correction: "sets the bit size of the command, optional command, address, option data, and read data" -> "sets the read data"
56	3796	SMCR.SPIE bit description corrected
56	3796	SMCR.SPIRE bit description corrected
56	3796	SMCR.SPIWE bit description corrected
56	3800	SMENR register description corrected
56	3802	SMRDR0 register description corrected
56	3803	SMRDR1 register description corrected
56	3809	SMDRENR.HYPE[2:0] bit description corrected
56	3810	access type for PHYCNT.WBUF bit corrected (R -> R/W)
56	3810	crossreference to write buffer section corrected in PHYCNT register description
56	3811	correction: "the serial flash memory and HyperBus to be connected" -> "the HyperBus to be connected"
56	3812	"Set the transfer data size in bit units." removed from initial setting flow figure
56	3813	correction: "SMWDR0 = 0x3900_0000" -> "SMWDR0 = 0x3800_0000"
56	3814	typo correction (HyperFlash -> HyperRAM)
56	3814	"SMDRENR = 0x0000_5111" added
56	3815	correction: "4 Kbyte boundary" -> "1 Kbyte boundary"

Revision 2.20 History, Date Jan 26, 2018 (8/8)

Section	Page	Summary
56	3815	correction: "serial flash memory" -> "HyperFlash/HyperRAM memory"
56	3819	table added
56	3820	"TEND Bit" description corrected
56	3820	correction: "F2FF 8800H to F2FF 8900H" -> "F2FF 8800H to F2FF 88FFH"
56	3821	"Is transfer completed?" check removed from flow diagram
57	3823	description of 2nd chip select for Octa memory interface added
57	3823	OctaBus chip select CS_1/MCS1 added
58	3879	"Separate remapping records tables for each device in the bank." removed
58	3879	note added to "Command queuing mechanism"
58	3883	CONTROL.AUTO_READ_STAT_EN bit description improved
58	3884	CONTROL.READ_STAT_EN bit description improved
58	3887	note added to STATUS.DATASIZE_ERROR_ST bit dscription
58	3892	ECC_CTRL.ECC_SEL[1:0] bits description improved
58	3893	ECC_STAT.ECC_ERROR_0 bit description improved
58	3893	ECC_STAT.ECC_OVER_0 bit description improved
58	3895	ECC_CNT.ERR_LVL[5:0] bit description improved
58	3901	BBM_CTRL.RMP_INIT bit description improved
58	3908	typo correction (read/buys -> read/busy)
58	3916	register name correction: "FIFO control register (FIFO_STATE)" -> "FIFO status control register (FIFO_STATE)"
58	3925	correction: "the second command code is encoded in the CMD_1 instruction field" -> "the second command code is encoded in the CMD_3 instruction field;"
58	3926	correction: "the second command code is encoded in the CMD_1 instruction field" -> "the second command code is encoded in the CMD_2 instruction field"
58	3926	figure corrected
58	3932	"TWO PLANE PAGE READ" instruction codes corrected
58	3943	"TWO PLANE PAGE READ" instruction codes corrected
58	3943	"QUEUE PAGE READ Instruction Encoding" corrected
58	3952	note added to diagram
58	3953	caution added to INT_STATUS.DMA_INT_FL description
58	3959	added recommendation concerning interrupt
58	3973	"Size of Correction Bytes" table corrected

RH850/D1L/D1M User's Manual: Hardware

Publication Date: Rev.2.20 Jan 26, 2018

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709 Quantum Plaza, No.27 ZhichunLu, Haidian District, Beijing, 100191 P. R. China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, 200333 P. R. China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 1207, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700, Fax: +91-80-67208777

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 5.0

RH850/D1L/D1M

Renesas Electronics Corporation

R01UH0451EJ0220